Bachelor Thesis

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

WinterCup at Haas-Tenniscenter

Author: Matthäus Vogla

Enrollment Number: 1252933

Examiner: ao.Univ.Prof. Dr. Rony G. Flatscher

June 11, 2017

WIRTSCHAFTSUNIVERSITÄT WIEN
Vienna University of Economics and Business

Bachelor Thesis

Title of the Bachelor Thesis:	WinterCup at Haas-Tenniscenter
Author (last name, first name):	Vogla, Matthäus
Student ID number:	1252933
Degree program:	Business, Economics and Social Sciences (BESC)
Examiner (degree, first name, last name):	ao.Univ.Prof. Dr. Rony G. Flatscher

I hereby declare that

1. I have written this Bachelor thesis independently and without the aid of unfair or unauthorized resources. Whenever content was taken directly or indirectly from other sources, this has been indicated and the source referenced.
2. this Bachelor thesis has neither previously been presented for assessment, nor has it been pub-lished.
3. this Bachelor thesis is identical with the assessed thesis and the thesis which has been submit-ted in electronic form.
4. (only applicable if the thesis was written by more than one author): this Bachelor thesis was written together with

   The individual contributions of each writer as well as the co-written passages have been indicated.

11. 06. 2017
Date

Signature

Abstract

This work presents a program named WinterCup, which is an administrative and evaluation program for the winter cup tournament. The program was conceived for the use in the Haas-Tenniscenter in Vösendorf (Austria) and will be introduced there in October.

Moreover, the structure will be explained step-by-step in this thesis. Beginning with an short introduction of the company and the winter cup tournament, followed by the installation guide of the needed programs. Afterwards, the database structure will be explained before coming to the graphical user interface. Last but not least, snippets with the most important code sections will be listed and explained to make it easier for the reader to understand the structure of WinterCup.

This work is not only intended to explain how the program is structured and created, but also to show how programs can easily be created with the human-orientated programming language ooRexx.

Table of Content

1 Introduction 10

2 Introduction of Haas-Tenniscenter 11

2.1 Winter Cup at Haas-Tenniscenter 11

2.1.1 Game Rules 11

2.1.2 Scoring 12

3 Overview of the used Programs and System Requirements 13

3.1 System Requirements 13

3.2 Open Object Rexx (ooRexx) 14

3.3 Bean Scripting Framework for ooRexx (BSF4ooRexx) 15

3.4 Apache Open Office 15

3.5 MySQL Connector/J Application Programming Interface (API) 16

4 Installation Guide 17

4.1 Obligatory Programs 17

4.1.1 Java 17

4.1.2 Apache OpenOffice 17

4.1.3 ooRexx 17

4.1.4 BSF4ooRexx 18

4.1.5 MySQL Connector API 18

4.2 Additional Programs 19

4.2.1 Gluon Scene Builder 19

4.2.2 MySQL Workbench 20

5 Database 21

5.1 Structure of the Tables 22

5.1.1 Players 22

5.1.2 Team 23

5.1.3 Singles 24
5.1.4 Double 25
5.2 Tables Relations 26
5.3 SQL Queries 28
5.3.1 Select Command 28
5.3.2 Insert Into Command 29
5.3.3 Update Command 29
5.3.4 DELETE Command 29
6 Graphical User Interface (GUI) 30
6.1 Creating a GUI 30
6.2 GUI of the Main Program 33
6.2.1 Design 33
6.2.2 Implementation 34
6.3 GUI of the Second Program 37
6.4 GUI with ooRexx 39
6.4.1 Launch of the GUI 39
6.4.2 Actual Code 40
7 Development of the Main Program 42
7.1 Overview 42
7.2 Database Connection 44
7.3 Filling the Choice Boxes with Relevant Data 46
7.4 Input of Data by the User 48
7.5 Save Form 50
7.6 Data Export 55
8 Development of the Second Program 62
8.1 Overview 62
8.2 New Player 63

8.3 New Team 66
8.4 Edit Team 67
8.4.1 Editing 67
8.4.2 Deleting 69
8.5 Edit Player 70
8.5.1 Editing 70
8.5.2 Deleting 71
9 Conclusion 72

List of Figures

Figure 1: Correct Installation of BSF4ooRexx on Windows 18
Figure 2: Players Table 22
Figure 3: Exemplary Entries of the Players Table 23
Figure 4: Team Table 23
Figure 5: Exemplary Entries of the Team Table 24
Figure 6: Singles Table 25
Figure 7: Exemplary Entries of the Singles Table 25
Figure 8: Double Table 26
Figure 9: Tables Relationships 27
Figure 10: GUI Anchor Pane 31
Figure 11: GUI Control Elements Settings 31
Figure 12: Official Scoresheet of the NÖTV 33
Figure 13: Main Program ‘Scoresheet’ 34
Figure 14: Error Message 37
Figure 15: Main Programm 'Data Export' 37
Figure 16: Secondary Program 'Add Player' 38
Figure 17: Secondray Programm Remaining Tabs 39
Figure 18: Choice Box Separation Line 49
Figure 19: Double Selection of a Player 50
Figure 20: Export Current Ranking 61

List of Snippets

Snippet 1: GUI FXML-File 32
Snippet 2: FXML-File ‘onAction’ Event 35
Snippet 3: FXML-File Loading Classes 35
Snippet 4: FXML-File ChoiceBox 36
Snippet 5: FXML-File TextField 'onAction' Event 36
Snippet 6: Launch of the GUI 40
Snippet 7: Actual Code Routine 40
Snippet 8: Snippet Rexx Annotation 41
Snippet 9: MySQL Database Connection 45
Snippet 10: Closing Database Objects 46
Snippet 11: Database Connection Routine – Choice Box properties 46
Snippet 12: Database Connection Routine 47
Snippet 13: Routines called by a Key Release Event 49
Snippet 14: 1st Match Input Control 51
Snippet 15: Routine that Checks the Correct Input of a Set 52
Snippet 16: Routine that Checks the Correct Input of a Match-Tie-Break 52
Snippet 17: Determining the Winner of a Match 53
Snippet 18: Points Division and Storage of Points, Games & Sets 53
Snippet 19: Saving All Match Information 55
Snippet 20: Start of the various Export Programs 56
Snippet 21: Export Current Ranking 1 58
Snippet 22: Writing a Text into the Spreadsheet 58
Snippet 23: Export Current Ranking 2 59
Snippet 24: SQL Query Current Ranking 59
Snippet 25: Export Current Ranking 3 60

Snippet 26: Access to the Handover Parameter 61
Snippet 27: New Player (Second Program) 64
Snippet 28: New Player 2 (Second Program) 65
Snippet 29: Check if Player Exists 65
Snippet 30: New Team (Second Program) 66
Snippet 31: Edit Team 67
Snippet 32: Edit Team 2 68
Snippet 33: Delete Team 69
Snippet 34: Delete Team 2 69
Snippet 35: Edit Player 70
Snippet 36: Check if a Player is a Team Leader 70

List of Flow Chart

Flow Chart 1: Main Program 44
Flow Chart 2: Second Program 62

List of Table

Table 1: Game Information Stored in the Database 54
Table 2: Export Options 57
Table 3: Second Program Tabs 63

1 Introduction

This bachelor thesis deals with the description of the WinterCup program. The task of the program is to create, edit and evaluate the winter cup, which is a tennis tournament over a whole winter season. Up to now, the winter cup has been documented and evaluated by hand. Since the Haas-Tenniscenter is a small family business and the evaluation can be time consuming, the desire for an electronic solution came up. The practical part of this thesis was to devise a software system that helps to organize the tournament. Thus, the idea of this program came up after consultation with the university professor Rony G. Flatscher, who supervised the implementation. The company Haas-Tenniscenter is hosting this event and will start using this program in the upcoming season. Tight interaction with the actual users during the design process helped to implement a software system that fits the needs by an easy to use user-interface.

What is more, the WinterCup program, also referred as WinterCup, was implemented with the use of ooRexx, JavaFX and BSF4ooRexx. BSF4ooRexx is a technology that enables the interaction of ooRexx and Java to take advantage of both programming languages. The program consists of two parts. First, the main program which allows the user to enter games scores and export data in tabular-format by using the scripting framework of Apache OpenOffice. The second part of the program, which can only be started by the main program, is used to create, edit and delete teams or players.

The paper is designed in such a way that the reader can fully understand the structure of the program and thus also be able to recreate the program independently after reading this paper. However, a basic knowledge of ooRexx is mandatory in order to understand the syntax of WinterCup fully as only the most important and special parts of the code are explained in this paper. The remaining code can be found in the appendix.

The following chapter, will briefly present the used programs which were needed to create the program. Subsequently, the database and the graphical user interface of WinterCup will be described. Once this is complete, the main parts of the programs syntax are explained.

2 Introduction of Haas-Tenniscenter

The Haas-Tenniscenter is a tennis center located in Vösendorf, Austria. The company was established in 1980 by the Haas family. In 2007 the company was taken over by the Vogla family and has been managed by them since then.

Since 2010 the winter cup is carried out by the Haas-Tenniscenter in order to entertain their guests by giving them a possibility to compete against each other in a friendly environment without feeling the pressure of an official game. As this company is a small family business, with three family members who work in it, the capacities are almost reached by the daily business. Therefore the implementation of an electronic solution can make work easier as the maintenance of the winter cup can be very time consuming.

2.1 Winter Cup at Haas-Tenniscenter

The winter cup is an event which is organized at the Haas-Tenniscenter for seven years. It is a tournament which is held over the entire winter season. At the beginning of each season, teams and players register for it and compete against each other.

Each team needs at least four players in which one has to be the team captain. The team captain or team leader is responsible that his team is complete on gamedays. Furthermore, the team leader has to set up his players, in singles, according to their playing skills. In double, a tactical line-up can be important.

On gamedays two teams compete against each other. One competition consists of four single matches and two double matches.

2.1.1 Game Rules

The winter cup is a privately organized tournament by the Haas-Tenniscenter and not an officially organized tournament by the ÖTV, Österreichischer Tennisverband (English: Austrian Tennis Association). Nevertheless, the winter cup follows the official rules of the ÖTV. Each match is played on two winning sets. Instead of the third set a match tiebreak is played out. A match tie-break is a tie-break played up to ten instead of seven points. What is more, in double the no ad rule is applied. [WiTen]

As mentioned above, a team consists of at least four players, though, no limits are set to the top. For tactical, temporal or other reasons it is not mandatory that a player must play in singles and double. One individual or all players can be exchanged.

In addition, teams are allowed to borrow team members to other teams in order to compensate spontaneous dropouts of players on gamedays. Additionally, some players participate in the tournament without being part of a team. This players help out in case of a dropout.

2.1.2 Scoring

A team receives two points for winning a whole encounter, while the loser receives zero points. Winning an encounter is achieved by winning four matches independent of the constellation. That means for example winning two singles and two double matches. This results that winning by the scores of 6:0, 5:1 or 4:2 will lead to two points for the winner and zero for the loser. For a tie, each team receives one point.

The scores are then summed up together and a ranking is created. In case of equality in the rankings table, first the direct comparison, then the game difference and then the set difference of all results is compared. If there is still a tie, finally the lot decides.

One can see that not only the pure victory is important but also with what score the victory was achieved as it can influence the overall ranking too.

3 Overview of the used Programs and System Requirements

This chapter provides information about system requirements which are needed in order to run the program. Furthermore, the programming languages ooRexx and BSF4ooRexx as well as Apache OpenOffice and the MySQL Connector/J API are briefly presented to reader.

3.1 System Requirements

Before the reader can execute the program WinterCup on its own computer, some software dependencies have to be installed to ensure that WinterCup can work properly. The following programs are needed:

- Java
- ooRexx
- BSF4ooRexx
- Apache OpenOffice
- MySQL Connector API

Since the programs communicate it is necessary that they have the same bitness. As Apache OpenOffice is offered only as a 32-bit version, the remaining programs have to be installed with the very same bitness as well. Installing the programs on different bitnesses will cause problems.

Furthermore, the instructions have been programmed and tested on a Windows 7 64-Bit system, however, it is also possible to make the program run on Macintosh and a Linux system as well, because all necessary programs are also available for them. Of course, a Windows 7 or higher system with arbitrary bitness may be used.

In order to make the development of a program easier, it is highly recommended to use additional development tools. The following tools were used for the development of WinterCup.:

- Gluon Scene Builder
- MySQL Workbench

The Scene Builder was used to create the graphical user interface (GUI) and will be explained later in Chapter ‘Graphical User Interface(GUI)’.

For an easier and faster way to create and edit the database tables the program MySQL Workbench was used. Following this link (https://dev.mysql.com/doc/workbench/en/) will forward the reader to the official documentation page of MySQL Workbench, which can help the reader to get started with it.

3.2 Open Object Rexx (ooRexx)

Open Object Rexx, also referred as ooRexx, is a programming language based on Rexx which was invented in 1979 by the IBM employee Mike F. Cowlishaw. Rexx is a special programming language as it is human-orientated – English-like syntax- in contrast to other languages. This has the advantage that it can be easily learned, remembered and faster adopted. Moreover, it makes the maintenance of programs cheap and easy. [Rexx13, p.6]

In the beginning of the 90ties, IBM started to extent Rexx and transformed it into an object-oriented language. In 1997, the first commercial version of Object Rexx was distributed with OS/2 Wrap4 for IBM AIX and MS Windows. Seven years later, in 2004, IBM decided to hand over the source code of Object Rexx to the no profit organization Rexx Language Association (RexxLA). This step was made to continue the programming language as open source. One year later, the first version of open source Object Rexx was released under the name Open Object Rexx.[Rexx13, p.4][FlatRx, p.5-8]

Not only the advantages mentioned above make ooRexx as a programming language so extraordinary. Literature on Rexx often states also the circumstances such as: [FlatRx, p.iv-vi]

- fast and powerful,
- great documentation,
- free and open source,
- multiplatform and

- extensible.

In this thesis, the programming language is not explained but functions and the reason for their use. For more information and explanations of the programming language, it is referred to the book ‘Introduction to Rexx and ooRexx’. [Flat13]

3.3 Bean Scripting Framework for ooRexx (BSF4ooRexx)

In the following a brief introduction of the Bean Scripting Framework for ooRexx (BSF4ooRexx) is given. BSF4ooRexx is a technology that allows the interaction between ooRexx and Java. Java has become one of the most important and most used programming languages over nowadays The reason for this is not only the platform independence but also the variety of libraries, software packages and technologies. BSF4ooRexx encapsulates Java as ooRexx and makes everything that is available in Java also available for ooRexx.

In the WinterCup program, this property is used to create the Graphical User Interface in JavaFX (see Chapter ‘6 Graphical User Interface (GUI)’). To enable BSF4ooRexx, the packages BSF.CLS must be included in each new program created.

> *“The ooRexx package BSF.CLS defines public routines and the ooRexx class BSF which allows creating Java objects from a Rexx program.”[Flat13, p.163]*

Prerequisite is, of course, that BSF4ooRexx, Java and ooRexx are installed on the computer. Further information on BSF4ooRexx and its syntax can be found in the book 'Introduction to Rexx and ooRexx'. [Flat13, p.159]

3.4 Apache Open Office

Apache OpenOffice, short AOO, is an office productivity software suite which is free and open source. Because OpenOffice supports a huge variety of data types and is free of charge, in contrast to other office suites which can be very expensive, it is very popular and frequently used. It has to be emphasized that AOO is equal in terms of functionality compared to competitor office suites. Furthermore, OpenOffice is

available in 38 languages and for the three most used operating systems Linux, Macintosh and Windows. AOO contains of the following applications: [WiAO17]

- Writer (word processor program)
- Calc (spreadsheet)
- Draw (drawing application)
- Math (formula editor)
- Base (database management application)

The WinterCup program uses only the Calc application in order to export data such as game results, the current ranking or a full list of all players. The implementation is done by means of the scripting framework of OpenOffice. Macros are created that automate the needed steps which create the wanted Calc-file with the desired information. These small programs are written with the help of the class library UNO.CLS which is a part of BSF4ooRexx. This library includes routines and ooRexx classes. These routines and classes enable the interaction with AOO. UNO.CLS uses the Java interface of the Open Office Scripting Framework and therefore includes the BSF.CLS package already. Good sources of syntax documentations and information about scripting AOO with ooRexx are the Slides of Professor Rony G. Flatscher named ‘Scripting Apache OpenOffice’ (http://wi.wu.ac.at:8002/rgf/wu/lehre/autojava/material/foils/AutoJava%2dBSF4ooRexx%2d04%2dOOo%2dACE12%2epdf) as well as the book ‘Introduction Rexx and ooRexx’. [Flat13]

3.5 MySQL Connector/J Application Programming Interface (API)

The MySQL Connector/J API is a driver which helps the programmer to easily establish a MySQL database connection. Since Rexx does not natively support the interaction with MySQL out-of-the-box, the Java Database Connectivity interface is used to circumvent this restriction.

The next chapter explains the integration of the API to the Java CLASSPATH, thereby enabling access to the functions therein. Furthermore, the usage is explained in chapter ‘7.2 Database Connection’.

4 Installation Guide

This short installation guide is intended to help the reader to install the necessary programs. (Note: If newer versions of the programs listed below are available, then these should be installed as errors will be corrected and functions could be extended.)

The given installation sequence should be preserved to avoid errors.

4.1 Obligatory Programs

The installation of the following programs is obligatory to run the WinterCup program.

4.1.1 Java

Under the following link the Java download for common systems can be found: https://java.com/de/download/manual.jsp

Minimum requirement is Java 8.

Note: The command line can be used to check the version on Windows, Macintosh and Linux, if Java is already installed. The following command is used: java -version.

4.1.2 Apache OpenOffice

The Apache OpenOffice productivity software usage in the WinterCup program is already explained above. The download for any operating system can be found on the Apache OpenOffice website under the following link: https://www.openoffice.org/de/download/

Note: If problems with the installation occur, a great installation guide can be found on: https://www.openoffice.org/de/doc/setupguide/

4.1.3 ooRexx

The programming language ooRexx is the next program to be installed. It should be taken care that the latest version – currently ooRexx 5.0.0beta – with the needed bitness is chosen. The download can be found under the following link: https://sourceforge.net/projects/oorexx/files/oorexx/

Note: For Macintosh, it is important to note that ooRexx is already included in BSF4ooRexx, and hence does not need to be installed separately.

4.1.4 BSF4ooRexx

It is mandatory that all the programs listed before have been installed correct to ensure that BSF4ooRexx can be installed without any errors. After downloading BSF4ooRexx under Windows, all files have to be unpacked at first. Now the installer for windows can be found in the subdirectory *bsf4oorexx\install\windows.*

In case of the Macintosh version, the DMG-File has to be opened and the displayed instructions have to be followed.

The latest version can be found here: https://sourceforge.net/projects/bsf4oorexx/files/beta/20161026/

Note: On Windows, the installation has been successful when the following image is displayed.

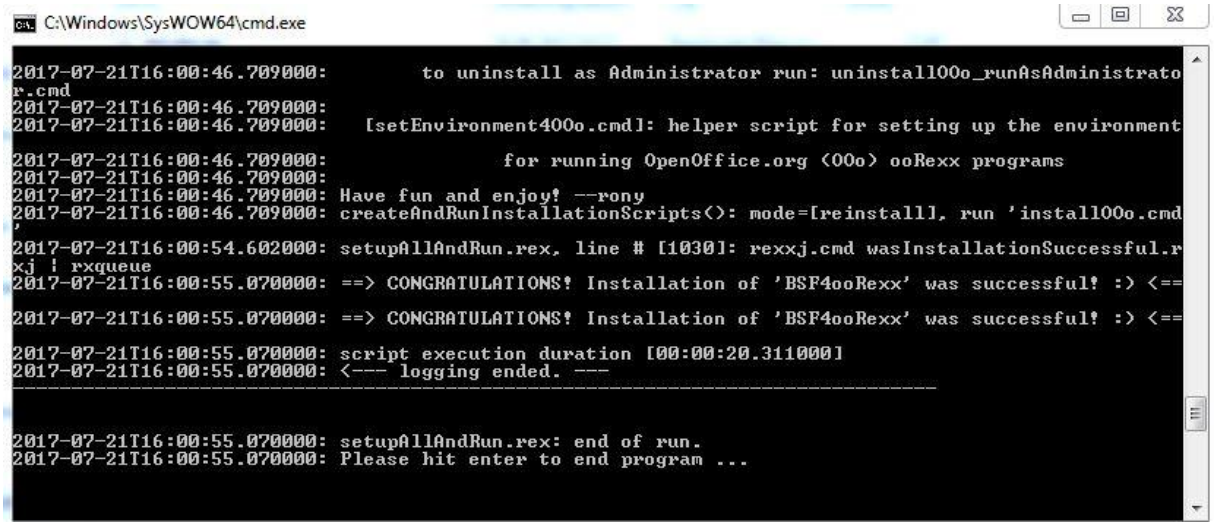


Figure 1: Correct Installation of BSF4ooRexx on Windows

4.1.5 MySQL Connector API

For the purpose of establishing an easy database connection, the MySQL Connector is used. After downloading the MySQL Connector application programming interface (API), the jar-file has to be added to the environment variable CLASSPATH of Java. The points listed below explain how to integrate the jar-file into the CLASSPATH of Java under Windows 7 [OrJaDo]:

- From the desktop, right click the Computer icon.
- Choose Properties from the context menu.
- Click the Advanced system settings link.
- Click Environment Variables. In the section System Variables, find the CLASSPATH environment variable and select it.
- Click Edit. If the CLASSPATH environment variable does not exist, click New.
- In the Edit System Variable or New System Variable window, specify the value of the CLASSPATH environment variable. Click OK. Close all remaining windows by clicking OK.

The download of the API is performed from this page: https://dev.mysql.com/downloads/connector/j/

After finishing the download one has to extract the 'mysql-connector-java-x.x.xx-bin.jar' file and add it to the CLASSPATH as explained above.

For more information about adding .jar-files to the CLASSPATH on other operating systems it is referred to the following link: https://docs.oracle.com/javase/tutorial/essential/environment/paths.html

4.2 Additional Programs

The additional programs allow to develop much more efficient, however are not mandatory.

4.2.1 Gluon Scene Builder

The Gluon Scene Builder is used for the easier creation of the graphical user interface. The operation of this program is explained, based on the example of WinterCup, in the chapter 'Graphical User Interface (GUI)'. Here the link to the download: **Fehler! Linkreferenz ungültig.**

Note: The installation is not mandatory for the use of the WinterCup program. What is more, this program should use the bitness of the operating system, as the program runs more smoothly and does not interact directly with the programs above. An explanation of how to determine the bitness of a Windows operating system is found

here: https://support.microsoft.com/en-us/help/827218/how-to-determine-whether-a-computer-is-running-a-32-bit-version-or-64

4.2.2 MySQL Workbench

This program was used for the creation of the database tables and to check if the entries were made correctly throughout the test phase. Under this link one can find the download of the program for all operating systems: https://dev.mysql.com/downloads/workbench/

The installation is self-explanatory, simply follow the introductions of the installer. If still any problems or questions arise then follow this link: https://dev.mysql.com/doc/workbench/en/wb-installing.html

5 Database

This chapter briefly and succinctly defines all the important aspects of a database in order to explain the database of WinterCup. There are many definitions of a database, one of which defines the database as an ordered, self-describing collection of data related to each other. The administration of a database is carried out by means of a computer program which is called the database management system (DBMS). The DBMS manages the organization of the data and the access to them. Database management systems enable better organization of data and allows to easily modify, add, or delete data of the database. Most important, data can be found quickly and easily by using search functions. [Geis14, p.21-23]

Databases consists of different infrastructures which are described by a database model. These models explain how data is stored, organized and manipulated in a database. Even though various types of databases gained in popularity throughout the last decade, the relational model is still one of the most widely used models in the nowadays systems landscape and used for the WinterCup database. [Geis14, p.52][WiDM17]

A database which is organized using the relational model is called a relational database. Relational databases are used for electronic data management in computer systems. The associated database management system is called relational database management system (RDBMS). The relational database is characterized by a two-dimensional table which consist of rows, also called tuples, and columns. Each tuples consists of a set of attributes – columns. Relationships between tables are created using primary and foreign keys. A primary key of a table consists of a subset of attributes that uniquely identify a record. A foreign key, is a primary key defined in another table. In other words the foreign key refers to a uniquely identified row of another table. Further down in the chapter, relationships between tables are explained based on the WinterCup database. Moreover, the data of a relational database is queried, manipulated and deleted by the use of the database language SQL(Structured Query Language). [Geis14, p.59-63][WiRMen][WiRDde][WiFK17]

In WinterCup the database is used to keep record of games results, scores of already played encounters, team names, players' names, etc.. The tables and the relationships between them is described in the following subchapters.

5.1 Structure of the Tables

In this chapter the tables of the WinterCup database will be displayed as well as some important information, so that their structure can be fully understand.

There are many ways creating a database and its tables. One easy way is the use of MySQL Workbench. This program relieves once work by its self-explanatory form and easy use. In the chapter 'Installation Guide' there is a download link and a link to an installation guide plus instructions for the use of MySQL Workbench.

5.1.1 Players

In the 'Players'-table, general information such as the first name, last name and telephone number, as well as the team to which the player belongs, are stored. This information are entered by the user via WinterCup's graphical user interface.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idSpieler	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
Vorname	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Nachname	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Telefonnummer	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Team	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Figure 2: Players Table

Figure 2 illustrates the attributes of the table and their datatypes. The primary key, as already mentioned before, is a unique identifier that must not be NULL. In addition, MySQL provides support of so-called sequences that lets the database autonomously decide on the value for new records. The name of the primary key is 'idSpieler' and the datatype is integer (INT(11)). The columns 'Vorname', 'Nachname' and 'Telefonnummer' have the datatype VARCHAR(45). The column 'Team' represents the team to which the player belongs. It should be emphasized that only the ID of the team is stored here. This ID is a foreign key which also means that it is a primary key in the 'Team'-table. In summary, the 'Players'- and the 'Team'-table have a relationship with each other.

Figure 3 shows exemplary entries of the table ‘Players‘:

idSpieler	Vorname	Nachname	Telefonnummer	Team
4	Andrei	Sokol	06764480372	43
6	Mamica	Voola	06766301677	25
7	Sandra	Stadlbauer	067612345678	33
17	Martin	Purtscher	06006969696	29
19	Ben	Voola	06767155171	25
20	Stefan	Sulic	06801234567	33
25	Dieter	Wimmer	0676213123	27
26	Günther	Kohl	012312987243	27
27	Matthias	Kohl	0676123145478	27
35	Sami	Rassv	0676696969696	33
36	Alioscha	Piech	06761616161	27
43	Johannes	Meixner	067612923	27
49	Flo	Duck	0676192300123	31
50	Daniel	Wimmer	0671238234	27
51	Michael	Strauß	017676123	42
52	Michael	Deimel	0654324567	42
53	Markus	Krenn	0676125142283	42
54	Matti	Voola	06763207675	1
55	Lukas	Amann	098545678923	25
57	Thomas	Kaufmann	06641239123712	1

Figure 3: Exemplary Entries of the Players Table

The leaping behavior of the ‘idSpieler’ can be explained by the fact that entries have been deleted and new entries are drawn to the end.

5.1.2 Team

The structure of the ‘Team’-table follows the same concepts as the ‘Players’-table. Here the storage of the team name, reference to a player which is the team captain as well as the current points of the team is performed. The column ‘TeamCaptain’ represents a foreign key. See Figure 4:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idTeam	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
TeamName	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
TeamCaptain	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Punkte	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Games	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Sets	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Figure 4: Team Table

What is more, not only the current points of a team but also the total played games and sets are stored. The reason for that is, if two or more teams have the same points(column ‘Punkte’), the played games are compared. If there is still a tie then the sets are compared last. This is done in order to create a ranking. However, it can still

happen that a tie between teams prevails. Finally the lot decides. For example, if ‘Team A’ and ‘Team B’ have both 13 points. ‘Team A’ has played 52 games and ‘Team B’ has only played 51, therefore ‘Team A’ would be higher ranked than ‘Team B’. The same principle is used for the sets. Here is an example of what a table will look like with plausible values:

idTeam	TeamName	TeamCaptain	Punkte	Games	Sets
1	NO TEAM	0	0	0	0
25	TC Haas	19	44	67	112
27	Kolibris	25	113	69	195
29	TC Hennersdorf	17	14	61	48
31	Break 1	49	85	85	171
33	SundavTeam	20	13	52	66
35	Team 5	6	3	51	29
42	TC Arsenal	52	22	57	41

Figure 5: Exemplary Entries of the Team Table

In the first row of the table, the team called ‘NO TEAM’ can be found. This team represents a team in which players are referenced to that do not belong to any team. These players still participate in the winter cup as they are used when a team on match day is not complete. The team ‘NO TEAM’ cannot be deleted or modified and must have the ID number 1 in order to manage that the program can work properly. Further, it has to be mentioned that this team has no team leader, which means that the column ‘TeamCaptain’ has the value 0.

5.1.3 Singles

The purpose of the ‘Singles’-table is to save every played single match and its results. This table is important as the WinterCup program provides a data export to the user, where the results of already played encounters can be exported as an Apache OpenOffice Calc-file. To make this possible, it is necessary to restore exactly the played encounters by saving important information which are listed in Figure 6.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idSingles	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
ID-Spieler1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Team1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Spieler2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Team2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz1SP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz1SP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz2SP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz2SP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz3SP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz3SP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Runde	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Datum	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ErgebnisSP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ErgebnisSP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Figure 6: Singles Table

It is important to note that the columns ‘ID-Spieler1’ and ‘ID-Spieler2’ as well as ‘ID-Team1’ and ‘ID-Team2’ represent connections to the tables ‘Players’ and ‘Team’. When looking closely, the question arises why the teams have to be stored separately, since the teams to which the players belong are known for every player. This can be checked again in ‘Figure 2’. The reason for this is that players can be borrowed from other teams. In order to correctly reconstruct an encounter, which is done while creating the exported file, the teams that played have to be known. ‘Figure 7’ displays some exemplary entries of the ‘Singles’-table.

idSingles	ID-Spieler1	ID-Team1	ID-Spieler2	ID-Team2	Satz1SP1	Satz1SP2	Satz2SP1	Satz2SP2	Satz3SP1	Satz3SP2	Runde	Datum	ErgebnisSP1	ErgebnisSP2
51	52	42	43	35	6	1	6	2	NULL	NULL	1	2017-05-24	1	0
52	53	42	29	35	7	6	5	7	10	12	1	2017-05-24	0	1
53	51	42	17	35	1	6	2	6	NULL	NULL	1	2017-05-24	0	1
54	51	42	35	35	4	6	6	2	10	8	1	2017-05-24	1	0
55	52	42	36	35	6	0	6	1	NULL	NULL	1	2017-05-31	1	0
56	53	42	35	35	4	6	2	6	NULL	NULL	1	2017-05-31	0	1
57	51	42	17	35	6	2	2	6	10	12	1	2017-05-31	0	1
58	51	42	19	35	1	6	0	6	NULL	NULL	1	2017-05-31	0	1
59	52	42	36	35	6	0	6	1	NULL	NULL	1	2017-05-31	1	0
60	53	42	35	35	4	6	2	6	NULL	NULL	1	2017-05-31	0	1
61	51	42	17	35	6	2	2	6	10	12	1	2017-05-31	0	1
62	51	42	19	35	1	6	0	6	NULL	NULL	1	2017-05-31	0	1
63	49	31	6	25	6	1	6	2	NULL	NULL	1	2017-05-23	1	0
64	52	31	19	25	6	3	6	1	NULL	NULL	1	2017-05-23	1	0
65	26	31	4	25	7	6	7	5	NULL	NULL	1	2017-05-23	1	0
66	26	31	7	25	6	4	6	2	NULL	NULL	1	2017-05-23	1	0

Figure 7: Exemplary Entries of the Singles Table

What is more, it can be seen – Figure 7 column ‘Datum’ - that the format of the date is JJJJ-MM-DD, including the hyphen symbol which leads to the circumstance that the datatype of the date attribute is variable char and not integer.

5.1.4 Double

The only difference between table ‘Singles’ and table ‘Double’ are the attributes ‘ID-Spieler3’ and ‘ID-Spieler4’.These attributes are required to store the additional players.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idDouble	INT(11)	☑	☑	☐	☐	☐	☐	☑	☐	
ID-Spieler1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Spieler2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Team1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Spieler3	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Spieler4	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ID-Team2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz1SP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz1SP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz2SP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz2SP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz3SP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Satz3SP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Runde	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Datum	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ErgebnisSP1	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
ErgebnisSP2	INT(11)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Figure 8: Double Table

5.2 Tables Relations

Since large amounts of data are usually collected in a database, the intelligent storage space utilization is very relevant. One way is to avoid redundancies, which is achieved by linking tables together using primary and foreign keys. It should be noted that redundancies are not completely eliminated, but only minimized. Furthermore, it has to be said that complex problems cannot be modelled with a single table; the data must be split into several tables and these must then be connected to each other. What is more, by the use of IDs, which consists of integers instead of strings, even more memory space can be saved, since it usually requires less space [Geis14, p.111 & 115-116]

The following figure illustrates the tables of the WinterCup database and the relationships among them.

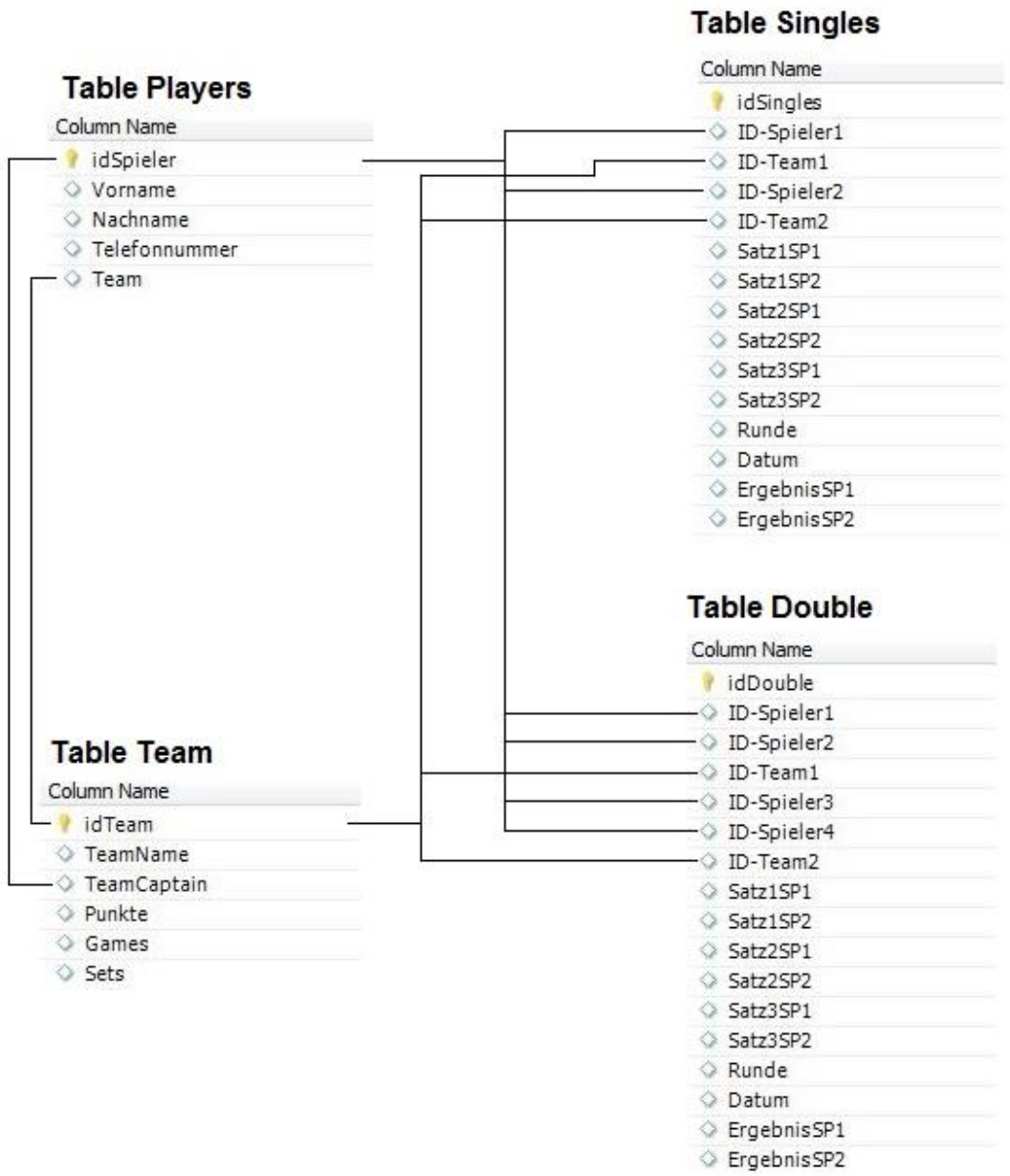


Figure 9: Tables Relationships

The primary key, idSpieler, of the table ‘Players’ is a foreign key of the table ‘Team’ called ‘TeamCaptain’. On the other side the primary key ‘idTeam’ of the table ‘Team’ is a foreign key of the table ‘Players’ called ‘Team’. As one can see in Figure 9 these two tables are related to each other.

Furthermore, it can be said that by linking these two tables, conclusions can be drawn. For example: To which team belongs Player X to or which team is led by which player. This is realized by tracing the primary/foreign keys in the corresponding tables.

The situation for the tables ‘Singles’ and ‘Double’ looks as follows. The relationships are created in such a way that the primary keys of the players and teams are the foreign key in the ‘Singles’- and ‘Double’-table but there is no reference from the tables ‘Singles’ and ‘Double’ to the tables ‘Players’ and ‘Teams’.

5.3 SQL Queries

To conclude the chapter database, the SQL(Structured Query Language) commands which are relevant for the usage of WinterCup will be explained briefly and succinctly. SQL is the standard language used to communicate with relational databases. Since WinterCup is a relational database, it is also addressed via SQL. With the SQL language and its commands, it is possible to manipulate, insert and retrieve data from a relational database. [Geis14, p.205-206] [W3scSq]

Though, the establishment of the database connection will be explained later in chapter ‘7.2 Database Connection’.

5.3.1 Select Command

With the select command, data from a database can be queried. The select command below retrieves all data from a table and stores them in a result table or result set [W3scSe]. What is more, only the data that fulfills the condition after the ‘WHERE’ clause is extracted [W3scWe]. If there is more than one condition which has to be fulfilled, logical operators like AND and OR may be used to combine several sub conditions [W3scAn].

*SELECT * FROM* table *WHERE* column1 = value1;

*SELECT * FROM* table *WHERE* column1 = value1 AND column2 = value2;

Additionally, the result set can be sorted in an ascending or descending order. The following example shows the syntax for the ‘ORDER BY’ keyword [W3scOr].

*SELECT * FROM* table *WHERE* column1 = value1 *ORDER BY* column1 *ASC(DESC);*

This example would extract data which fulfills the condition of the ‘WHERE’ clause and sorted ascending by the chosen column.

Note: It is important to know that SQL statement are not case sensitive, though, it requires single quotes around text values and no quotes around numeric values. [W3scWe]

5.3.2 Insert Into Command

With the ‘INSERT INTO’ command it is possible to insert new data into a table. Below an example of the ‘INSERT INTO’ syntax is illustrated.[W3scIn]

INSERT INTO table(column1, column2, column3,…) *VALUES*
(value1,value2,value3,…);

5.3.3 Update Command

The ‘UPDATE’ command allows to modify data of a table. The ‘WHERE’ clause in the syntax specifies the exact data which should be modified. If there is no ‘WHERE’ clause all data of this table is modified.[W3scUp]

UPDATE table *SET* column1 = value1, column2 = value2,… *WHERE* condition;

5.3.4 DELETE Command

The ‘DELETE’ statement is used to delete data of a table. As in the ‘UPDATE’ command the ‘WHERE’ clause has an important task. By using the ‘WHERE’ clause one can specify which data has to be deleted. Without it all data of the table will be erased. [W3scDe]

DELETE FROM table *WHERE* condition;

6 Graphical User Interface (GUI)

A very important area of the WinterCup program is the graphical user interface, which is intended to facilitate the interaction of the user with the program by visual means. The realization was achieved by the use of JavaFX, the most recent Java standard. JavaFX has replaced the previous technologies like Java Swing and Java AWT. Moreover, JavaFX is intended to build applications for desktops and web browsers on Windows, Macintosh and Linux as well as for Mobile devices. A JavaFX GUI is created as an FXML file. FXML files are based on XML and can be created by code or by using programs such as the Scene Builder which creates a GUI, in this case the FXML file interactively. [OrFX13][FlatFx, p. 3-5][WiJaFx]

The following subchapters explain how to create the GUI for the WinterCup program. It will be explained how it is established by using the Scene Builder, how the automatically generated code looks and thereby important code passages will be highlighted. In addition, important changes in the code (FXML-file) are discussed which are necessary for the use of the GUI with ooRexx in order to run the program flawless. What is more, the GUI can be used by means of ooRexx only by the extension of BSF4ooRexx, since Java classes are used to which access must be made.

6.1 Creating a GUI

As mentioned before, the easiest way to create a GUI with JavaFX is to use the Scene Builder by Gluon. After the installation, the program can be started and a new project can be opened. Now the user has to place an AnchorPane as the first item by just dragging the corresponding item and dropping it in the designated field. Now all possible items can be placed on this AnchorPane. This works with the same principle of dragging and dropping. The following figure illustrates the beginning of a project with an AnchorPane as base, containing a label and a button.

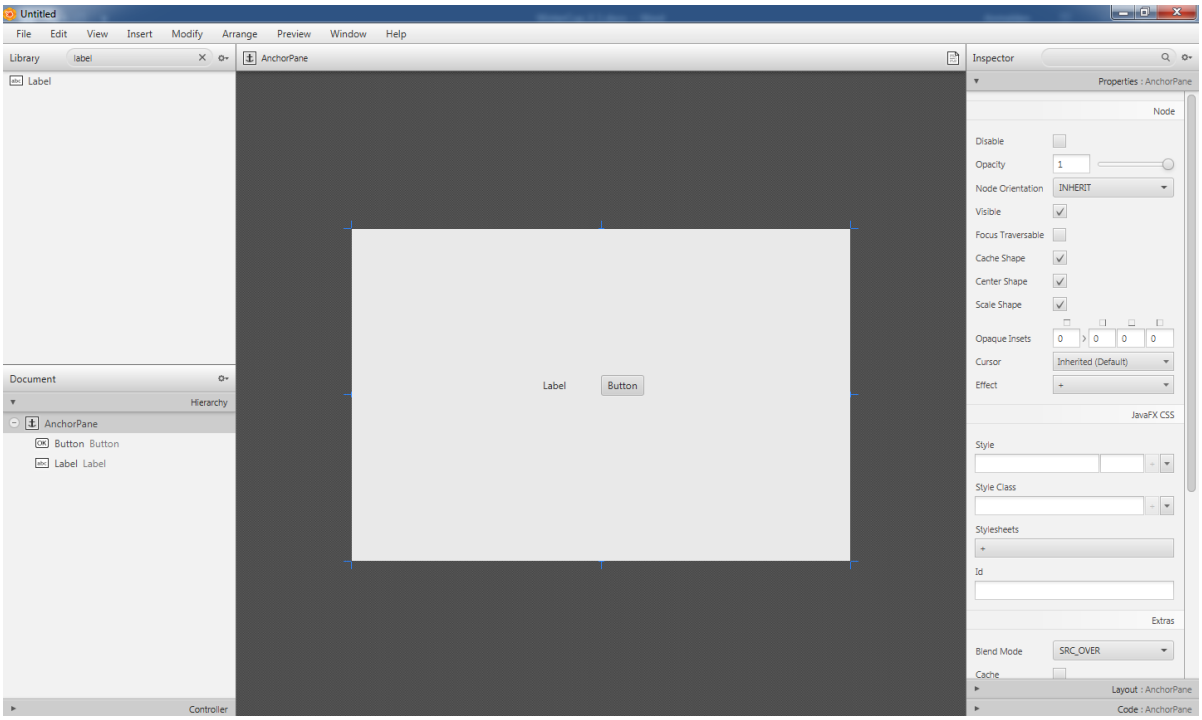


Figure 10: GUI Anchor Pane

Of course, the properties such as size, font, font size, position, name and much more of the items can be changed in the right column under ‘Properties’ and ‘Layout’. Moreover, it is necessary to give all used GUI controls a unique’ fx:id’ attribute so that they can be addressed directly by the ooRexx program. Figure 11 marks the three menu entries needed for the changes described above:

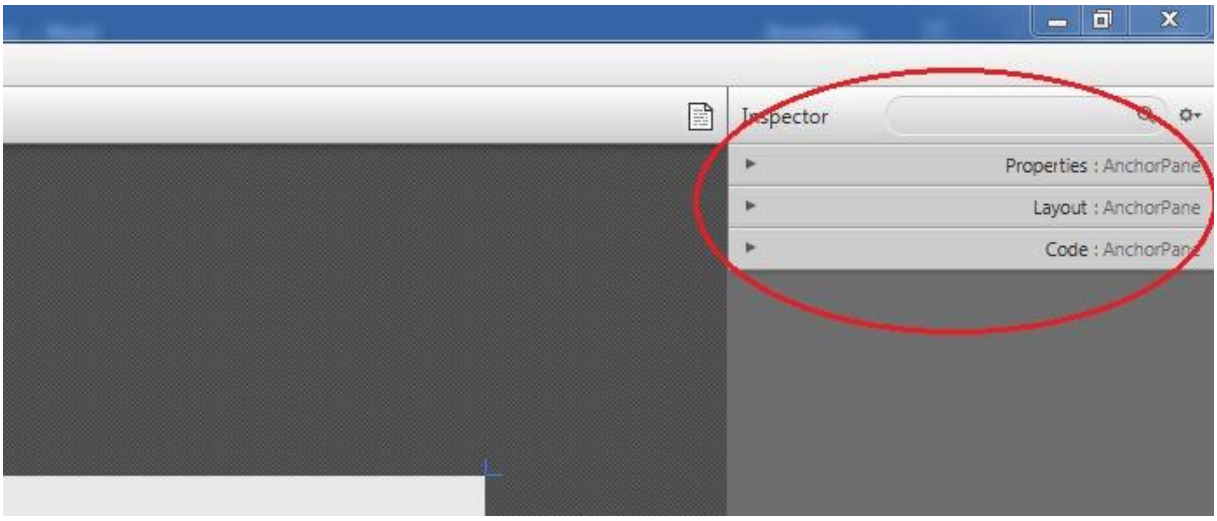


Figure 11: GUI Control Elements Settings

If the now created graphical user interface is saved, an FXML file will be created. In addition, a few changes have to be made in the FXML-File so that the GUI can be accessed and used by ooRexx.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.scene.control.Button?>
4  <?import javafx.scene.control.Label?>
5  <?import javafx.scene.layout.AnchorPane?>
6
7  <?language rexx?>
8
9
10 <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity"
11     minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0"
12         xmlns="http://javafx.com/javafx/8.0.111" xmlns:fx="http://javafx.com/fxml/1">
13
14    <fx:script source = "xxx_controller.rex" />
15
16    <children>
17       <Button fx:id="btn" layoutX="300.0" layoutY="176.0" mnemonicParsing="false" text="Button"
18                                onAction = "slotDir=arg(arg()); call klickButtonAction slotDir;"/>
19       <Label fx:id="lb" layoutX="230.0" layoutY="180.0" text="Label" />
20    </children>
21 </AnchorPane>
```

Snippet 1: GUI FXML-File

Snippet 1 shows the code generated by the Scene Builder. This FXML file defines the created GUI and it items. The extensions are highlighted by the red circles. Circle number one, defines the java script engine – called ‘rexx’- which will be used in case of code that is invoked by an event. The third circle defines such an event. This event is called ‘onAction’ and is started by the users click on the button. Further, this event contains Rexx Code which is then invoked by JavaFX in Rexx as it was defined in circle one. This Rexx code calls a routine with the name ‘klickButtonAction’ and passes the slotDir argument which has all the information needed to get access to the elements used in the GUI. The invoked routine can now be found by JavaFX because of that information given in circle two. Here the source of the Rexx-File including the routine ‘klickButtonAction’ is given.[FlatFx, p.14]

It is also possible to recognize how properties, for example the text, the position, the size, etc., of an element are defined in the FXML file. Note code line 16 for the properties of the button and code line 19 for the properties of the label. Of course, changes of the properties can be made, but they have to be compliant with the XML-notation.

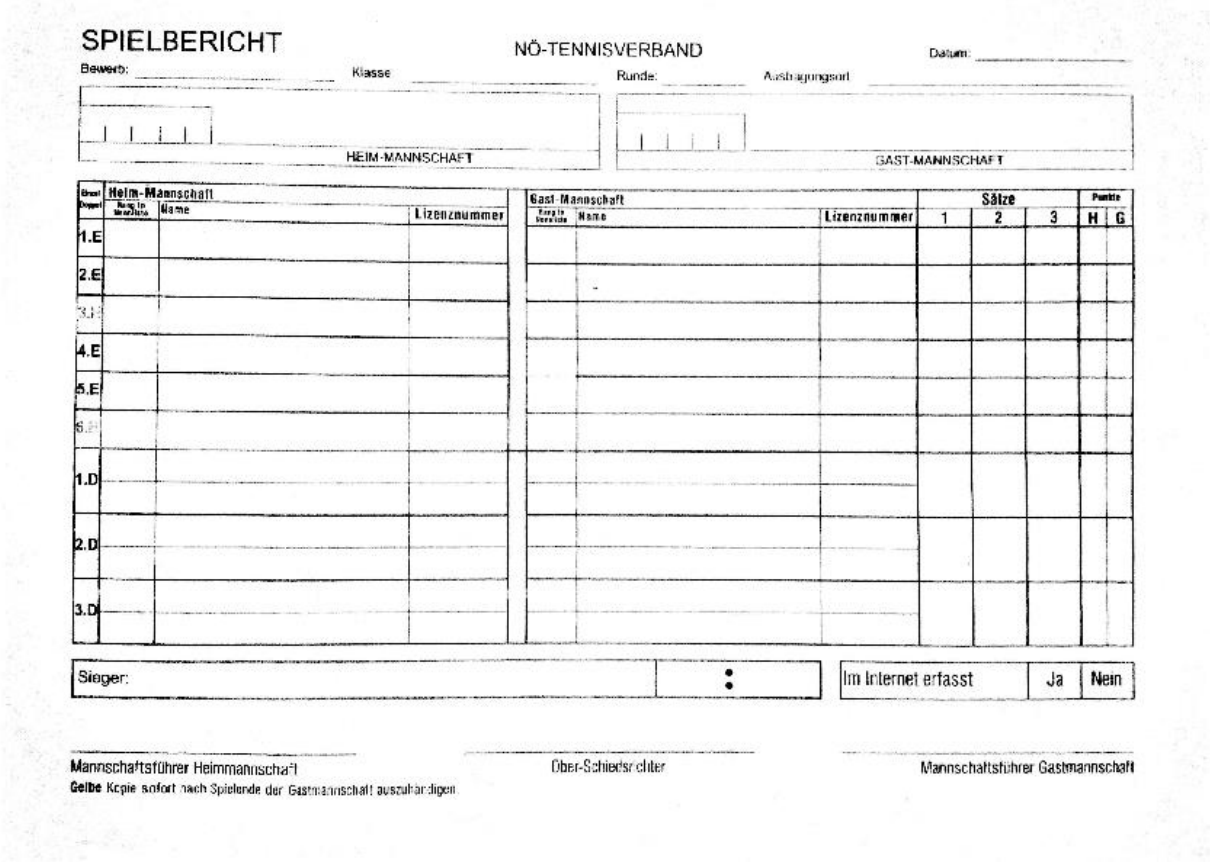
Further information about how to start the GUI with ooRexx and what happens after a routine call is triggered, will be explained in the chapter ‘6.4 GUI with ooRexx’.

6.2 GUI of the Main Program

After explaining how a graphical user interface is built up and prepared to be controlled by ooRexx, this chapter explains how the design of the main program WinterCup came about and how this design was implemented.

6.2.1 Design

The design is based on the official game report of the Lower Austrian Tennis Association (NÖTV- Niederösterreichischer Tennisverband), since it is also used in the summer championship. The majority of the players who compete in the winter cup have been playing the summer championship for years and are used to this report. Thus, the conversion to a digital platform can be realized more easily for the user.


SPIELBERICHT

NÖ-TENNISVERBAND

Bewerb: Klasse Runde: Austragungsort Datum:

HEIM-MANNSCHAFT

GAST-MANNSCHAFT

	Heim-Mannschaft Rang lt. Meldliste	Name	Lizenznummer	Gast-Mannschaft Rang lt. Meldliste	Name	Lizenznummer	Sätze 1	2	3	Punkte H	G
1.E											
2.E											
3.E											
4.E											
5.E											
6.E											
1.D											
2.D											
3.D											

Sieger: : Im Internet erfasst Ja Nein

Mannschaftsführer Heimmannschaft Ober-Schiedsrichter Mannschaftsführer Gastmannschaft

Gelbe Kopie sofort nach Spielende der Gastmannschaft auszuhändigen

Figure 12: Official Scoresheet of the NÖTV

6.2.2 Implementation

The GUI of the main program consists of two windows reside under one parent window. This method is called multi document interface(MDI) and is also used in browsers. There are several advantages why MDI is used. For the WinterCup GUI the main reasons were the sharing of one menu bar as well as the efficient use of screen space. The two windows are consolidated under a tab pane but on the other side split by tabs.[WiMuDI]

The first window, called ‘Spielbericht’(English: *Scoresheet*), serves the recording of the games. The GUI of Window 1 is described in the following.(The tasks of the windows are explained in chapters 7 and 8 in more detail.)

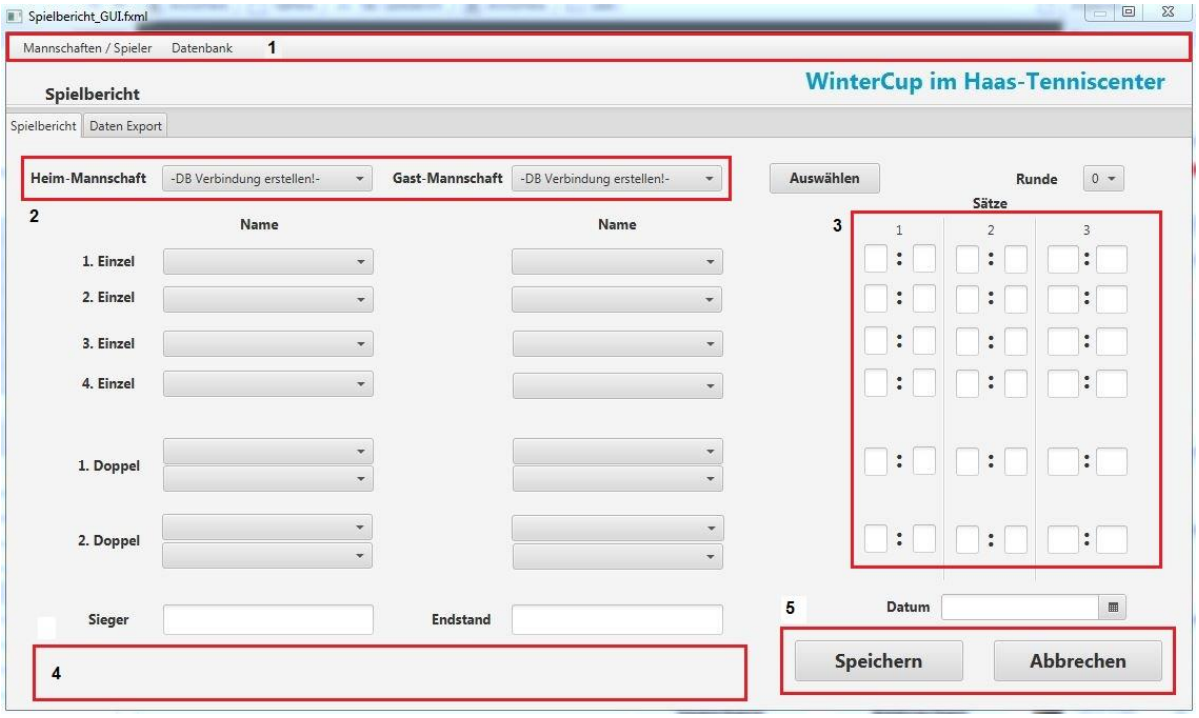


Figure 13: Main Program ‘Scoresheet’

At the top of the window is a Menu Bar (mark 1) which has two sub-menus. These submenus have the same function as buttons. If they are clicked the ‘onAction’ event of this element is invoked and a routine is started.

```
<MenuBar fx:id="menuBar" layoutY="2.0" prefHeight="25.0" prefWidth="1126.0">
  <menus>
    <Menu mnemonicParsing="false" text="Mannschaften / Spieler">
      <items>
        <MenuItem fx:id="menuBar_Team_Player" mnemonicParsing="false"
          onAction="slotDir=arg(arg());call klickOpenTeamPlayerAction slotDir;"
                                                    text="Bearbeiten / Hinzufügen"/>
      </items>
    </Menu>
    <Menu mnemonicParsing="false" text="Datenbank">
      <items>
        <MenuItem fx:id="menuBar_DBConnection" mnemonicParsing="false"
          onAction="slotDir=arg(arg());call klickConnectDBAction slotDir;"
                                                    text="Verbindung herstellen"/>
      </items>
    </Menu>
  </menus>
</MenuBar>
```

Snippet 2: FXML-File 'onAction' Event

The figure above highlights the 'onAction' events of the control elements button and label. Each of them calls a different routine which processes different tasks.

Mark 2 highlights two Choice Boxes which have a predefined value. In order to give them a predefined value, some changes must be made in the FXML file. First of all, the classes 'String' and 'FXCollection' have to be integrated.

```
<?import java.lang.String?>
<?import javafx.collections.FXCollections?>
```

Snippet 3: FXML-File Loading Classes

Now changes at the Choice Box FXML code have to be made. First of all, a new instance of the 'observableArrayList' must be created, for the Choice Box, which is a method of the 'FXCollections' class. This 'observableArrayList' is now filled with data. At this point, the Choice Box can also be filled with several data. However, for the WinterCup program only one data record is added to the Choice Box. In this case the data is a string which is the reason for loading the 'String' class to the FXML file. Besides that, the desired data record has to be preselected explicitly otherwise the Choice Box will not select any data record and the Choice Box would be empty. Just by clicking on the Choice Box the data will show up. The preselection is done by setting the property value of the Choice Box to the desired preselection. This means that the only available data record of the 'observableArrayList' is set as a value of the property. The next snippet shows the realization at the WinterCup GUI. The preselected text, "-DB Verbindung erstellen!-", of the Choice Box informs the user to establish a database connection in order to get the Choice Box filled with relevant data. All the other Choice Boxes will get filled automatically by the main program afterwards.

```
<ChoiceBox fx:id="cb_homeTeam" layoutX="155.0" layoutY="26.0" prefHeight="25.0"
                               prefWidth="200.0" value="-DB Verbindung erstellen!-">
<items>
    <FXCollections fx:factory="observableArrayList">
        <String fx:value="-DB Verbindung erstellen!-" />
    </FXCollections>
</items>
</ChoiceBox>
```

Snippet 4: FXML-File ChoiceBox

Looking back at Figure 13, the text fields, which are located on the right side of the program, are the input elements for the score of the game operated by the user. At this point it is not intended to explain how the input is read in concrete, but the purpose is to highlight that this Text Fields have an event, called ‘onKeyReleased’, which is triggered as soon as a key is released. A routine is then called which starts a query. This query is used to check the entered scores, of the first, second and third set. Snippet 5 illustrates the events just mentioned. It can be seen that each text field has an own routine to invoke.

```
<TextField fx:id="tf_1s1sh" layoutX="822.0" layoutY="101.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1s1sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1s1sa" layoutX="868.0" layoutY="101.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1s1sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1s2sh" layoutX="909.0" layoutY="102.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1s2sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1s2sa" layoutX="955.0" layoutY="102.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1s2sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1s3sh" layoutX="996.0" layoutY="102.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1s3sh slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_1s3sa" layoutX="1042.0" layoutY="102.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1s3sa slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_2s1sh" layoutX="822.0" layoutY="139.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2s1sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2s1sa" layoutX="868.0" layoutY="139.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2s1sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2s2sh" layoutX="909.0" layoutY="140.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2s2sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2s2sa" layoutX="955.0" layoutY="140.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2s2sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2s3sh" layoutX="996.0" layoutY="140.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2s3sh slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_2s3sa" layoutX="1042.0" layoutY="140.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2s3sa slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_3s1sh" layoutX="822.0" layoutY="179.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_3s1sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_3s1sa" layoutX="868.0" layoutY="179.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_3s1sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_3s2sh" layoutX="909.0" layoutY="180.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_3s2sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_3s2sa" layoutX="955.0" layoutY="180.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_3s2sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_3s3sh" layoutX="996.0" layoutY="180.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_3s3sh slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_3s3sa" layoutX="1042.0" layoutY="180.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_3s3sa slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_4s1sh" layoutX="822.0" layoutY="219.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_4s1sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_4s1sa" layoutX="868.0" layoutY="219.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_4s1sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_4s2sh" layoutX="909.0" layoutY="220.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_4s2sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_4s2sa" layoutX="955.0" layoutY="220.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_4s2sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_4s3sh" layoutX="996.0" layoutY="220.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_4s3sh slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_4s3sa" layoutX="1042.0" layoutY="220.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_4s3sa slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_1d1sh" layoutX="822.0" layoutY="294.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1d1sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1d1sa" layoutX="868.0" layoutY="293.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1d1sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1d2sh" layoutX="909.0" layoutY="294.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1d2sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1d2sa" layoutX="955.0" layoutY="294.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1d2sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_1d3sh" layoutX="996.0" layoutY="294.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1d3sh slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_1d3sa" layoutX="1042.0" layoutY="294.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_1d3sa slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_2d1sh" layoutX="822.0" layoutY="367.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2d1sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2d1sa" layoutX="868.0" layoutY="367.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2d1sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2d2sh" layoutX="909.0" layoutY="368.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2d2sh slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2d2sa" layoutX="955.0" layoutY="368.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2d2sa slotDir;" prefHeight="27.0" prefWidth="22.0" />
<TextField fx:id="tf_2d3sh" layoutX="996.0" layoutY="368.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2d3sh slotDir;" prefHeight="27.0" prefWidth="30.0" />
<TextField fx:id="tf_2d3sa" layoutX="1042.0" layoutY="368.0" onKeyReleased="slotDir=arg(arg());call KeyReleaseCheck_tf_2d3sa slotDir;" prefHeight="27.0" prefWidth="30.0" />
```

Snippet 5: FXML-File TextField 'onAction' Event

Mark 4 – Figure 13 - highlights a label, which cannot be seen as it has no text at the start. This label is considered to issue error messages during run time. Figure 14 shows an error example. This error will pop up if the user has clicked the button without establishing a database connection beforehand.

Figure 14: Error Message

The last highlighted area of Figure 13 marks two buttons. These buttons have an ordinary ‘onAction’ event as the other buttons. They are highlighted because they are invisible when the program is started and get visible after the ‘Auswählen’ button is triggered. This characteristic can be changed in the properties ‘Setting’ of the button or in the FXML file by adding the attribute *visible* = “false” to the buttons characteristics.

The second window, called ‘Daten Export’(English: *Data Export*), has the purpose of giving the user the possibility to export important data of the winter cup.

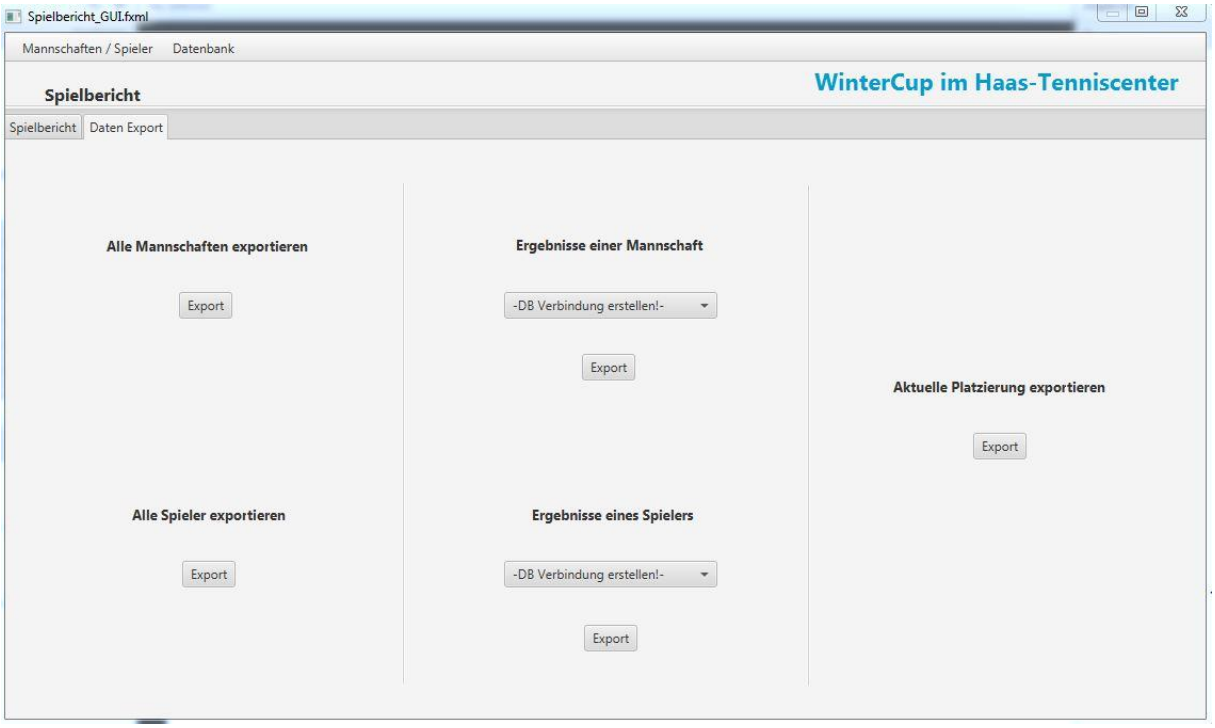


Figure 15: Main Programm 'Data Export'

All the elements have been set up same as the elements of the ‘Spielbericht’. The buttons also have the ‘onAction’ event which will invoke an own routine.

6.3 GUI of the Second Program

With the second program, the user can add, edit and delete players and teams. The reason why a separate program was created is that the normal usage of this program is only at the beginning of the winter cup season. All teams and players should register

before the winter cup starts because of organizational reasons. However, changes are also allowed during the season. Additionally, a clean separation between the initial administration and the maintenance of an ongoing tournament is tried by splitting the tasks in two separate programs. Though, they are still connected as the second program can only be started by the main program under the submenu ‘Mannschaften / Spieler’ of the menu bar.

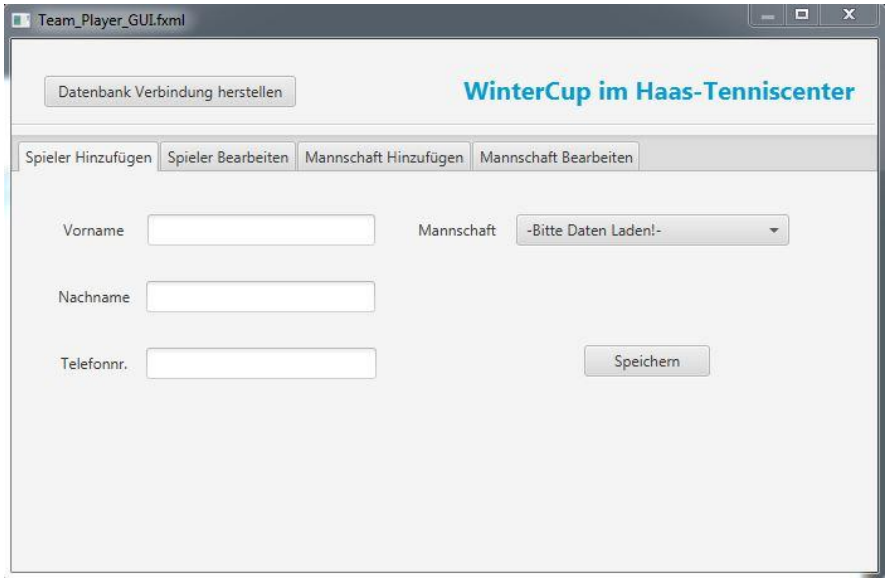


Figure 16: Secondary Program 'Add Player'

The design of this program looks similar to the main program. Also, here the multi document interface was used in order to manage the efficient use of screen space. What applies to all tabs is that they all have a label at the bottom of the window so that error messages can be issued while run time. What is more, prefilled and preselected Choice Boxes were subjected to the same settings and extensions as the Choices Boxes of the Main Program GUI. Further, all buttons have the ‘onAction’ event triggering a routine. The next figure illustrates the remaining three tabs. Areas marked with a red square, highlight elements that are invisible at the start of the program. However, these elements will get visible after pushing the ‘Bearbeiten’ Button in Tab ‘Spieler Bearbeiten’ and Tab ‘Mannschaften Bearbeiten’. In contrast, the Button ‘Datenbank Verbindung Herstellen’ gets invisible after been clicked by the user.

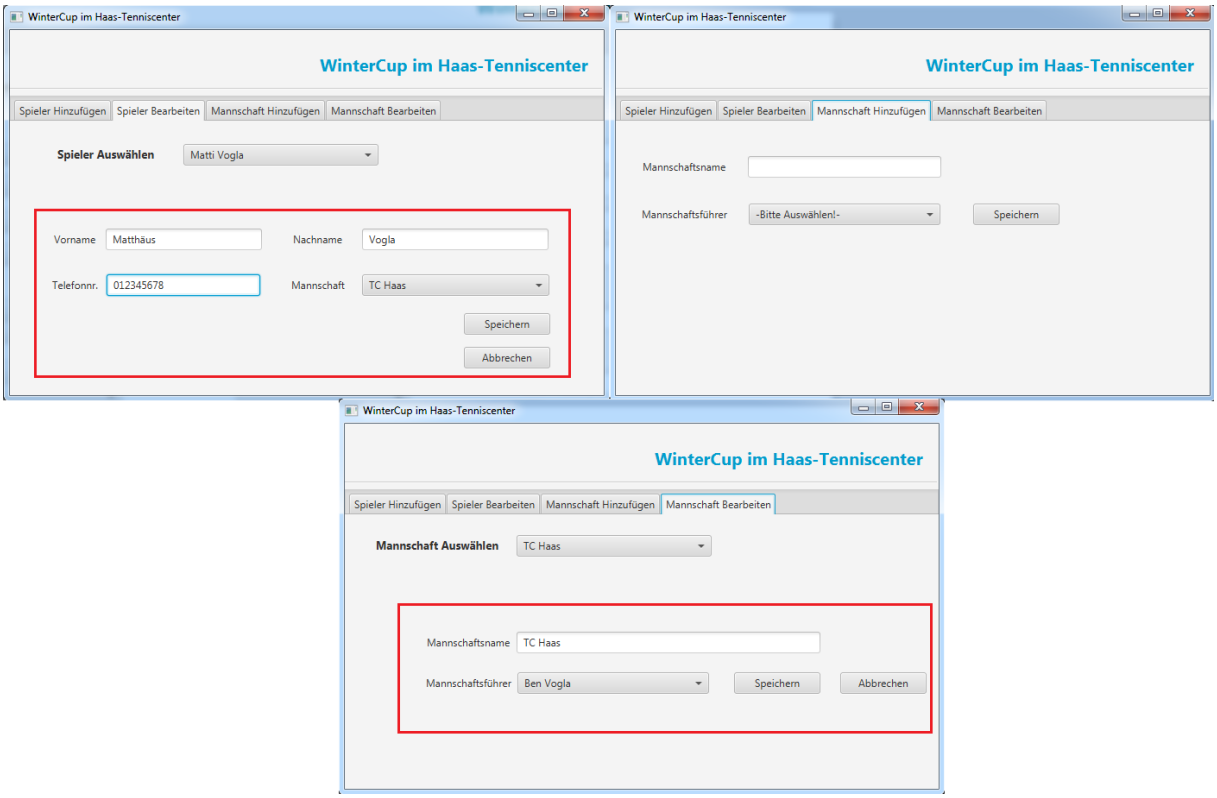


Figure 17: Secondray Programm Remaining Tabs

Comparing Figure 16 and Figure 17 it can be seen that the Button ‘Datenbank Verbindung Herstellen’ is invisible and therefore cannot be clicked anymore by the user.

6.4 GUI with ooRexx

For the sake of completeness, the launch of a GUI using ooRexx is explained in this chapter.

First of all, there are two ooRexx files created, the one launches the GUI and the other contains the actual program and the routines which were previously mentioned.

6.4.1 Launch of the GUI

The JavaFX application builds a window, also called stage, which contains a GUI container also called scene. The GUI container, in this case, is the Anchor Pane which was placed at the very beginning of this chapter. In this container all desired controls, such as buttons, labels, etc, are placed.

The GUI can only be started by ooRexx when using the bridge of BSF4ooRexx because of the fact that the Java class ‘javafx.application.Application’ is required. An instance of the Java class javafx.application.Application is created which calls the start method of a Rexx class via its own launch method. By calling the start method, the stage and the corresponding scene are set up.

```
1  SAY "PROGRAM START!"
2  rexxHandler=.rxDocHandler~new
3
4  rxApp=BsfCreateRexxProxy(rexxHandler,,"javafx.application.Application")
5
6  rxApp~launch(rxApp~getClass, .nil)
7
8  ::requires "BSF.CLS"
9
10 ::class RxDocHandler
11
12 ::method start
13   use arg stage
14
15   stage~setTitle("Spielbericht WinterCup im Haas-Tenniscenter")
16
17   rootDocUrl=.bsf~new("java.net.URL", "file:Spielbericht_GUI.fxml")
18
19   root=bsf.loadClass("javafx.fxml.FXMLLoader")~load( rootDocUrl  )
20   scene=.bsf~new("javafx.scene.Scene", root)
21   stage~setScene(scene)
22 _ stage~show
```

Snippet 6: Launch of the GUI

The Rexx class which includes the start method is called ‘RxDocHandler’. If the Java class is instantiated, using the ‘BsfCreateRexxProxy()’, the launch method can be called, which then calls the start method of the ‘RxDocHandler’ class. Now the desired scene is inserted into the stage (window). The scene is the pre-built graphical user interface.

6.4.2 Actual Code

Now that the GUI can be started, a second Rexx-file is required which contains the actual code. In this file, all routines are defined which are called by the events of the GUI controls. This file is already defined in the FXML-file of the GUI. (see Snippet 1)

```
:: routine klickRankingExportAction public
use arg slotDir
/*@get("lb_Info_Export")*/

        lb_Info_Export~Text = ""
        "start /D" directory() "rexxj.cmd Results.rex"
```

Snippet 7: Actual Code Routine

The first line of code, starting with the double points, defines a routine called ‘klickRankingExportAction’. This routine is started by the Rexx code defined with the ‘onAction’ event of a button. In the second line, the ‘slotDir’ argument which is supplied by the event is initialized. The ‘slotDir’ argument is very important as it enables access to the control elements of the GUI via their ‘fx:id’. The instruction /*@get(“lb_Info_Export”)*/ makes the label with the fx:id ‘lb_Info_Export’ available and initializes it as a Rexx variable named ’lb_Info_Export’. If the statement looks strange, the reason might be that this statement is written as a Rexx script annotation. The reason for this is the shorter spelling-syntactic sugar. The following statement is also possible, but is much longer and comes to the same result.

```
lb_Info_Export = slotDir~scriptContext~getAttribute("lb_Info_Export")
```

Snippet 8: Snippet Rexx Annotation

Now the Rexx variable 'lb_Info_Export' can be used. In this case, for example, the text (value) of the label is removed.

7 Development of the Main Program

After introducing the design of the database and the GUI in the previous chapters, now the code of the WinterCup program can be explained. At the beginning of this chapter a brief overview of the entire software component is provided. Later on, the focus is placed on more important parts of the implementation and some sections are described in more detail, This gives the reader an understanding of how WinterCup works. In order to get a better and easier overview, only the main program is explained in this chapter, continuing the next chapter with the second program.

7.1 Overview

The task of the main program is to keep encounters between teams during the ongoing competition for the record, evaluate results and create a ranking which then can be exported by the user. Furthermore, it is possible to export not only the current ranking but also a full list of all teams and players as well as encounters of selected teams and players.

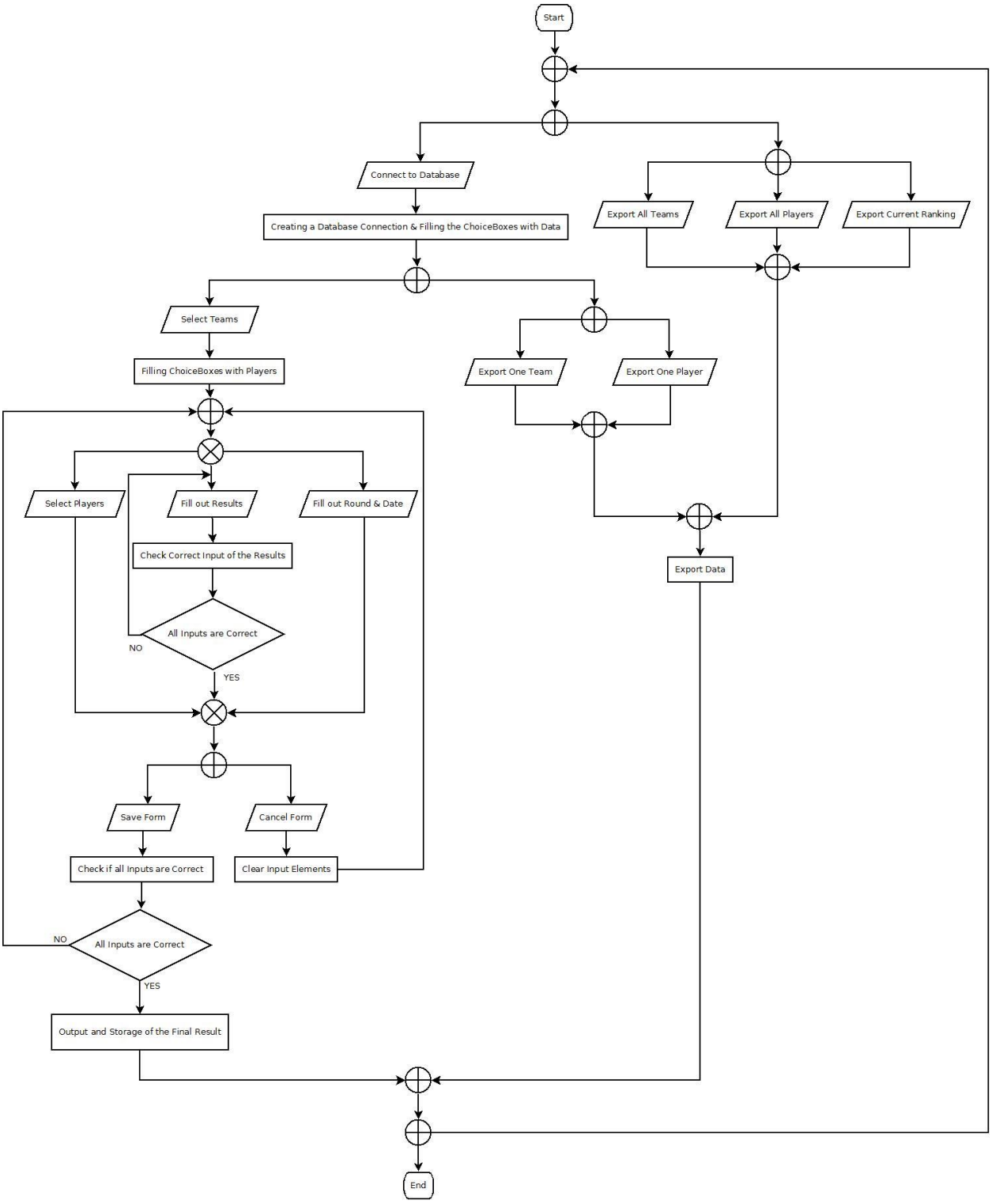
‘Flow-Chart 1’ is intended to visualize the flow of the main program in order to gain a better understanding of it. After the user has started the program, the GUI opens with the scoresheet (‘Spielbericht’ Tab) as an opening screen. Without being connected to the database, the user can only open the ‘Data Export’ Tab where three options are available.

- Export a complete team list
- Export a complete player list,
- Export the current ranking

In order to use all the other applications offered by WinterCup, the user must establish a database connection first. Once a database connection is created, all applications of WinterCup can be used. Now the user can select the two additional data export options.

- Export results of a player
- Export results of a team

Returning back to the tab ‘Spielbericht’, the user can now select the teams that compete against each other. After selecting the competing teams, the user has to fill out all empty fields. By this the selection of the players, the selection of the round, the entry of the date and the entry of the game results is meant. When the user has filled out everything, the form can be saved. By saving the form all information are stored into the database and moreover all fields are cleared so that the user can start a new entry. By clicking the ‘Abbrechen’ (English: cancel) button instead of the ‘Speichern’ (English: save) button, all fields are cleared as well and the user can start the input again by selecting the competing teams. In this case, nothing will be stored in the database.



Flow Chart 1: Main Program

7.2 Database Connection

As already mentioned in the upper section, WinterCup requires a connection to the WinterCup database in order to be able to use all applications of the program. The connection is performed by using the MySQL Connector for Java. The API has the

advantage that it makes the use of the database easier and faster to implement. Before the classes and methods of the MySQL Connector can be used, the installation of the MySQL Connector has to be done.(It is referred to chapter ‘4.1.5 MySQL Connector API’)

```
/***MySQL Conntection***/
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
team = statement~executeQuery(sqlkomm)
```

Snippet 9: MySQL Database Connection

Snippet 9illustrates how a database connection is established with ooRexx and the MySQL connector. First of all, the Java class driver manager (fully qualified name: java.sql.DriverManager) is loaded using BSF4ooRexx. This class is required in order to establish a database connection. The next step tells the driver manager which driver to try while creating a database connection. This is done by loading the driver class and creating a new instance of it. Now the database connection can be created via the 'getConnection' method of the driver manager. Moreover, the URL of the database, the user and the password have to be passed to the method to finally establish a connection. Important is the correct separation of the parameters using the question mark symbol.[MyDrMa]

Once the database connection has been established, SQL statements can be executed. In ooRexx SQL queries can be performed by using a statement object that returns a result object with a status code and a set of result records on execution of a certain method. In detail the sequence looks like this: The ‘createStatement’ method is called on the Connection object in order to create an instance of the Statement class. When this instance is created, a 'SELECT' SQL query, like illustrated in Snippet 9, can be executed using the 'executeQuery' method of this instance. In the example above, all entries of the ‘Team’-table sorted in ascending order are retrieved and saved in the result set class named ‘team’. (It should be noted that if an ‘INSERT’, ‘UPDATE’ or ‘DELETE’ SQL query is execute the method called ‘executeUpdate’ instead of ‘executeQuery’ is needed.)[MyStOb]

Nevertheless, the respective interface function then returns the requested data from the database that can be read out via an iterator. The implementation of this task will

be explained in detail in the upcoming chapter.Finally, all database objects should be closed in the reverse order in which they were opened.

```
team~Close()
statement~Close()
conn~Close()
```

Snippet 10: Closing Database Objects

7.3 Filling the Choice Boxes with Relevant Data

After a database connection has been established and an SQL query has been sent to the database, the returned result set can be used to fill control elements of the WinterCup program with relevant data. Thus enabling functions that were not possible to use before. In particular, the Choice Boxes of the team selection on the ‘Spielbericht’ Tab and the Choice Boxes of the team and of the players in the ‘Data Export’ Tab, are meant. The following snippet illustrates the routine that is invoked when the user clicks Connect to the Database on the main program menu.

```
1287 ::routine DBConn
1288 use arg slotDir
1289
1290 /*@get("cb_homeTeam")*/
1291 /*@get("cb_awayTeam")*/
1292
1293 /*@get("cb_Team_Export")*/
1294 /*@get("cb_Player_Export")*/
1295 /*@get("dp_date")*/
1296 /*@get("cb_round")*/
1297 /*@get("lb_Info_Export")*/
1298
1299 cb_awayTeam~getItems~Clear()
1300 cb_homeTeam~getItems~Clear()
1301 cb_Team_Export~getItems~Clear()
1302 cb_Player_Export~getItems~Clear()
1303 cb_round~getItems~Clear()
1304
1305 lb_Info_Export~Text = ""
1306
1307 cb_homeTeam~getItems~Add("Bitte Mannschaft ausw"'E4'x"hlen!")
1308 cb_awayTeam~getItems~Add("Bitte Mannschaft ausw"'E4'x"hlen!")
1309 cb_Team_Export~getItems~Add("Bitte Mannschaft ausw"'E4'x"hlen!")
1310 cb_Player_Export~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
```

Snippet 11: Database Connection Routine – Choice Box properties

First of all, the required control elements are initialized as explained in chapter ‘6.4.2 Actual Code’. Now that the access to these elements is enabled, the predefined text of the Choice Boxes is deleted. In the next step, new text is added to the Choice Boxes. In general, the choice box object internally maintains a list that keeps record of the items which are displayed. So to add a value to the pool of strings that is displayed by the choice box, one simply has to add another string into the value list. To do this, the

choice box objects provides a getter method that returns a reference to the internally used list to which a value can then simply be appended.

```
1312 /***MySQL Conntection***/
1313 driverMgr = bsf.loadClass("java.sql.DriverManager")
1314 bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
1315 conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?"-
1316 "user=4440_3&password=Haas-Tenniscenter")
1317
1318 statement = conn~createStatement
1319 sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
1320 team = statement~executeQuery(sqlkomm)
1321
1322 check = .false
1323 check = team~First
1324 mannschaftenIndex = 0
1325 DO WHILE check = .true
1326         IF (team~getString("TeamName") = "NO TEAM") THEN check = team~Next
1327         ELSE DO
1328                 cb_homeTeam~getItems~Add(team~getString("TeamName"))
1329                 cb_awayTeam~getItems~Add(team~getString("TeamName"))
1330                 cb_Team_Export~getItems~Add(team~getString("TeamName"))
1331                 mannschaftenIndex = mannschaftenIndex + 1
1332                 check = team~Next
1333         END
1334 END
1335 sqlkomm = "SELECT * FROM Players ORDER BY Nachname ASC"
1336 player = statement~executeQuery(sqlkomm)
1337
1338 check = player~First
1339 DO WHILE check = .true
1340                 Vorname = player~getString("Vorname")
1341                 Nachname = player~getString("Nachname")
1342                 Name = Vorname Nachname
1343                 cb_Player_Export~getItems~Add(Name)
1344                 check = player~Next
1345 END
1346 player~Close()
1347 team~Close()
1348 statement~Close()
1349 conn~Close()
1350 IF (mannschaftenIndex > 0) THEN DO
1351         DO i = 1 TO mannschaftenIndex -1
1352                 cb_round~getItems~Add(i)
1353         END
1354         cb_round~getSelectionModel()~Select(0)
1355 END
1356 cb_homeTeam~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
1357 cb_awayTeam~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
1358 cb_Team_Export~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
1359 cb_Player_Export~getSelectionModel()~Select("Bitte Spieler ausw"'E4'x"hlen!")
```

Snippet 12: Database Connection Routine

After initializing the connection, as described previously, the example in Snippet 12 line 1320 shows that a query returns a result set that contains the actual data. The result of an executed SQL statement is provided by the database driver as an iterator object, known as the result set, that allows to iterate through the returned records, by a simple ‘First’ and ‘Next’ interface. While the ‘First’ method may be used to test whether the iterator has entirely processed the result set, the ‘Next’ method returns the actual value of the current iterator position. This process is controlled by a while-loop which continues as long as the value of the iterator object is true.

In the next step, the Choice Boxes are filled with the team names, except of the team called 'No Team', which is just a virtual team that contains players without an actual

team and hence should not be selectable. The rows of the table are processed by the iterator, with the ‘Next’ method as mentioned before. The value of a certain record attribute can then by read out by the different getter methods for specific datatypes (e.g. getString). By invoking the method 'getString' of the ‘team’ result set and passing the exact name of the column needed, the value of a cell is read out. The result can now be used to fill all Choice Boxes. Using the same principle, it can be seen that in the next program sequence a Choice Box on the ‘Data Export’ tab is filled with the players’ names.

After all objects of the database have been closed, the next instruction is to fill a Choice Box (‘cb_round’) with the maximum number of possible rounds. This is calculated by: <number of teams> - 1, since one cannot play against oneself. Finally, the last program sequence, assigns to all just filled out Choice Boxes a start value.

The next chapter describes the further course of the main task of the program whereby the data export is explained further in chapter ‘7.6 Data Export’.

Note: Since WinterCup has been programmed for the German-speaking region, all texts are written in German. As ooRexx does not recognize umlauts like ‘Ä’, ‘ä’, ‘Ö’, ‘ö’, and ‘Ü’, ‘ü’, the hexadecimal notation is used. This means for example, that in the word ‘auswählen’ the ‘ä’ has to be written by using this hexadecimal code "'E4'x". If the program is started now the letter is displayed correctly.

7.4 Input of Data by the User

If the user selects two teams, which compete against each other, the additional (empty) Choice Boxes of this form are filled with players’ names. The user must now select the players who play against each other. The operation of filling a Choice Box with data, has already been explained above and will not be repeated at this point. However, it should be noted that a broken line separates the members of the selected teams from all other players so that they can be recognized immediately. The next figure gives an example of how this separation looks like.

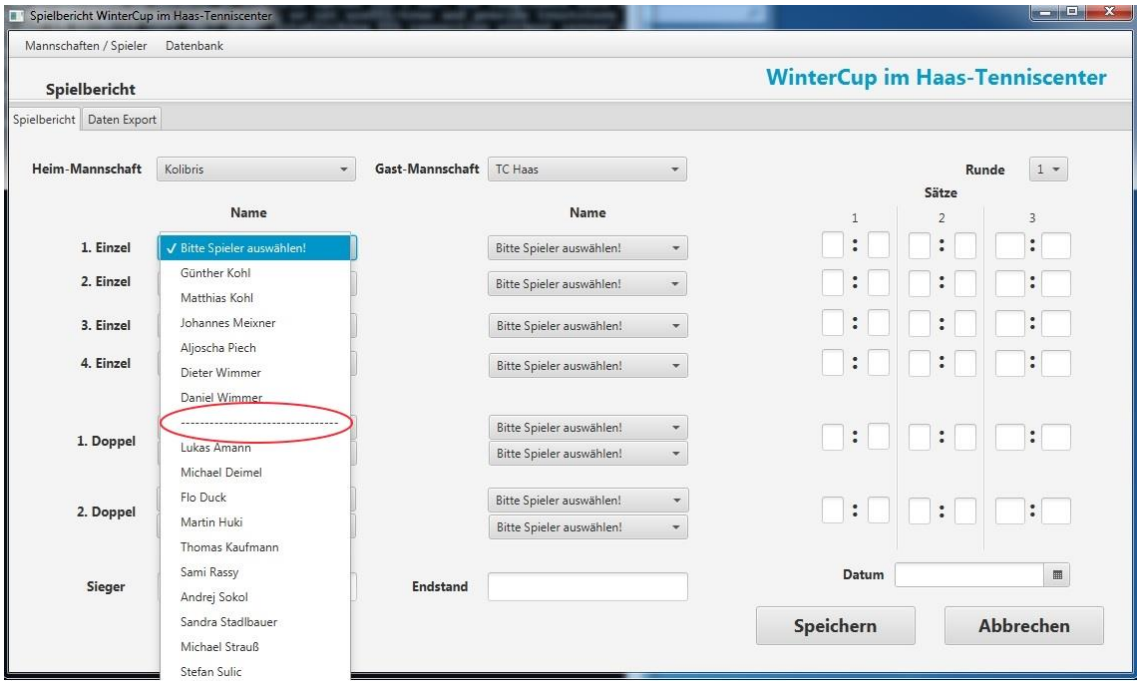


Figure 18: Choice Box Separation Line

Once all players have been selected, the user can now enter the results of the played games. It should be emphasized that all input fields (text fields), have an event that is triggered when a key is released which then leads to a routine call in the program. Snippet 13 illustrates two key release routines and their instructions.

```
1364 ::routine CheckNum public
1365 use arg Num
1366 Check = .false
1367
1368 IF (Num >= 0 & Num <= 7) THEN Check = .true
1369 ELSE DO
1370         Check = .false
1371 END
1372 return Check
1373 /*****************************************/
1374 ::routine CheckNum3Set public
1375 use arg Num
1376 Check = .false
1377
1378 IF (Num >= 0 & Num <= 20) THEN Check = .true
1379 ELSE DO
1380         Check = .false
1381 END
1382 return Check
1383 /*****************************************/
1384 ::routine KeyReleaseCheck_tf_1s1sh public
1385 use arg slotDir
1386 Check = .false
1387
1388 /*@get("tf_1s1sh")*/
1389
1390 Num = tf_1s1sh~Text
1391 Check = CheckNum(Num)
1392
1393 IF Check = .false THEN tf_1s1sh~Text = ""
1394 /*****************************************/
1395 ::routine KeyReleaseCheck_tf_1s1sa public
1396 use arg slotDir
1397 Check = .false
1398
1399 /*@get("tf_1s1sa")*/
1400
1401 Num = tf_1s1sa~Text
1402 Check = CheckNum(Num)
1403
1404 IF Check = .false THEN tf_1s1sa~Text = ""
```

Snippet 13: Routines called by a Key Release Event

All routines called by the release of a key, the routine ‘CheckNum’ to verify whether the entered number is less than or equal to 7. The value 7 results from the tennis rules whereby a set is finished with maximum 7 games. In the 3rd set a match-tiebreak is played, which is ended by reaching a minimum of 10 points and is open to the top. However, the query is limited to a maximum of 20 points, since it is never experienced that 20 points are exceeded.(The set is normally ended by a two points difference before the 20 points are reached). If any input exceeds the limitations the whole input is deleted and the user has to reenter the score.

Note that the names of the routines are not arbitrary, follow a certain naming scheme For example, tf_1s1h stands for text field with the score of the first match and the first set of the home team.

7.5 Save Form

If all input fields have been filled out by the user, the form could be saved. However, before the data can be stored in the database, the input data has to be validated first. First, it is checked whether the selection of the players did not result in a double player selection. Though, the singles and doubles must be considered separately. The next figure illustrates this scenario.

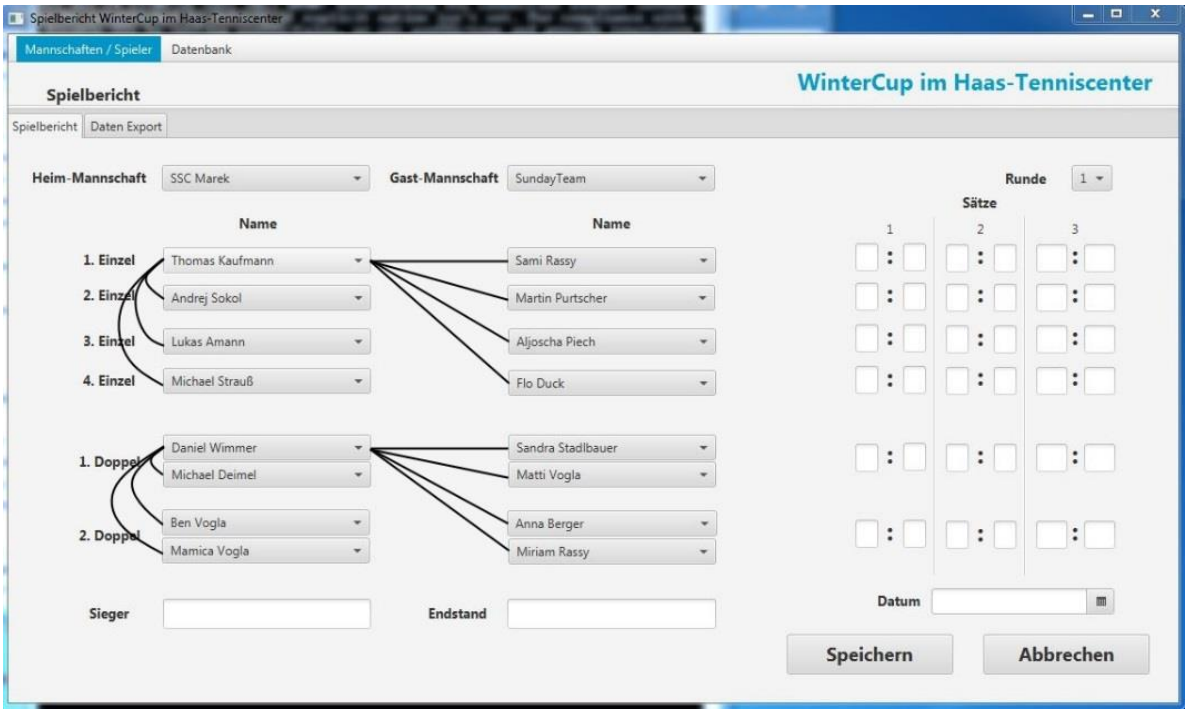


Figure 19: Double Selection of a Player

It is important to note that this example illustrates only the first player, nevertheless, all players must be checked this way. An excerpt of the code can be found in the appendix (Controller_Spielbericht.rex).

If there is no double selection of a player then the code continues with the further control of the entered results. The routine ‘CheckInput’ is invoked which checks whether the entered scores are correct. Snippet 14 shows a section of this routine which deals with the control of the first match. The other games are controlled according to the same principle, but with different input fields.

```
712         /*************************************************** 1 SET ***************************************************************/
713         pointsHomeSet = 0; pointsAwaySet = 0; Mistake = .false; Mistake_1_Set = .false
714         IF (V1s1sh = "" | V1s1sa = "" | V1s2sh = "" | V1s2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte Felder des 1.Satzes ausf"'FC'x"llen! | "
715         ELSE IF (V1s1sh = V1s1sa | V1s2sh = V1s2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des 1.Satzes sind falsch! | "
716         ELSE DO
717                 Points = CheckCorrectNumberInput(V1s1sh V1s1sa)
718                 parse var Points pointsHome pointsAway Mistake
719                 IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 1.Satz! | "
720                 ELSE DO
721                         pointsHomeSet = pointsHomeSet + pointsHome
722                         pointsAwaySet = pointsAwaySet + pointsAway
723                         Points = CheckCorrectNumberInput(V1s2sh V1s2sa)
724                         parse var Points pointsHome pointsAway Mistake
725                         IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 1.Satz! | "
726                         ELSE DO
727                                 pointsHomeSet = pointsHomeSet + pointsHome
728                                 pointsAwaySet = pointsAwaySet + pointsAway
729                                 IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
730                                         IF(V1s3sh = "" | V1s3sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte alle Felder des 1.Satzes ausf"'FC'x"llen! | "
731                                         ELSE IF (V1s3sh = V1s3sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des 1.Satzes sind falsch! | "
732                                         ELSE DO
733                                                 Points = CheckCorrectNumberInputTieBreak(V1s3sh V1s3sa)
734                                                 parse var Points pointsHome pointsAway Mistake
735
736                                                 IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 1.Satz! | "
737                                                 ELSE DO
738                                                         pointsHomeSet = pointsHomeSet + pointsHome
739                                                         pointsAwaySet = pointsAwaySet + pointsAway
740                                                 END
741                                         END
742                                 END
743                                 ELSE IF (V1s3sh <> "" | V1s3sa <> "") THEN DO
744                                         lb_Info~Text = lb_Info~Text"Eingabefehler 1.Spiel 3.Satz! | "
745                                         Mistake_1_Set = .true
746                                 END
747                         END
748                         IF(pointsHomeSet = 2 & Mistake_1_Set = .false) THEN DO
749                                         pointsOverallHome = pointsOverallHome + 1
750                                         Set1HomeResult = 1
751                         END
752                         ELSE IF(pointsAwaySet = 2 & Mistake_1_Set = .false) THEN DO
753                                         pointsOverallAway = pointsOverallAway +1
754                                         Set1AwayResult = 1
755                         END
756                 END
757         END
758         setsOverallHome = setsOverallHome + pointsHomeSet
759         setsOverallAway = setsOverallAway + pointsAwaySet
```

Snippet 14: 1st Match Input Control

In the first step, the values entered are checked regarding a non-existing entry. Further, it is checked if the scores of the same set contain the same value, as there is no such result possible in tennis. If correctness of this control is given, then another routine (‘CheckCorrectNumberInput’, line 717) is called and the scores of the first set are passed to this routine. The individual sets of the first match are checked one after another. However, the ‘CheckCorrectNumberInput’ routine checks whether the scores lead to a correct result. The return value of the routine contains not only an interim result, which is needed later in order to get the final result but also a variable named 'Mistake'. This variable is true if the entered score is wrong and false if the entered score is correct. The following snippet shows the routine ‘CheckCorrectNumberInput’.

```
1205 :: routine CheckCorrectNumberInput public
1206 use arg Points
1207 parse Var Points Home Away
1208 pointsHome = 0
1209 pointsAway = 0
1210 mistake = .false
1211 IF(Home = 7 ) THEN DO
1212                 IF(Away < 7) THEN pointsHome = 1
1213                 ELSE mistake = .true
1214         END
1215 ELSE IF(Away = 7 ) THEN DO
1216                 IF(Home < 7) THEN pointsAway = 1
1217                 ELSE mistake = .true
1218         END
1219 ELSE IF (Home = 6) THEN DO
1220                 IF (Away < 5) THEN pointsHome = 1
1221                 ELSE mistake = .true
1222         END
1223 ELSE IF (Away = 6) THEN DO
1224         IF (Home < 5) THEN pointsAway = 1
1225         ELSE mistake = .true
1226 END
1227 ELSE mistake = .true
1228 .environment~gamesOverallHome = .environment~gamesOverallHome + Home
1229 .environment~gamesOverallAway = .environment~gamesOverallAway + Away
1230 return pointsHome pointsAway mistake
```

Snippet 15: Routine that Checks the Correct Input of a Set

This routine tests whether the input of the record is entered in the correct format and notes which player won the set.

Returning to Snippet 14, it can be seen that the same procedure also applies to the second set (line 723). The third set has a small distinction; the routine 'CheckCorrectNumberInputTieBreak' must be invoked, since there is no normal set played but a match-tie-break. The routine looks as follows.

```
1235 :: routine CheckCorrectNumberInputTieBreak public
1236 use arg Points
1237 parse Var Points Home Away
1238 pointsHome = 0
1239 pointsAway = 0
1240 mistake = .false
1241
1242 IF((Home = 10) & (Away < 9)) THEN pointsHome = 1
1243 ELSE IF ((Home > 10) & ((Home - Away) = 2)) THEN pointsHome = 1
1244 ELSE IF((Away = 10) & (Home < 9)) THEN pointsAway = 1
1245 ELSE IF((Away > 10) & ((Away - Home) = 2)) THEN pointsAway = 1
1246 ELSE mistake = .true
1247 .environment~gamesOverallHome = .environment~gamesOverallHome + Home
1248 .environment~gamesOverallAway = .environment~gamesOverallAway + Away
1249 return pointsHome pointsAway mistake
```

Snippet 16: Routine that Checks the Correct Input of a Match-Tie-Break

It is checked whether the result corresponds to the rules of tennis regarding a match-tie-break. (In detail, that means that a match-tie-break is won as soon as a player has scored 10 points and the opponent has 8 points or less than 8 points. If this 2 points difference does not exist, the set is continued until a player has a 2 point lead. For example, the final score is 15:13, the player who has 15 points wins.) What is more, both routines ‘CheckCorrectNumberInput’ and ‘CheckCorrectNumberInputTieBreak’,

make entries in the '.environment' directory because the directory can be accessed over the boundaries of a routine and the saved values are required by another routine. [Flat13, p.107]

Now that the correct input has been checked and an interims result is created, the winner of the game has to be found. The player who wins two sets wins the game. Snippet 17 is used to determine the winner of a match.

```
748                         IF(pointsHomeSet = 2 & Mistake_1_Set = .false) THEN DO
749                                     pointsOverallHome = pointsOverallHome + 1
750                                     Set1HomeResult = 1
751                         END
752                         ELSE IF(pointsAwaySet = 2 & Mistake_1_Set = .false) THEN DO
753                                     pointsOverallAway = pointsOverallAway +1
754                                     Set1AwayResult = 1
755                         END
756                 END
757         END
758         setsOverallHome = setsOverallHome + pointsHomeSet
759         setsOverallAway = setsOverallAway + pointsAwaySet
```

Snippet 17: Determining the Winner of a Match

Thus, the first match is checked for errors, the winner is found and all the information needed are noted which will later be saved in the database. Moreover, these steps must be made for all 4 singles and 2 doubles matches in order to finish an encounter.

The last step is to store the relevant data, such as points, games, sets and further game information. These are needed for the data export in order to reconstruct the encounters and to create the ranking.

```
1080 SetsHome = SetsHome + setsOverallHome
1081 SetsAway = SetsAway + setsOverallAway
1082
1083
1084 IF ((pointsOverallHome = 3) & (pointsOverallAway = 3)) THEN DO
1085  tf_winner~Text = "Unentschieden!"
1086  tf_result~Text = "3  :  3"
1087  PunkteHome = PunkteHome + 1
1088  PunkteAway = PunkteAway + 1
1089  sqlkomm = "UPDATE Team SET Punkte="PunkteHome",Games=".environment~gamesOverallHome",Sets="SetsHome" WHERE idTeam="IDHome""
1090  update = statement~executeUpdate(sqlkomm)
1091  sqlkomm = "UPDATE Team SET Punkte="PunkteAway",Games=".environment~gamesOverallAway",Sets="SetsAway" WHERE idTeam="IDAway""
1092  update = statement~executeUpdate(sqlkomm)
1093 END
1094 ELSE IF (pointsOverallHome > pointsOverallAway) THEN DO
1095  tf_winner~Text = .environment~homeTeam
1096  tf_result~Text = pointsOverallHome"  :  "pointsOverallAway
1097  PunkteHome = PunkteHome + 2
1098  sqlkomm = "UPDATE Team SET Punkte="PunkteHome",Games=".environment~gamesOverallHome",Sets="SetsHome" WHERE idTeam="IDHome""
1099  update = statement~executeUpdate(sqlkomm)
1100  sqlkomm = "UPDATE Team SET Games=".environment~gamesOverallAway",Sets="SetsAway" WHERE idTeam="IDAway""
1101  update = statement~executeUpdate(sqlkomm)
1102 END
1103 ELSE IF (pointsOverallHome < pointsOverallAway) THEN DO
1104  tf_winner~Text = .environment~awayTeam
1105  tf_result~Text = pointsOverallHome"  :  "pointsOverallAway
1106  PunkteAway = PunkteAway + 2
1107  sqlkomm = "UPDATE Team SET Games=".environment~gamesOverallHome",Sets="SetsHome" WHERE idTeam="IDHome""
1108  update = statement~executeUpdate(sqlkomm)
1109  sqlkomm = "UPDATE Team SET Punkte="PunkteAway",Games=".environment~gamesOverallAway",Sets="SetsAway" WHERE idTeam="IDAway""
1110  update = statement~executeUpdate(sqlkomm)
1111 END
```

Snippet 18: Points Division and Storage of Points, Games & Sets

The points are distributed according to the scoring scheme presented in ‘2.1.2 Scoring’. The snippet above shows the SQL query which updates the table ‘Team’ with the new final results.

Next, the following information are stored in the database:

Table 1: Game Information Stored in the Database

Stored Data	**Description**
Player 1	ID of the player
Team 1	ID of the team; Is needed because the player can be borrowed from another team.
Player 2	ID of the player
Team 2	ID of the team; Is needed because the player can be borrowed from another team.
Result Player 1 of the 1st Set	For example: 6
Result Player 2 of the 1st Set	For example: 4
Result Player 1 of the 2nd Set	For example: 7
Result Player 2 of the 2nd Set	For example: 5
(Result Player 1 of the 3rd Set)	In brackets since a 3rd set is not played in every match and depending on that a different SQL query is required.
(Result Player 2 of the 3rd Set)	In brackets since a 3rd set is not played in every match and depending on that a different SQL query is required.

Winter cup Round	
Date	Date of the game day.
Finale Result Player 1	This entry is 1 for winning and 0 for losing.
Finale Result Player 2	This entry is 1 for winning and 0 for losing.

The following excerpt shows the storage of the first single match, whereby also all other matches in singles and double must be stored.

```
1128 IF(V1s3sh = "" | V1s3sa = "") THEN DO
1129
1130  sqlkomm_1Satz = "insert into 4440_3.Singles (`ID-Spieler1`,`ID-Team1`,`ID-Spieler2`,`ID-Team2`,"-
1131 "Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2) Values ("spielerID11","-
1132 ""teamID1","spielerID12","teamID2","V1s1sh","V1s1sa","V1s2sh","V1s2sa","cb_round~Value","-
1133 """dp_date~getValue~toString"'","Set1HomeResult","Set1AwayResult")"
1134
1135 singles = statement_Spiele~executeUpdate(sqlkomm_1Satz)
1136 END
1137 ELSE DO
1138
1139 sqlkomm_1Satz = "insert into 4440_3.Singles (`ID-Spieler1`,`ID-Team1`,`ID-Spieler2`,`ID-Team2`,"-
1140 "Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2) Values"-
1141 " ("spielerID11","teamID1","spielerID12","teamID2","V1s1sh","V1s1sa","V1s2sh","V1s2sa","V1s3sh","-
1142 ""V1s3sa","cb_round~Value",'"dp_date~getValue~toString"',"Set1HomeResult","Set1AwayResult")"
1143
1144 singles = statement_Spiele~executeUpdate(sqlkomm_1Satz)
1145 END
```

Snippet 19: Saving All Match Information

7.6 Data Export

After the tab ‘Spielbericht’ of the main program has been explained, the ‘Data Export’ tab is now explained in more detail. The functions offered in this tab are used to export important data which the user may need during the winter cup tournament. With each export, information is retrieved from the database and edited to create tables with the needed data. The implementation is carried out using the scripting framework of AOO. This chapter explains the implementation of the export functions.

If a routine is invoked via a button (triggered by the user) of the various data export options, one of the following calls occurs. As it can be seen in Snippet 20 the code for the data export is located in different individual programs which are started at this point

by means of a DOS command. At this point, only two of the total five options are shown, since the functioning of the others is the same.

```
37 /************************************************/
38 /*          RESULTAT SPIELER EXPORT             */
39 /************************************************/
40 ::routine klickPlayerExportAction public
41 use arg slotDir
42
43 /*@get("cb_Player_Export")*/
44 /*@get("btn_Player_Export")*/
45 /*@get("lb_Info_Export")*/
46
47 IF(cb_Player_Export~Value = "-DB Verbindung erstellen!-") THEN -
48                         lb_Info_Export~Text = "Bitte zuerst mit der Datenbank verbinden!"
49 ELSE IF (cb_Player_Export~Value = "Bitte Spieler ausw"'E4'x"hlen!") THEN -
50                         lb_Info_Export~Text = "Bitte Spieler ausw"'E4'x"hlen!"
51 ELSE DO
52         lb_Info_Export~Text = ""
53         SpielerID = DBPlayerID(cb_Player_Export~Value)
54         "start /D" directory() "rexxj.cmd Spieler.rex" SpielerID
55         cb_Player_Export~getSelectionModel()~Select("Bitte Spieler ausw"'E4'x"hlen!")
56 END
57
58 /************************************************************/
59 /*              RESULTAT MANNSCHAFT EXPORT                  */
60 /************************************************************/
61 ::routine klickTeamExportAction public
62 use arg slotDir
63
64 /*@get("cb_Team_Export")*/
65 /*@get("btn_Team_Export")*/
66 /*@get("lb_Info_Export")*/
67
68 IF(cb_Team_Export~Value = "-DB Verbindung erstellen!-") THEN -
69                         lb_Info_Export~Text = "Bitte zuerst mit der Datenbank verbinden!"
70 ELSE IF (cb_Team_Export~Value = "Bitte Mannschaft ausw"'E4'x"hlen!") THEN -
71                         lb_Info_Export~Text = "Bitte Mannschaft ausw"'E4'x"hlen!"
72 ELSE DO
73         lb_Info_Export~Text = ""
74         TeamID = DBgetTeamID(cb_Team_Export~Value)
75         "start /D" directory() "rexxj.cmd Mannschaft.rex" TeamID
76         cb_Team_Export~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
77 END
```

Snippet 20: Start of the various Export Programs

First it is checked whether a database connection has been established and further a player or a team has been selected, otherwise the user is prompted by means of the error label to do so. If a database connection exists and a player or team is selected, the routine ‘DBPlayerID’ or ‘DBTeamID’ is called which returns the ID of the player or team. Moreover, the name of the team or the player must be handed over to the routine. Now the individual programs can be started via a DOS command. Important is that the ID is passed to the individual programs. The last section of this routine brings the Choice Box to the starting position in which the text 'Bitte Mannschaft auswählen!' or 'Bitte Spieler auswählen!' is selected.

The following data export options are available to the user:

Table 2: Export Options

Export Options	**Description**
Export all teams	Displays all teams, their team leaders and the telephone number of the team leaders.
Export all players	Displays all players with their telephone numbers and the team they belong to.
Results of a team	Displays all the results of a certain team.
Results of player	Displays all the results of a certain player.
Current Ranking	Displays the current ranking.

The following snippet explains the code that creates and exports the current ranking. All other export options are structured similarly and are therefore not discussed in detail.

```
oDesktop  = UNO.createDesktop()
xComponentLoader = oDesktop~XDesktop~XComponentLoader

url = "private:factory/scalc"
xCalcComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/*****************************************************************************/
/* Spalten Namen */
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"
CALL UNO.setCell xSheet, 1, 2, "Mannschaften"
CALL UNO.setCell xSheet, 2, 2, "Mannschaftf"'FC'x"hrer"
CALL UNO.setCell xSheet, 3, 2, "Telefonnummer"
/*****************************************************************************/
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=***")
statement = conn~createStatement
sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
team = statement~executeQuery(sqlkomm)

index = 3
check = team~First
DO WHILE check = .true
        IF (team~getString("TeamName") <> "NO TEAM") THEN DO
                CALL UNO.setCell xSheet, 1 , index, team~getString("TeamName")

                statement1 = conn~createStatement
                sqlkomm = "SELECT * FROM Players  WHERE (idSpieler = "team~getString("TeamCaptain")")"
                player = statement1~executeQuery(sqlkomm)
                check_player = player~First
                IF check_player = .true THEN DO
                 Vorname = player~getString("Vorname")
                 Nachname = player~getString("Nachname")
                 Leader = Vorname Nachname
                 /*************************************************************/
                 CALL UNO.setCell xSheet, 2 , index, Leader
                 CALL UNO.setCell xSheet, 3 , index, "'"player~getString("Telefonnummer")
                 /*************************************************************/
                END
                 index = index + 1
        END
        check = team~Next
END
player~Close()
statement1~Close()
team~Close()
statement~Close()
conn~Close()
```

Snippet 21: Export Current Ranking 1

As a reminder the scripting framework of AOO, can be enabled due to BSF4ooRexx and the contained UNO.CLS package. The first step is to load the object ‘uno.createDesktop()’ which is requested to get the ‘XComponentLoader’. With the ‘XComponentLoader’ a new Calc-file can be created. The first spreadsheet of the document is now selected using the method ‘getByIndex (0)’.

Now new entries can be inserted into the spreadsheet. The method ‘setCell’ of the ‘uno’ object is invoked, and the information of the spreadsheet (xSheet), the position where the entries have to be written (0,0) and the text are handed over.

```
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"
```

Snippet 22: Writing a Text into the Spreadsheet

Snippet 22 shows how the text "WinterCup im Haas-Tenniscenter" is written in the first cell (position 0,0) of the spreadsheet. It is important to know that the first 0 specifies

the column and the second 0 specifies the row of the spreadsheet. Before the ranking is created, a few general information will be added to the spreadsheet.

Now the ranking is created. The following snippet shows the progress of the program.

```
/*****************************************************************************/
indexJ = index + 3
/*****************************************************************************/
CALL UNO.setCell xSheet, 0, indexJ, "Rang"
CALL UNO.setCell xSheet, 1, indexJ, "Mannschaften"
CALL UNO.setCell xSheet, 2, indexJ, "Punkte"
CALL UNO.setCell xSheet, 3, indexJ, "Games"
CALL UNO.setCell xSheet, 4, indexJ, "Sets"
/*****************************************************************************/
indexJ = indexJ + 1

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user="-
"4440_3&password=Haas-Tenniscenter")
statement = conn~createStatement
sqlkomm = "SELECT * FROM Team ORDER BY Punkte DESC, Games DESC, Sets DESC"
team = statement~executeQuery(sqlkomm)

Rang = 1
check = team~First
DO WHILE check = .true
 IF (team~getString("TeamName") <> "NO TEAM") THEN DO
   /*****************************************************************************/
   CALL UNO.setCell xSheet, 0, indexJ, Rang
   CALL UNO.setCell xSheet, 1, indexJ, team~getString("TeamName")
   CALL UNO.setCell xSheet, 2, indexJ, team~getString("Punkte")
   CALL UNO.setCell xSheet, 3, indexJ, team~getString("Games")
   CALL UNO.setCell xSheet, 4, indexJ, team~getString("Sets")
   /*****************************************************************************/
   indexJ = indexJ + 1
   Rang = Rang + 1
 END
check = team~Next
END

team~Close()
statement~Close()
conn~Close()
```

Snippet 23: Export Current Ranking 2

First of all, the table head of the ranking is created. ‘indexJ’ is an auxiliary variable created by means of the ‘index’. What is more, the ‘index’ variable indicates the exact number of registered teams. Nevertheless, ‘indexJ’ is used to specify the row in which the table head is created. As the number of teams can vary, therefore the position of the ranking must be changed dynamically so that overlaps are avoided. Once the table head is created, a database connection is established in the next step to retrieve the required data.

```
sqlkomm = "SELECT * FROM Team ORDER BY Punkte DESC, Games DESC, Sets DESC"
```

Snippet 24: SQL Query Current Ranking

Important is the SELECT query as the returned result set is already sorted according to the requirements of the game rules. Now the data has only to be read out step by step and added to the Calc-file.

The last step is the formatting of the created file in order to present the data more structured and more clearly.

```
/* Format Title & Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                          bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Telefonnummer: Right*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                          bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index + 3,4,index + 3)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                          bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
/* Format Table: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index + 3,0,indexJ)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify", -
                        bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
/* Format Table: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(2,index + 3,4,indexJ)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify", -
                        bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
/* Changing Column Width */
xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns

/*Column 2 & 4*/
DO i = 1 to 4
 xCol = xCols~getByIndex(i) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth + 1000
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
END

/*Column 1*/
 xCol = xCols~getByIndex(1) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth - 200
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width

::requires UNO.CLS
```

Snippet 25: Export Current Ranking 3

In order to display a text centered or bold, the properties of the required cells are selected in the first step. Now the use of the setter method 'setPropertyValue' is needed to change the desired properties.

Finally, the widths of the cells are changed since the inserted values are longer than the cells. For this purpose, getting the columns of the spreadsheet in the first step is required. With the 'getByIndex ()' method, the desired column can be accessed. Afterwards, the properties are called to get the current width of the column using the getter method ‘getPropertyValue(“Width”)’. The old width is added by 1000 and with

the setter method ‘setPropertyValue’ the property of the column is changed to the new width. This change is made for columns 2 to 4. Finally, column 1 is scaled down because it was too large to represent the information properly.

The spreadsheet looks then as follows:

	A	B	C	D	E
1	**WinterCup im Haas-Tenniscenter**				
2					
3		**Mannschaften**	**Mannschaftführer**	**Telefonnummer**	
4		Break 1	Flo Duck	0676192300123	
5		Kolibris	Dieter Wimmer	0676213123	
6		SSC Marek	Andrej Sokol	06764480372	
7		SundayTeam	Stefan Sulic	06801234567	
8		TC Arsenal	Michael Deimel	0654324567	
9		TC Haas	Ben Vogla	06767155171	
10		TC Hennersdorf	Martin Purtscher	06006969696	
11		Team 5	Mamica Vogla	06766301677	
12					
13					
14					
15	**Rang**	**Mannschaften**	**Punkte**	**Games**	**Sets**
16	1	Kolibris	113	69	195
17	2	Break 1	85	85	171
18	3	TC Arsenal	22	57	41
19	4	SSC Marek	19	45	34
20	5	TC Hennersdorf	15	56	54
21	6	SundayTeam	13	52	66
22	7	Team 5	3	51	29
23	8	TC Haas	1	51	6

Figure 20: Export Current Ranking

One thing has yet to be added. The ID of the exporting team or player is handed over to some export programs when started through the DOS command. These can be accessed as follows.

```
parse arg TeamID
```

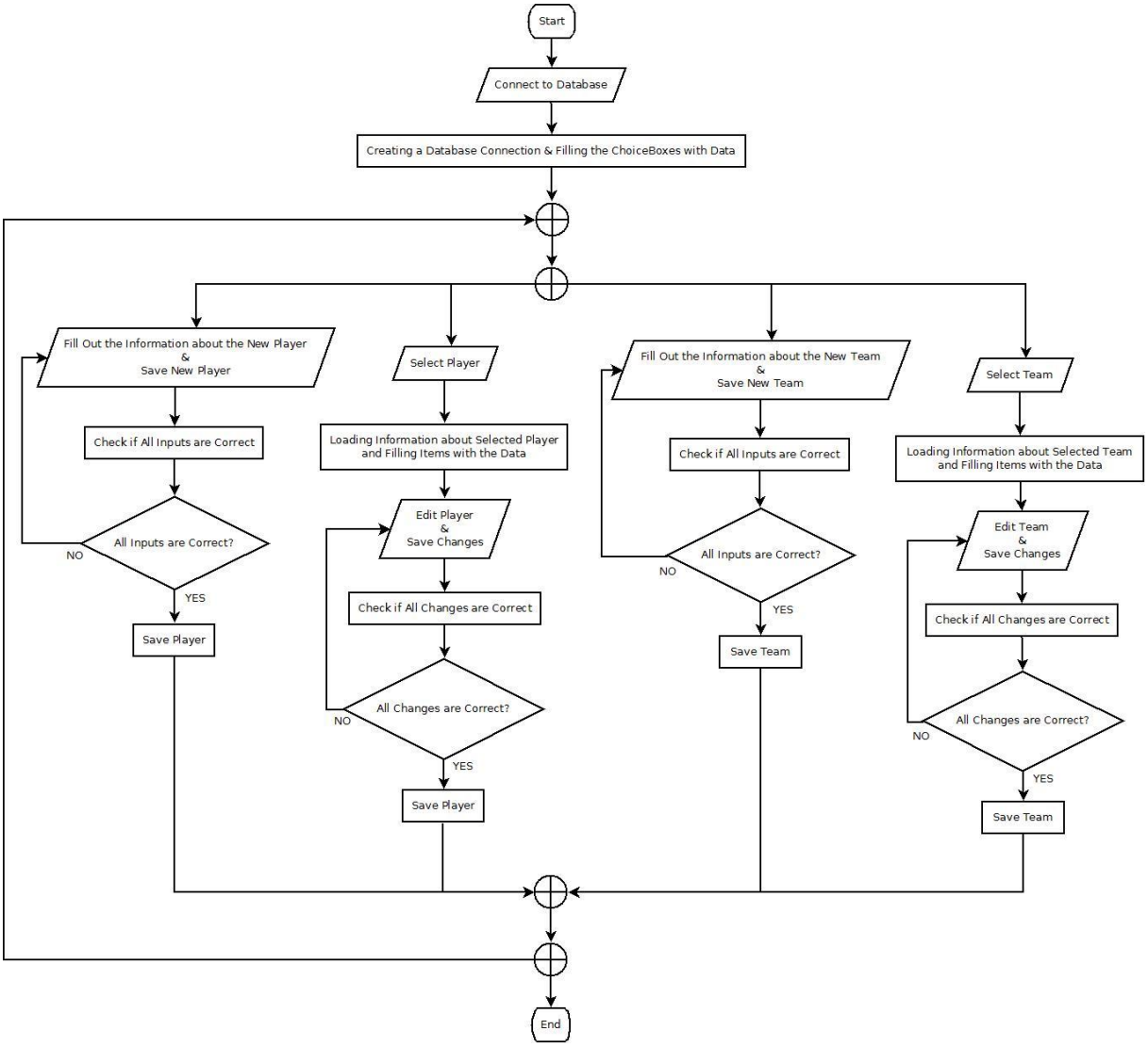
Snippet 26: Access to the Handover Parameter

8 Development of the Second Program

This chapter explains the second program which is started from the main program via the menu ‘Mannschaften / Spieler’ (English: ‘Teams / Players’).

8.1 Overview

The second program has the task to create, edit or delete players or teams. The following flow chart is intended to visualize the task sequences of the program.



Flow Chart 2: Second Program

After the program is started, the user must establish a database connection. Now the user can freely use all functions of the program. The program has four tabs which categorizes the various functions to ensure easier and faster use.

Table 3: Second Program Tabs

Tab	Description
Spieler Hinzufügen	Adding a new player.
Spieler Bearbeiten	Editing or deleting an existing player.
Mannschaft Hinzufügen	Adding a new team.
Mannschaft Bearbeiten	Editing or deleting an existing team.

Before new entries are created or old entries are changed in the database, the inputs are validated. This means that new entries and changed entries are checked in order that duplicate entries do not occur. In the narrow sense that means, if a player or team already exists the user should not be able to create the same person or team again. Further, persons cannot be deleted as long as they are team leaders. A team always needs a team leader. However, a team leader can be changed in the 'Mannschaften Bearbeiten' tab and then respective person can be deleted. Moreover, if a team is deleted all players belonging to this team are automatically added to the team called 'No Team'. (Players always have to belong to a team)

8.2 New Player

If the button 'Speichern', in the tab 'Spieler Hinzufügen' is triggered by the user, the routine 'klickNewPlayerSaveAction' is called. This routine executes the code for storing a new player in the database. As already mentioned before, all control elements of the GUI are made accessible using the 'slotDir' argument at the beginning of a routine. In the next step the input is initialized and separated by the 'parse var' instruction(see snippet below) from unneeded blank spaces after the input text. Afterwards, it is tested whether all text fields have been filled out by the user and otherwise an error message is prompted

```
499 Vorname = tf_newPlayer_Surname~Text
500 parse var Vorname Vorname .
501 Nachname = tf_newPlayer_Lastname~Text
502 parse var Nachname Nachname .
503 Tel = tf_newPlayer_Telnr~Text
504 parse var Tel Tel .
505 Mannschaft = cb_newPlayer_Team~Value
506 IF .environment~DBConnection = .true THEN DO
507 SELECT
508 WHEN (Vorname = " " & Nachname = " " & Tel = " " & (Mannschaft = "-Bitte"-
509                         " Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte Daten Laden!-")) THEN DO
510         tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
511         tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
512         tf_newPlayer_Telnr~Text = "Bitte TELEFONNR. eingeben!"
513         lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
514 END
515 WHEN (Vorname = " " & Nachname = " " & Tel = " ") THEN DO
516         tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
517         tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
518         tf_newPlayer_Telnr~Text = "Bitte TELEFONNR. eingeben!"
519 END
520 WHEN (Vorname = " " & Nachname = " ") THEN DO
521         tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
522         tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
523 END
524 WHEN (Vorname = " " & Tel = " ") THEN DO
525         tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
526         tf_newPlayer_Telnr~Text = "Bitte NACHNAME eingeben!"
527 END
528 WHEN (Vorname = " " & (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte"-
529                                                                 " Daten Laden!-")) THEN DO
530         tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
531         lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
532 END
533 WHEN (Tel = " " & Nachname = " ") THEN DO
534         tf_newPlayer_Telnr~Text = "Bitte VORNAME eingeben!"
535         tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
536 END
537 WHEN (Tel = " " & (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte"-
538                                                             " Daten Laden!-")) THEN DO
539         tf_newPlayer_Telnr~Text = "Bitte VORNAME eingeben!"
540         lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
541 END
542 WHEN Vorname = " " THEN tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
543 WHEN Nachname = " " THEN tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
544 WHEN Tel = " " THEN tf_newPlayer_Telnr~Text = "Bitte TELEFONNR. eingeben!"
545 WHEN (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte Daten Laden!-") -
546                         THEN lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
```

Snippet 27: New Player (Second Program)

If the user has filled out all text fields and also has selected a team where the player belongs to, the next step is to check if the new player does not already exist before a new entry is made in the database.

```
547 OTHERWISE
548  CheckPlayerName = .false
549  CheckPlayerName = DBPlayerCheckName(Vorname Nachname)
550
551  IF CheckPlayerName = .true THEN DO
552
553  /* MySQL Verbindung erstellen  */
554  driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
555  bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
556  conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user"-
557                                                  "=4440_3&password=Haas-Tenniscenter")
558  statement = conn~createStatement
559  sqlkomm = "Select * FROM Team WHERE (TeamName = '"Mannschaft"')"
560  team = statement~executeQuery(sqlkomm)
561
562  check = team~First
563  IF check = .true THEN teamID = team~getString("idTeam")
564   statement = conn~createStatement
565   sqlkomm = "INSERT INTO Players(Vorname,Nachname,Telefonnummer,Team) VALUES"-
566                                        " ('"Vorname"','"Nachname"','"Tel"',"teamID")"
567   spieler = statement~executeUpdate(sqlkomm)
568
569   team~Close()
570   statement~Close()
571   conn~Close()
572
573   lb_newPlayer_Info~Text = "Spieler '"Vorname Nachname"' wurde erfolgreich hinzugef"'FC'x"gt!"
574   CALL DBConn(slotDir)
575  END
576  ELSE lb_newPlayer_Info~Text = "Spieler existiert bereits!"
577  END
578 END
579 ELSE lb_newPlayer_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"
```

Snippet 28: New Player 2 (Second Program)

The control of a duplicate entry in the database is done in an extra routine named ‘CheckPlayerName’. This routine is handed over the first and last name of the new player. Moreover, the return value of the routine is true if there is no other player existing under this name and a new entry can be made in the database. If this player already exists, the user is made aware and no new entry is created in the database.

The following snippet shows how to check whether a player already exists or not.

```
1031 ::routine DBPlayerCheckName public
1032 use arg Info
1033 parse var Info Vorname Nachname
1034 checkName = .false
1035
1036 /* MySQL Verbindung erstellen  */
1037 driverMgr = bsf.loadClass("java.sql.DriverManager")
1038 bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
1039 conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver"-
1040         ".at:3306/4440_3?user=4440_3&password=Haas-Tenniscenter")
1041
1042 statement = conn~createStatement
1043 sqlkomm = "SELECT * FROM Players WHERE"-
1044         " (Vorname = '"Vorname"' AND Nachname = '"Nachname"')"
1045 spieler = statement~executeQuery(sqlkomm)
1046
1047 check = spieler~First
1048 IF check <> .true THEN checkName = .true
1049
1050 spieler~Close()
1051 statement~Close()
1052 conn~Close()
1053
1054 return checkName
```

Snippet 29: Check if Player Exists

The most important part of this snippet is the SQL query (line 1043) as the SELECT command searches in the table ‘Players’ for entries which satisfy the condition given in the SELECT command. The condition is the name of the new player. If the result set contains an entry, a player already exists under this name and the return value is false. Otherwise, the return value is true and a new player is created.

8.3 New Team

Adding a new team is very similar to adding a new player. Therefore, only the small differences are explained in this section.

```
175 lb_newTeam_Info~Text = ""
176 TeamName = tf_newTeam_Name~Text
177 parse var TeamName TN1 TN2 TN3 .
178 Leader = cb_newTeam_Leader~Value
179 SAY Leader
180 CheckName = .false
181
182 IF .environment~DBConnection = .true THEN DO
183
184  IF (TN1 = "") THEN lb_newTeam_Info~Text = "Ung"'FC'x"1tiger Mannschaftsname!"
185  ELSE IF (TN1 <> "") THEN DO
186         TeamName = TN1
187         IF (TN2 <> "" | TN2 <> " ") THEN DO
188                 TeamName = TN1 TN2
189                 IF (TN3 <> "" | TN3 <> " ") THEN DO
190                         TeamName = TN1 TN2 TN3
191                 END
192         END
193         ELSE IF (TN3 <> "" | TN3 <> " ") THEN DO
194                 TeamName = TN1 TN3
195         END
```

Snippet 30: New Team (Second Program)

A team name can contain of up to three words. Snippet 30 illustrates the section where the input of the user is checked for blank spaces and if the input contains more than three words. The overfilled words are simply cut off by using the 'parse var' statement which also divides the input into three separate variables. The point, at the end of the statement, indicates that everything after the third word is no longer needed and cut off.

The next step checks whether an entry was made at all and how many words (one, two or three) it contains. Afterwards the team name is handed over to the routine ‘DBTeamCheckName’ in order to check if this team already exists or not.

The rest of the process can be found in the previous chapter.

8.4 Edit Team

The ‘Mannschaft Bearbeiten’ tab has two important functions. On the one hand the editing of an already existing team and on the other the deletion of an existing team.

8.4.1 Editing

If the database connection has been established in the first step, the Choice Boxes of the ‘Mannschaft Bearbeiten' tab are filled with the corresponding data. Now the user can select a team to be edited. After the user has made a selection, some control elements become visible. A look back to chapter ‘6.3 GUI of the Second Program’ gives an idea of how this looks like.

To make an item visible or invisible, the method 'setVisible ()' is used. The parameters, ‘.true’ must be passed for visible and ‘.false’ for invisible.

```
cb_editTeam_Leader~setVisible(.true)
```

Now the user is provided with all other functions and the editing of a team can begin. If the user has made the desired changes, the save button has to be triggered.

Giving the next snippet a look, it can been seen that again the input of the user is checked in order to avoid a duplicate entries. The instructions before the illustrated snippet are once again the initialization of all necessary elements of the GUI.

```
309 lb_editTeam_Info~Text = ""
310 TeamName = tf_editTeam_Name~Text
311 parse var TeamName TN1 TN2 TN3 .
312 TeamLeader = cb_editTeam_Leader~Value
313 .environment~SelectedTeam = cb_editTeam_Select~Value
314 CheckName = .false
315
316 IF (TN1 = "") THEN lb_editTeam_Info~Text = "Ung"'FC'x"ltiger Mannschaftsname!"
317 ELSE IF (TN1 <> "") THEN DO
318  TeamName = TN1
319  IF (TN2 <> "" | TN2 <> " ") THEN DO
320         TeamName = TN1 TN2
321         IF (TN3 <> "" | TN3 <> " ") THEN DO
322                TeamName = TN1 TN2 TN3
323         END
324 END
325 ELSE IF (TN3 <> "" | TN3 <> " ") THEN DO
326         TeamName = TN1 TN3
327 END
328         CheckName = DBTeamCheckName(TeamName)
```

Snippet 31: Edit Team

If the team name is not used by another team, the program can be continued. The team name is made accessible over the boundaries of the routine by using the .environment directory. The next step is to invoke the routine ‘DBSaveEditTeam’ and to hand over the team leaders name which also could be changed by the user. This routine also needs the team name which is why the team name has to be accessible in the global scope.

```
329 CheckName = DBTeamCheckName(TeamName)
330
331 IF (CheckName = .false & .environment~SelectedTeam != TeamName) -
332          THEN lb_editTeam_Info~Text = "Dieser Mannschaftsname existiert bereits!"
333  ELSE DO
334   mistake = .false
335   .environment~TName = TeamName
336   mistake =  DBSaveEditTeam(TeamLeader)
337
338   IF (mistake = .true) THEN lb_editTeam_Info~Text = "Spieler ist"-
339         " Mannschaftsf"'FC'x"hrer!"' 0DC4'x"nderungen konnten nicht vorgenommen werden!"
340   ELSE lb_editTeam_Info~Text = "Mannschaft '"TeamName"' erfolgreich bearbeitet!"
341
342    /*****************************************/
343    cb_editTeam_Leader~setVisible(.false)
344
345    btn_editTeam_Save~setVisible(.false)
346    btn_editTeam_Delete~setVisible(.true)
347    btn_editTeam~setVisible(.true)
348    btn_editTeam_Cancel~setVisible(.false)
349
350    lb_editTeam_Name~setVisible(.false)
351    lb_editTeam_Leader~setVisible(.false)
352
353    tf_editTeam_Name~setVisible(.false)
354    /*****************************************/
355    CALL DBConn(slotDir)
356   END
357 END
```

Snippet 32: Edit Team 2

One task of the ‘DBSaveEditTeam’ (line 336) routine is to check if the chosen team leader is not already a team leader of another team. The other task is to save the changes made by the user, if the chosen team leader is allowed to be a team leader. Otherwise the changes will be denied and the return value of the routine is true. Further an error message is displayed to the user. Nevertheless, if all changes are correct the last step is the reset of the form by switching all previously made visible elements, invisible again. Moreover, the routine 'DBConn(slotDir)' (line 355) is called. This routine has the same task as the database connection which is triggered by the user at the program start up. This step is needed in order to update all Choice Boxes as new changes have been made. The 'DBConn(slotDir)' routine is executed as soon as something has been successfully changed at the database.

8.4.2 Deleting

In the following, the procedure of deleting is briefly described. A common technique in nowadays user interfaces is to let the user confirm deletion operations twice to avoid unintentional operations. This technique has also be applied in this application.

```
387 IF .environment~delete = .true THEN DO
388  lb_editTeam_Info~Text = "Moechten Sie '"delTeam"' wirklich LOESCHEN?"
389  btn_editTeam_Delete_Cancel~setVisible(.true)
390  btn_editTeam~setVisible(.false)
391  .environment~delete = .false
392 END
```

Snippet 33: Delete Team

If the user triggers the delete button for the first time, the code in the snippet above is executed and the user is asked whether the deletion of the team should actually be made. As one can see the variable '.environment~delete' which is initialized with the value true, is changed to false in this section. If the user triggers the button a second time, this sequence is not executed anymore but instead the program section which actually performs the delete operation on the database.is executed. It is important to note that players who belonged to the deleted team are after that assigned to the team called 'NO TEAM'. Snippet 34 illustrates the second part of the delete routine.

```
393 ELSE IF .environment~delete = .false THEN DO
394   /* MySQL Verbindung erstellen */
395  driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
396  bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
397  conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/"-
398 "4440_3?user=4440_3&password=Haas-Tenniscenter")
399
400  statement = conn~createStatement
401  sqlkomm = "SELECT * FROM Team WHERE (TeamName='"delTeam"')"
402  team = statement~executeQuery(sqlkomm)
403
404  check = team~First
405  IF check = .true THEN teamID = team~getString("idTeam")
406    sqlkomm = "DELETE FROM Team WHERE (idTeam ="teamID")"
407    del = statement~executeUpdate(sqlkomm)
408
409    sqlkomm = "UPDATE Players SET Team="1" WHERE Team="teamID""
410    update = statement~executeUpdate(sqlkomm)
411
412    team~Close()
413    statement~Close()
414    conn~Close()
415
416    .environment~delete = .true
417    btn_editTeam_Delete_Cancel~setVisible(.false)
418    btn_editTeam~setVisible(.true)
419    lb_editTeam_Info~Text = "'"delTeam"' wurde erfolgreich gel"'F6'x"scht!";
420    CALL DBConn(slotDir)
421   END
422  END
423 END
```

Snippet 34: Delete Team 2

8.5 Edit Player

Since editing a player and a team is rather similar, this section will only point out the major differences.

8.5.1 Editing

Again, it is first checked whether the changed name of the player is not identical to an already existing player to avoid duplicated database records. The next step is to check if the user has changed the team to which the player belongs.

```
712 IF (.environment~Team = Team_Selected) THEN DO
713  sqlkomm = "UPDATE Players SET Vorname='"tf_editPlayer_Surname~Text"',"-
714                  " Nachname= '"tf_editPlayer_Lastname~Text"',Telefonnummer="-
715                         " '"tf_editPlayer_Telnr~Text"', Team="teamID" WHERE idSpieler="spielerID""
716
717  update = statement~executeUpdate(sqlkomm)
718  lb_editPlayer_Info~Text = "Spieler '"tf_editPlayer_Surname~Text"-
719                  " tf_editPlayer_Lastname~Text"' erfolgreich bearbeitet!"
720 END
721 ELSE DO
722  sqlkomm = "SELECT * FROM Team WHERE (TeamCaptain="spielerID")"
723  checkLeader = statement~executeQuery(sqlkomm)
724  check = checkLeader~First
725  IF check = .true THEN lb_editPlayer_Info~Text = "Spieler ist"-
726                  " Mannschaftsf"'FC'x"hrer!"'0DC4'x"nderungen konnten nicht vorgenommen werden!"
727  ELSE DO
728   sqlkomm = "UPDATE Players SET Vorname='"tf_editPlayer_Surname~Text"',"-
729                  " Nachname= '"tf_editPlayer_Lastname~Text"',Telefonnummer="-
730                         " '"tf_editPlayer_Telnr~Text"', Team="teamID" WHERE idSpieler="spielerID""
731
732   update = statement~executeUpdate(sqlkomm)
733   lb_editPlayer_Info~Text = "Spieler '"tf_editPlayer_Surname~Text"-
734                               " tf_editPlayer_Lastname~Text"' wurde erfolgreich bearbeitet!"
735  END
736 checkLeader~Close()
737 END
```

Snippet 35: Edit Player

The snippet above shows how to process the saving of the modified entries in the database. If the team to which the player belongs is not changed, it can be continued immediately with the updating of the database entry. If not, it must be checked whether the player is a team leader of another team, since otherwise a team would exist without a team leaders. In detail, this looks as follows:

```
722  sqlkomm = "SELECT * FROM Team WHERE (TeamCaptain="spielerID")"
723  checkLeader = statement~executeQuery(sqlkomm)
724  check = checkLeader~First
725  IF check = .true THEN lb_editPlayer_Info~Text = "Spieler ist"-
726                  " Mannschaftsf"'FC'x"hrer!"'0DC4'x"nderungen konnten nicht vorgenommen werden!"
```

Snippet 36: Check if a Player is a Team Leader

The ‘SELECT’ command searches, in the ‘Team’-table, for entries which have respective player as team leader. If the query leads to a result the changes made by the user are denied. (Important: For the just mentioned ‘SELECT’ query the ID of the

player is required. This ID is extracted from the Database one sequence before. It is important that when the ID is extracted, not the changed name, but the original (old) name is used, otherwise no ID is found, since the changes have not yet been made in the Database.)

On the other hand, if the player is not a team leader of a team all user-requested changes can be made and the database entry can be updated.

8.5.2 Deleting

Deleting a player is no longer explained in detail. As with deleting a team, the user has to trigger the delete button two times before a player is definitely deleted. Furthermore, it is also necessary to ensure that no team leader is deleted.

9 Conclusion

The intention behind programming WinterCup was to combine my Bachelor thesis with a practical application which can be used even after the work is done. Since my family owns a small indoor tennis center –named Haas-Tenniscenter - creating an application for this family-company seemed self-evident to me. Therefore the idea of WinterCup, a digital solution for managing and evaluating the winter cup tournament was born. As member of this company the needs on functions and design were obvious to me from the beginning. However, they needed to be accepted by all parties. The program will be used in early October with the start of the season.

It is important to note that the knowledge required for this work has been acquired in one semester, thus refute the general belief that learning how to program something takes years. Furthermore, as WinterCup is written in ooRexx it has the advantage that it runs on all common operating systems and is not limited to one system.

On the one hand the program is intended to support the company, on the other hand the thesis is intended to help the reader solving similar problems. The seminar works and Bachelor theses of colleagues were a great help while programming WinterCup. Furthermore, the lecture Business Programming 1 & 2 were necessary in order to learn the basic knowledge in ooRexx, BSF4ooRexx, JavaFX and Apache OpenOffice scripting and thus being able to clarify ambiguities.

References

[FlatFx] Flatscher, Rony G.: JavaFx for ooRexx, Creating Powerful Portable GUI's. http://wi.wu.ac.at:8002/rgf/wu/lehre/autojava/material/foils/AutoJava-BSF4ooRexx-07-JavaFx.pdf, , Retrieved on 2017-06-22

[FlatRx] Flatscher, Rony G.: An Introduction to Procedural and Object-oriented Programming (ooRexx) 1. http://wi.wu.ac.at:8002/rgf/wu/lehre/autowin/material/foils/ooRexx%5f1%2epdf, , Retrieved on 2017-07-15

[Flat13] Flatscher, Rony G.: Introduction to Rexx and ooRexx, From Rexx to Open Object Rexx (ooRexx). 1st Edition. Facultas Verlags- und Buchhandels AG., Vienna 2013.

[Geis14] Geisler, Frank: Datenbanken, Grundlagen und Design. 5th Edition. mitp eine Marke der Verlagsgruppe Hüthig Jehle Rehm GmbH, Heidlberg, München, Landsberg, Hamburg 2014

[MyDrMa] MySQL: 6.1 Connecting to MySQL Using the JDBC DriverManager Interface. https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-connect-drivermanager.html, ,2017-07-01

[MyStOb] 6.2 Using JDBC Statement Objects to Execute SQL. https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-statements.html, ,2017-07-01

[OrFX13] Oracle: What is JavaFX?. http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm, 2013, Retrieved on 2017-06-22

[OrJaDo] Oracle Java Documentation: PATH and CLASSPATH. https://docs.oracle.com/javase/tutorial/essential/environment/paths.html, , Retrieved on 2017-07-15

[WiAO17] Wikipedia: Apache OpenOffice. https://en.wikipedia.org/w/index.php?title=Apache_OpenOffice&direction=prev&oldid=786480594, 2017-06-14, Retrieved on 2017-06-17

[WiDM17] Wikipedia: Database model. https://en.wikipedia.org/w/index.php?title=Database_model&oldid=784818337, 2017-07-10, Retrieved on 2017-06-16

[WiFK17] Wikipedia: Foreign key. https://en.wikipedia.org/wiki/Foreign_key, 2017-07-18, Retrieved on 2017-07-19

[WiJaFx] Wikipedia: JavaFx. https://en.wikipedia.org/w/index.php?title=JavaFX&oldid=784171989, 2017-06-06, Retrieved on 2017-06-22

[WiMuDI] Wikipedia: Multiple document interface. https://en.wikipedia.org/wiki/Multiple_document_interface, 2016-08-19, Retrieved on 2017-06-23

[WiRDde] Wikipedia: Relationale Datenbank. https://de.wikipedia.org/w/index.php?title=Relationale_Datenbank&oldid=164958215, 2017-04-27, Retrieved on 2017-06-16

[WiRMen] Wikipedia: Relational model. https://en.wikipedia.org/w/index.php?title=Relational_model&oldid=779316364, 2017-05-08, Retrieved on 2017-06-16

[WiTe17] Wikipedia: Tennis. https://en.wikipedia.org/w/index.php?title=Tennis&direction=prev&oldid=786849235, 2017-05-26, Retrieved on 2017-07-13.

[W3scAn] w3schools: SQL AND, OR and NOT Operators. https://www.w3schools.com/sql/sql_and_or.asp, , Retrieved on 2017-06-19

[W3scDe] w3schools: SQL DELETE Statement. https://www.w3schools.com/sql/sql_delete.asp, , Retrieved on 2017-06-19

[W3scIn] w3schools: SQL INSERT INTO Statement. https://www.w3schools.com/sql/sql_insert.asp, , Retrieved on 2017-06-19

[W3scOr] w3schools: SQL ORDER BY Keyword. https://www.w3schools.com/sql/sql_orderby.asp, , Retrieved on 2017-06-19

[W3scSe] w3schools: SQL SELECT Statement. https://www.w3schools.com/sql/sql_select.asp, , Retrieved on 2017-06-19

[W3scSq] w3schools: SQL Tutorial. https://www.w3schools.com/sql/default.asp, , Retrieved on 2017-06-19

[W3scUp] w3schools: SQL UPDATE Statement. https://www.w3schools.com/sql/sql_update.asp, , Retrieved on 2017-06-19

[W3scWe] w3schools: SQL WHERE Clause. https://www.w3schools.com/sql/sql_where.asp, , Retrieved on 2017-06-19

Appendix

Main_team_player.rex

```
SAY "PROGRAM START!"

rexxHandler=.rxDocHandler~new

rxApp=BsfCreateRexxProxy(rexxHandler,,"javafx.application.Application")

rxApp~launch(rxApp~getClass, .nil)

::requires "BSF.CLS"

::class RxDocHandler

::method start
  use arg stage

  stage~setTitle("WinterCup im Haas-Tenniscenter")

  rootDocUrl=.bsf~new("java.net.URL", "file:Team_Player_GUI.fxml")

  root=bsf.loadClass("javafx.fxml.FXMLLoader")~load( rootDocUrl  )
  scene=.bsf~new("javafx.scene.Scene", root)
  stage~setScene(scene)

  stage~show
```

Controller_team_player.rex

```
.environment~delete = .true
.environment~deletePlayer = .true
.environment~DBConnection = .false

::requires BSF.CLS


/**************************************************************************************************
*************/
 /*                                CLICK BUTTON DATABASE CONNECTION                                 */
/**************************************************************************************************
*************/
::routine klickConnectDBAction public
use arg slotDir

/*@get("lb_editTeam_Info")*/
/*@get("lb_newTeam_Info")*/
/*@get("lb_editPlayer_Info")*/
/*@get("lb_newPlayer_Info")*/

lb_editTeam_Info~Text = ""
lb_newTeam_Info~Text = ""
lb_editPlayer_Info~Text = ""
lb_newPlayer_Info~Text = ""

CALL DBConn(slotDir)
.environment~DBConnection = .true


/**************************************************************************************************
*************/
 /*                                      DATABASE CONNECTION                                       */
/**************************************************************************************************
*************/
:: routine  DBConn public
use arg slotDir

/*@get("btn_connectDB")*/

  /********************/
 /* Clear Everything */
/********************/
/*****************************************************************************/
/*@get("cb_editTeam_Select")*/
/*@get("cb_editTeam_Leader")*/

/*@get("btn_editTeam_Save")*/
/*@get("btn_editTeam_Delete")*/
/*@get("btn_editTeam")*/
/*@get("btn_editTeam_Cancel")*/

/*@get("lb_editTeam_Select")*/
/*@get("lb_editTeam_Name")*/
/*@get("lb_editTeam_Leader")*/
/*@get("lb_editTeam_Info")*/

/*@get("tf_editTeam_Name")*/

cb_editTeam_Select~getItems~Clear()
cb_editTeam_Leader~getItems~Clear()

--lb_editTeam_Info~Text = ""

tf_editTeam_Name~Text = ""
/*****************************************************************************/
/*@get("cb_newTeam_Leader")*/

/*@get("lb_newTeam_Info")*/

/*@get("tf_newTeam_Name")*/
```

```
cb_newTeam_Leader~getItems~Clear()

--lb_newTeam_Info~Text = ""

tf_newTeam_Name~Text = ""
/*****************************************************************************/
/*@get("cb_newPlayer_Team")*/

/*@get("lb_newPlayer_Info")*/

/*@get("tf_newPlayer_Surname")*/
/*@get("tf_newPlayer_Lastname")*/
/*@get("tf_newPlayer_Telnr")*/

cb_newPlayer_Team~getItems~Clear()

--lb_newPlayer_Info~Text = ""

tf_newPlayer_Surname~Text = ""
tf_newPlayer_Lastname~Text = ""
tf_newPlayer_Telnr~Text = ""
/*****************************************************************************/
/*@get("cb_editPlayer_Select")*/
/*@get("cb_editPlayer_Team")*/

/*@get("lb_editPlayer_Info")*/

/*@get("tf_editPlayer_Surname")*/
/*@get("tf_editPlayer_Lastname")*/
/*@get("tf_editPlayer_Telnr")*/

cb_editPlayer_Select~getItems~Clear()
cb_editPlayer_Team~getItems~Clear()

--lb_editPlayer_Info~Text = ""

tf_editPlayer_Surname~Text = ""
tf_editPlayer_Lastname~Text = ""
tf_editPlayer_Telnr~Text = ""
/*****************************************************************************/

cb_editTeam_Select~getItems~add("-Bitte Ausw"'E4'x"hlen!-")
cb_newPlayer_Team~getItems~add("-Bitte Ausw"'E4'x"hlen!-")

cb_editTeam_Select~getSelectionModel()~select("-Bitte Ausw"'E4'x"hlen!-")
cb_newPlayer_Team~getSelectionModel()~select("-Bitte Ausw"'E4'x"hlen!-")

cb_editPlayer_Select~getItems~add("-Bitte Ausw"'E4'x"hlen!-")
cb_newTeam_Leader~getItems~add("-Bitte Ausw"'E4'x"hlen!-")

cb_editPlayer_Select~getSelectionModel()~select("-Bitte Ausw"'E4'x"hlen!-")
cb_newTeam_Leader~getSelectionModel()~select("-Bitte Ausw"'E4'x"hlen!-")

/***MySQL Conntection***/
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
team = statement~executeQuery(sqlkomm)

check = team~First
DO WHILE check = .true
        IF          (team~getString("TeamName")          <>          "NO          TEAM")          THEN
cb_editTeam_Select~getItems~add(team~getString("TeamName"))
        cb_newPlayer_Team~getItems~add(team~getString("TeamName"))
        cb_editPlayer_Team~getItems~add(team~getString("TeamName"))
        check = team~next
END

sqlkomm = "SELECT * FROM Players ORDER BY Nachname ASC"
player = statement~executeQuery(sqlkomm)
```

```
check = player~First
DO WHILE check = .true
        surname = player~getString("Vorname")
        familyname = player~getString("Nachname")
        fullname = surname familyname
        cb_editPlayer_Select~getItems~add(fullname)
        cb_editTeam_Leader~getItems~add(fullname)
        cb_newTeam_Leader~getItems~add(fullname)
        check = player~next
END

player~Close()
team~Close()
statement~Close()
conn~Close()

btn_connectDB~setVisible(.false)


/**************************************************************************************************
************/

/**************************************************************************************************
************/
  /*                                          ADD TEAM                                              */

/**************************************************************************************************
************/
/**************************************************************************************************
************/

/**********************************/
/** Click Button Saving New Team **/
/**********************************/
::routine klickNewTeamSaveAction public
use arg slotDir

/*@get("cb_newTeam_Leader")*/

/*@get("lb_newTeam_Info")*/

/*@get("tf_newTeam_Name")*/

lb_newTeam_Info~Text = ""
TeamName = tf_newTeam_Name~Text
parse var TeamName TN1 TN2 TN3 .
Leader = cb_newTeam_Leader~Value
SAY Leader
CheckName = .false

IF .environment~DBConnection = .true THEN DO

        IF (TN1 = "") THEN lb_newTeam_Info~Text = "Ung"'FC'x"ltiger Mannschaftsname!"
        ELSE IF (TN1 <> "") THEN DO
                TeamName = TN1
                IF (TN2 <> "" | TN2 <> " ") THEN DO
                        TeamName = TN1 TN2
                        IF (TN3 <> "" | TN3 <> " ") THEN DO
                                TeamName = TN1 TN2 TN3
                        END
                END
                ELSE IF (TN3 <> "" | TN3 <> " ") THEN DO
                        TeamName = TN1 TN3
                END
                say "["TeamName"]"
                CheckName = DBTeamCheckName(TeamName)
                say CheckName

                IF (CheckName = .false) THEN lb_newTeam_Info~Text = "Dieser Mannschaftsname existiert
bereits!"
                ELSE   IF   Leader   =   "-Bitte   Ausw"'E4'x"hlen!-"   THEN   lb_newTeam_Info~Text   =
"Mannschaftsf"'FC'x"hrer Ausw"'E4'x"hlen!"
                ELSE DO
                        spielerID = DBPlayerID(Leader)
```

```
                        /* MySQL Verbindung erstellen */
                        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
                        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance

        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

                        statement = conn~createStatement
                        sqlkomm = "SELECT * FROM Team WHERE (TeamCaptain = "spielerID")"
                        captain = statement~executeQuery(sqlkomm)

                        check = captain~First
                        IF   (check   =   .true)   THEN   lb_newTeam_Info~Text   =   "Spieler   ist
Mannschaftsf"'FC'x"hrer!"'0DC4'x"nderungen konnten nicht vorgenommen werden!"
                        ELSE DO
                                sqlkomm = "INSERT INTO Team (TeamName,TeamCaptain,Punkte,Games,Sets)
VALUES ('"TeamName"',"spielerID",0,0,0)"
                                new = statement~executeUpdate(sqlkomm)
                                teamID = DBTeamID(TeamName)
                                sqlkomm    =    "UPDATE    Players    SET    Team="teamID"    WHERE
idSpieler="spielerID""
                                update = statement~executeUpdate(sqlkomm)
                                lb_newTeam_Info~Text = "'"TeamName"' erfolgreich hinzugef"'FC'x"gt!"
                        END

                        captain~Close()
                        statement~Close()
                        conn~Close()

                        CALL DBConn(slotDir)
                END
        END
END
ELSE lb_newTeam_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"


/*********************************************************************************************
************/

/*********************************************************************************************
************/
  /*                                          EDIT TEAM                                        */

/*********************************************************************************************
************/
/*********************************************************************************************
************/

  /*************************************/
 /**Click Button Editing Existing Team **/
/*************************************/
::routine klickEditTeamAction public
use arg slotDir

/*@get("cb_editTeam_Select")*/
/*@get("cb_editTeam_Leader")*/

/*@get("btn_editTeam")*/
/*@get("btn_editTeam_Save")*/
/*@get("btn_editTeam_Delete")*/
/*@get("btn_editTeam_Cancel")*/

/*@get("lb_editTeam_Select")*/
/*@get("lb_editTeam_Name")*/
/*@get("lb_editTeam_Leader")*/
/*@get("lb_editTeam_Info")*/

/*@get("tf_editTeam_Name")*/

lb_editTeam_Info~Text = ""
ChoosenTeam = cb_editTeam_Select~value

IF .environment~DBConnection = .true THEN DO
        IF (ChoosenTeam = .NIL | ChoosenTeam = "-Bitte Ausw"'E4'x"hlen!-") THEN lb_editTeam_Info~Text
= "Bitte Mannschaft Ausw"'E4'x"hlen!";
```

```
        ELSE DO
                /****************************************/
                cb_editTeam_Leader~setVisible(.true)

                btn_editTeam_Save~setVisible(.true)
                btn_editTeam_Delete~setVisible(.false)
                btn_editTeam~setVisible(.false)
                btn_editTeam_Cancel~setVisible(.true)

                lb_editTeam_Name~setVisible(.true)
                lb_editTeam_Leader~setVisible(.true)

                tf_editTeam_Name~setVisible(.true)
                /****************************************/
                .environment~Leader = DBSearchLeader(ChoosenTeam)
                cb_editTeam_Leader~getSelectionModel()~select(.environment~Leader)
                tf_editTeam_Name~Text = ChoosenTeam
        END
END
ELSE lb_editTeam_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"

/**********************************/
/**Click Button Saving Edited Team */
/**********************************/
::routine klickEditTeamSaveAction public
use arg slotDir

/*@get("cb_editTeam_Select")*/
/*@get("cb_editTeam_Leader")*/

/*@get("btn_editTeam_Save")*/
/*@get("btn_editTeam_Delete")*/
/*@get("btn_editTeam")*/
/*@get("btn_editTeam_Cancel")*/

/*@get("lb_editTeam_Select")*/
/*@get("lb_editTeam_Name")*/
/*@get("lb_editTeam_Leader")*/
/*@get("lb_editTeam_Info")*/

/*@get("tf_editTeam_Name")*/

lb_editTeam_Info~Text = ""
TeamName = tf_editTeam_Name~Text
parse var TeamName TN1 TN2 TN3 .
TeamLeader = cb_editTeam_Leader~Value
.environment~SelectedTeam = cb_editTeam_Select~Value
CheckName = .false

IF (TN1 = "") THEN lb_editTeam_Info~Text = "Ung"'FC'x"ltiger Mannschaftsname!"
ELSE IF (TN1 <> "") THEN DO
        TeamName = TN1
        IF (TN2 <> "" | TN2 <> " ") THEN DO
                TeamName = TN1 TN2
                IF (TN3 <> "" | TN3 <> " ") THEN DO
                        TeamName = TN1 TN2 TN3
                END
        END
        ELSE IF (TN3 <> "" | TN3 <> " ") THEN DO
                TeamName = TN1 TN3
        END
        say "["TeamName"]"
        CheckName = DBTeamCheckName(TeamName)
        SAY CheckName

        IF (CheckName = .false & .environment~SelectedTeam != TeamName) THEN lb_editTeam_Info~Text =
"Dieser Mannschaftsname existiert bereits!"
        ELSE DO
                mistake = .false
                .environment~TName = TeamName
                mistake =  DBSaveEditTeam(TeamLeader)

                IF    (mistake    =    .true)    THEN    lb_editTeam_Info~Text    =    "Spieler    ist
Mannschaftsf"'FC'x"hrer!"'0DC4'x"nderungen konnten nicht vorgenommen werden!"
                ELSE lb_editTeam_Info~Text = "Mannschaft '"TeamName"' erfolgreich bearbeitet!"
```

```
                /*****************************************/
                cb_editTeam_Leader~setVisible(.false)

                btn_editTeam_Save~setVisible(.false)
                btn_editTeam_Delete~setVisible(.true)
                btn_editTeam~setVisible(.true)
                btn_editTeam_Cancel~setVisible(.false)

                lb_editTeam_Name~setVisible(.false)
                lb_editTeam_Leader~setVisible(.false)

                tf_editTeam_Name~setVisible(.false)
                /*****************************************/
                CALL DBConn(slotDir)
        END
END
/*************************************/
/**Click Button Deleting Existing Team**/
/*************************************/
::routine klickEditTeamDeleteAction public
use arg slotDir

/*@get("cb_editTeam_Select")*/
/*@get("cb_editTeam_Leader")*/

/*@get("btn_editTeam_Save")*/
/*@get("btn_editTeam_Delete")*/
/*@get("btn_editTeam_Delete_Cancel")*/
/*@get("btn_editTeam")*/
/*@get("btn_editTeam_Cancel")*/

/*@get("lb_editTeam_Select")*/
/*@get("lb_editTeam_Name")*/
/*@get("lb_editTeam_Leader")*/
/*@get("lb_editTeam_Info")*/

/*@get("tf_editTeam_Name")*/

lb_editTeam_Info~Text = ""
delTeam = cb_editTeam_Select~Value

IF .environment~DBConnection = .true THEN DO
        IF (delTeam = .NIL | delTeam = "-Bitte Daten Laden!-" | delTeam = "-Bitte Ausw"'E4'x"hlen!-")
THEN lb_editTeam_Info~Text = "Bitte eine Mannschaft Ausw"'E4'x"hlen!"
        ELSE IF delTeam = "NO TEAM" THEN lb_editTeam_Info~Text = "'NO TEAM' kann nicht gel"'F6'x"scht
werden!";
        ELSE DO
                IF .environment~delete = .true THEN DO
                        lb_editTeam_Info~Text = "Moechten Sie '"delTeam"' wirklich LOESCHEN!"
                        btn_editTeam_Delete_Cancel~setVisible(.true)
                        btn_editTeam~setVisible(.false)
                        .environment~delete = .false
                END
                ELSE IF .environment~delete = .false THEN DO
                        /* MySQL Verbindung erstellen */
                        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
                        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance

        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

                        statement = conn~createStatement
                        sqlkomm = "SELECT * FROM Team WHERE (TeamName='"delTeam"')"
                        team = statement~executeQuery(sqlkomm)

                        check = team~First
                        IF check = .true THEN teamID = team~getString("idTeam")

                        sqlkomm = "DELETE FROM Team WHERE (idTeam ="teamID")"
                        del = statement~executeUpdate(sqlkomm)

                        sqlkomm = "UPDATE Players SET Team="1" WHERE Team="teamID""
                        update = statement~executeUpdate(sqlkomm)
```

```
                        team~Close()
                        statement~Close()
                        conn~Close()

                        .environment~delete = .true
                        btn_editTeam_Delete_Cancel~setVisible(.false)
                        btn_editTeam~setVisible(.true)
                        lb_editTeam_Info~Text = "'"delTeam"' wurde erfolgreich gel"'F6'x"scht!";
                        CALL DBConn(slotDir)
                END

        END
END
ELSE lb_editTeam_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"
/*********************************************/
/**Click Button Cancel Deleting Existing Team**/
/*********************************************/
::routine klickEditTeamDeleteCancelAction public
use arg slotDir

/*@get("btn_editTeam_Delete_Cancel")*/
/*@get("btn_editTeam")*/
/*@get("lb_editTeam_Info")*/

lb_editTeam_Info~Text = ""
.environment~delete = .true
btn_editTeam_Delete_Cancel~setVisible(.false)
btn_editTeam~setVisible(.true)

/*********************************************/
/**Click Button Cancel Editing Existing Team */
/*********************************************/
::routine klickEditTeamCancelAction public
use arg slotDir

/*@get("cb_editTeam_Select")*/
/*@get("cb_editTeam_Leader")*/

/*@get("btn_editTeam_Save")*/
/*@get("btn_editTeam_Delete")*/
/*@get("btn_editTeam")*/
/*@get("btn_editTeam_Cancel")*/

/*@get("lb_editTeam_Select")*/
/*@get("lb_editTeam_Name")*/
/*@get("lb_editTeam_Leader")*/
/*@get("lb_editTeam_Info")*/

/*@get("tf_editTeam_Name")*/

/*****************************************/
cb_editTeam_Leader~setVisible(.false)

btn_editTeam_Save~setVisible(.false)
btn_editTeam_Delete~setVisible(.true)
btn_editTeam~setVisible(.true)
btn_editTeam_Cancel~setVisible(.false)

lb_editTeam_Name~setVisible(.false)
lb_editTeam_Leader~setVisible(.false)

tf_editTeam_Name~setVisible(.false)
/*****************************************/

lb_editTeam_Info~Text = ""
cb_editTeam_Select~getSelectionModel()~select("-Bitte Ausw"'E4'x"hlen!-")


/*****************************************************************************************************
************/

/*****************************************************************************************************
************/
  /*                                        ADD PLAYER                                                 */
```

```
/*****************************************************************************************************
************/
/*****************************************************************************************************
************/

/*********************************/
/** Click Button Saving New Team **/
/*********************************/
::routine klickNewPlayerSaveAction public
use arg slotDir

/*@get("cb_newPlayer_Team")*/

/*@get("lb_newPlayer_Info")*/

/*@get("tf_newPlayer_Surname")*/
/*@get("tf_newPlayer_Lastname")*/
/*@get("tf_newPlayer_Telnr")*/

lb_newPlayer_Info~Text = ""
Vorname = tf_newPlayer_Surname~Text
parse var Vorname Vorname .
Nachname = tf_newPlayer_Lastname~Text
parse var Nachname Nachname .
Tel = tf_newPlayer_Telnr~Text
parse var Tel Tel .
Mannschaft = cb_newPlayer_Team~Value

IF .environment~DBConnection = .true THEN DO
SELECT
WHEN (Vorname = " " & Nachname = " " & Tel = " " & (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft
= "-Bitte Daten Laden!-")) THEN DO
        tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
        tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
        tf_newPlayer_Telnr~Text = "Bitte TELEFONNR. eingeben!"
        lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
END
WHEN (Vorname = " " & Nachname = " " & Tel = " ") THEN DO
        tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
        tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
        tf_newPlayer_Telnr~Text = "Bitte TELEFONNR. eingeben!"
END
WHEN (Vorname = " " & Nachname = " ") THEN DO
        tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
        tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
END
WHEN (Vorname = " " & Tel = " ") THEN DO
        tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
        tf_newPlayer_Telnr~Text = "Bitte NACHNAME eingeben!"
END
WHEN (Vorname = " " & (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte Daten Laden!-
")) THEN DO
        tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
        lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
END
WHEN (Tel = " " & Nachname = " ") THEN DO
        tf_newPlayer_Telnr~Text = "Bitte VORNAME eingeben!"
        tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
END
WHEN (Tel = " " & (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte Daten Laden!-"))
THEN DO
        tf_newPlayer_Telnr~Text = "Bitte VORNAME eingeben!"
        lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
END
WHEN Vorname = " " THEN tf_newPlayer_Surname~Text = "Bitte VORNAME eingeben!"
WHEN Nachname = " " THEN tf_newPlayer_Lastname~Text = "Bitte NACHNAME eingeben!"
WHEN Tel = " " THEN tf_newPlayer_Telnr~Text = "Bitte TELEFONNR. eingeben!"
WHEN (Mannschaft = "-Bitte Ausw"'E4'x"hlen!-" | Mannschaft = "-Bitte Daten Laden!-") THEN
lb_newPlayer_Info~Text = "Bitte Mannschaft Ausw"'E4'x"hlen!"
OTHERWISE
CheckPlayerName = .false
CheckPlayerName = DBPlayerCheckName(Vorname Nachname)
SAY CheckPlayerName
IF CheckPlayerName = .true THEN DO
```

```
/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "Select * FROM Team WHERE (TeamName = '"Mannschaft"')"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .true THEN teamID = team~getString("idTeam")

statement = conn~createStatement
sqlkomm      =      "INSERT      INTO      Players(Vorname,Nachname,Telefonnummer,Team)      VALUES
('"Vorname"','"Nachname"','"Tel"',"teamID")"
spieler = statement~executeUpdate(sqlkomm)

team~Close()
statement~Close()
conn~Close()

lb_newPlayer_Info~Text = "Spieler '"Vorname Nachname"' wurde erfolgreich hinzugef"'FC'x"gt!"
CALL DBConn(slotDir)
END
ELSE lb_newPlayer_Info~Text = "Spieler existiert bereits!"
END
END
ELSE lb_newPlayer_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"


/*************************************************************************************************
************/

/*************************************************************************************************
************/
  /*                                        EDIT PLAYER                                          */

/*************************************************************************************************
************/
/*************************************************************************************************
************/

  /***************************************/
 /**Click Button Editing Existing Player **/
/***************************************/
::routine klickEditPlayerAction public
use arg slotDir

/*@get("cb_editPlayer_Select")*/
/*@get("cb_editPlayer_Team")*/

/*@get("btn_editPlayer")*/
/*@get("btn_editPlayer_Delete")*/
/*@get("btn_editPlayer_Save")*/
/*@get("btn_editPlayer_Cancel")*/

/*@get("lb_editPlayer_Select")*/
/*@get("lb_editPlayer_Surname")*/
/*@get("lb_editPlayer_Lastname")*/
/*@get("lb_editPlayer_Telnr")*/
/*@get("lb_editPlayer_Team")*/
/*@get("lb_editPlayer_Info")*/

/*@get("tf_editPlayer_Surname")*/
/*@get("tf_editPlayer_Lastname")*/
/*@get("tf_editPlayer_Telnr")*/

lb_editPlayer_Info~Text = ""
ChoosenPlayer = cb_editPlayer_Select~value

IF .environment~DBConnection = .true THEN DO
        IF (ChoosenPlayer = .NIL | ChoosenPlayer = "-Bitte Daten Laden!-" | ChoosenPlayer = "-Bitte
Ausw"'E4'x"hlen!-") THEN lb_editPlayer_Info~Text = "Bitte Spieler Ausw"'E4'x"hlen!";
```

```
        ELSE DO
                /*****************************************/
                cb_editPlayer_Team~setVisible(.true)

                btn_editPlayer_Save~setVisible(.true)
                btn_editPlayer_Delete~setVisible(.false)
                btn_editPlayer~setVisible(.false)
                btn_editPlayer_Cancel~setVisible(.true)

                lb_editPlayer_Surname~setVisible(.true)
                lb_editPlayer_Lastname~setVisible(.true)
                lb_editPlayer_Telnr~setVisible(.true)
                lb_editPlayer_Team~setVisible(.true)

                tf_editPlayer_Surname~setVisible(.true)
                tf_editPlayer_Lastname~setVisible(.true)
                tf_editPlayer_Telnr~setVisible(.true)
                /*****************************************/
                .environment~Team = DBSearchTeam(ChoosenPlayer)
                cb_editPlayer_Team~getSelectionModel()~select(.environment~Team)

                parse var ChoosenPlayer Vorname Nachname

                tf_editPlayer_Surname~Text = Vorname
                tf_editPlayer_Lastname~Text = Nachname
                tf_editPlayer_Telnr~Text = getTelNr(slotDir)
        END
END
ELSE lb_editPlayer_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"

/************************************/
/**Click Button Saving Edited Player */
/************************************/
::routine klickEditPlayerSaveAction public
use arg slotDir

/*@get("cb_editPlayer_Select")*/
/*@get("cb_editPlayer_Team")*/

/*@get("btn_editPlayer")*/
/*@get("btn_editPlayer_Delete")*/
/*@get("btn_editPlayer_Save")*/
/*@get("btn_editPlayer_Cancel")*/

/*@get("lb_editPlayer_Select")*/
/*@get("lb_editPlayer_Surname")*/
/*@get("lb_editPlayer_Lastname")*/
/*@get("lb_editPlayer_Telnr")*/
/*@get("lb_editPlayer_Team")*/
/*@get("lb_editPlayer_Info")*/

/*@get("tf_editPlayer_Surname")*/
/*@get("tf_editPlayer_Lastname")*/
/*@get("tf_editPlayer_Telnr")*/

lb_editPlayer_Info~Text = ""
Vorname = tf_editPlayer_Surname~Text
parse var Vorname Vorname .
Nachname = tf_editPlayer_Lastname~Text
parse var Nachname Nachname .
Fullname = Vorname Nachname
Telefonnr = tf_editPlayer_Telnr~Text
parse var Telefonnr Telefonnr .
Team_Selected = cb_editPlayer_Team~Value
SelectedPlayer = cb_editPlayer_Select~Value

CheckName = .false
CheckName = DBPlayerCheckName(Vorname Nachname)

IF (CheckName = .false & SelectedPlayer <> Fullname) THEN lb_editPlayer_Info~Text = "Dieser Spieler
existiert bereits!"
ELSE IF (Vorname = " " | Vorname = "" | Nachname = " " | Nachname = "" | Telefonnr = " " | Telefonnr
= "") THEN lb_editPlayer_Info~Text = "Ung"'FC'x"ltige Eingaben!";
ELSE DO
```

```
        /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++*/
        /* MySQL Verbindung erstellen */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Team WHERE TeamName='"Team_Selected"'"
        team = statement~executeQuery(sqlkomm)

        check = team~First
        IF check = .true THEN DO
                teamID = team~getString("idTeam")
        END

        parse var SelectedPlayer Vname Nname
        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Players WHERE (Vorname='"Vname"' AND Nachname='"Nname"')"
        spieler = statement~executeQuery(sqlkomm)

        check = spieler~First
        IF check = .true THEN spielerID = spieler~getString("idSpieler")

        IF (.environment~Team = Team_Selected) THEN DO
                sqlkomm = "UPDATE Players SET Vorname='"tf_editPlayer_Surname~Text"', Nachname=
'"tf_editPlayer_Lastname~Text"',Telefonnummer= '"tf_editPlayer_Telnr~Text"', Team="teamID" WHERE
idSpieler="spielerID""
                update = statement~executeUpdate(sqlkomm)
                lb_editPlayer_Info~Text = "Spieler '"tf_editPlayer_Surname~Text
tf_editPlayer_Lastname~Text"' erfolgreich bearbeitet!"
        END
        ELSE DO
                sqlkomm = "SELECT * FROM Team WHERE (TeamCaptain="spielerID")"
                checkLeader = statement~executeQuery(sqlkomm)
                check = checkLeader~First
                IF check = .true THEN lb_editPlayer_Info~Text = "Spieler ist
Mannschaftsf"'FC'x"hrer!"'0DC4'x"nderungen konnten nicht vorgenommen werden!"
                ELSE DO
                        sqlkomm = "UPDATE Players SET Vorname='"tf_editPlayer_Surname~Text"',
Nachname= '"tf_editPlayer_Lastname~Text"',Telefonnummer= '"tf_editPlayer_Telnr~Text"', Team="teamID"
WHERE idSpieler="spielerID""
                        update = statement~executeUpdate(sqlkomm)
                        lb_editPlayer_Info~Text = "Spieler '"tf_editPlayer_Surname~Text
tf_editPlayer_Lastname~Text"' wurde erfolgreich bearbeitet!"
                END
                checkLeader~Close()
        END

        spieler~Close()
        team~Close()
        statement~Close()
        conn~Close()
        /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++*/

        /*******************************************/
        cb_editPlayer_Team~setVisible(.false)

        btn_editPlayer_Save~setVisible(.false)
        btn_editPlayer_Delete~setVisible(.true)
        btn_editPlayer~setVisible(.true)
        btn_editPlayer_Cancel~setVisible(.false)

        lb_editPlayer_Surname~setVisible(.false)
        lb_editPlayer_Lastname~setVisible(.false)
        lb_editPlayer_Telnr~setVisible(.false)
        lb_editPlayer_Team~setVisible(.false)

        tf_editPlayer_Surname~setVisible(.false)
        tf_editPlayer_Lastname~setVisible(.false)
        tf_editPlayer_Telnr~setVisible(.false)
        /*******************************************/
        CALL DBConn(slotDir)
```

```
END
/****************************************/
/**Click Button Deleting Existing Player**/
/****************************************/
::routine klickEditPlayerDeleteAction public
use arg slotDir

/*@get("cb_editPlayer_Select")*/
/*@get("cb_editPlayer_Team")*/

/*@get("btn_editPlayer")*/
/*@get("btn_editPlayer_Delete")*/
/*@get("btn_editPlayer_Delete_Cancel")*/
/*@get("btn_editPlayer_Save")*/
/*@get("btn_editPlayer_Cancel")*/

/*@get("lb_editPlayer_Select")*/
/*@get("lb_editPlayer_Surname")*/
/*@get("lb_editPlayer_Lastname")*/
/*@get("lb_editPlayer_Telnr")*/
/*@get("lb_editPlayer_Team")*/
/*@get("lb_editPlayer_Info")*/

/*@get("tf_editPlayer_Surname")*/
/*@get("tf_editPlayer_Lastname")*/
/*@get("tf_editPlayer_Telnr")*/

lb_editPlayer_Info~Text = ""
delPlayer = cb_editPlayer_Select~Value

IF .environment~DBConnection = .true THEN DO
        IF (delPlayer = .NIL | delPlayer = "-Bitte Daten Laden!-" | delPlayer = "-Bitte
Ausw"'E4'x"hlen!-") THEN lb_editPlayer_Info~Text = "Bitte einen Spieler Ausw"'E4'x"hlen!"
        ELSE DO
                SELECT
                        WHEN .environment~deletePlayer = .true THEN DO
                                lb_editPlayer_Info~Text = "M"'F6'x"chten Sie '"delPlayer"' wirklich
L"'D6'x"SCHEN!"
                                btn_editPlayer_Delete_Cancel~setVisible(.true)
                                btn_editPlayer~setVisible(.false)
                                .environment~deletePlayer = .false
                        END
                        WHEN .environment~deletePlayer = .false THEN DO
                                        /* MySQL Verbindung erstellen */
                                        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load
Java class
                                        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance

        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

                                        parse var delPlayer Vorname Nachname
                                        statement = conn~createStatement
                                        sqlkomm = "SELECT * FROM Players WHERE (Vorname='"Vorname"'
AND Nachname='"Nachname"')"
                                        spieler = statement~executeQuery(sqlkomm)

                                        check = spieler~First
                                        IF check = .true THEN DO

                                                spielerID = spieler~getString("idSpieler")
                                                sqlkomm = "SELECT * FROM Team WHERE (TeamCaptain =
"spielerID")"
                                                checkLeader = statement~executeQuery(sqlkomm)

                                                check = checkLeader~First
                                                IF check = .true THEN lb_editPlayer_Info~Text =
"Spieler ist Mannschaftsf"'FC'x"hrer!"'0DC4'x"nderungen konnten nicht vorgenommen werden!"
                                                ELSE DO
                                                        sqlkomm = "DELETE FROM Players WHERE
(idSpieler ="spielerID")"
                                                        del = statement~executeUpdate(sqlkomm)
                                                        lb_editPlayer_Info~Text = '"delPlayer"'
wurde erfolgreich gel"'F6'x"scht!";
                                                END
```

```
                                        END
                                        spieler~Close
                                        statement~Close
                                        conn~Close

                                        .environment~deletePlayer = .true
                                        btn_editPlayer_Delete_Cancel~setVisible(.false)
                                        btn_editPlayer~setVisible(.true)
                                        CALL DBConn(slotDir)
                                    END
                END
        END
END
ELSE lb_editPlayer_Info~Text = "Bitte zuerst eine Datenbankverbindung herstellen!"

/************************************************/
/*Click Button Cancel DELETE Editing Existing Team */
/************************************************/
::routine klickEditPlayerDeleteCancelAction public
use arg slotDir

/*@get("btn_editPlayer")*/
/*@get("btn_editPlayer_Delete_Cancel")*/
/*@get("lb_editPlayer_Info")*/

lb_editPlayer_Info~Text = ""
.environment~deletePlayer = .true
btn_editPlayer_Delete_Cancel~setVisible(.false)
btn_editPlayer~setVisible(.true)

/*****************************************/
/*Click Button Cancel Editing Existing Team */
/*****************************************/
::routine klickEditPlayerCancelAction public
use arg slotDir

/*@get("tab_editPlayer")*/

/*@get("cb_editPlayer_Select")*/
/*@get("cb_editPlayer_Team")*/

/*@get("btn_editPlayer")*/
/*@get("btn_editPlayer_Delete")*/
/*@get("btn_editPlayer_Save")*/
/*@get("btn_editPlayer_Cancel")*/

/*@get("lb_editPlayer_Select")*/
/*@get("lb_editPlayer_Surname")*/
/*@get("lb_editPlayer_Lastname")*/
/*@get("lb_editPlayer_Telnr")*/
/*@get("lb_editPlayer_Team")*/
/*@get("lb_editPlayer_Info")*/

/*@get("tf_editPlayer_Surname")*/
/*@get("tf_editPlayer_Lastname")*/
/*@get("tf_editPlayer_Telnr")*/

/*****************************************/
cb_editPlayer_Team~setVisible(.false)

btn_editPlayer_Save~setVisible(.false)
btn_editPlayer_Delete~setVisible(.true)
btn_editPlayer~setVisible(.true)
btn_editPlayer_Cancel~setVisible(.false)

lb_editPlayer_Surname~setVisible(.false)
lb_editPlayer_Lastname~setVisible(.false)
lb_editPlayer_Telnr~setVisible(.false)
lb_editPlayer_Team~setVisible(.false)

tf_editPlayer_Surname~setVisible(.false)
tf_editPlayer_Lastname~setVisible(.false)
tf_editPlayer_Telnr~setVisible(.false)
/*****************************************/
```

```
lb_editPlayer_Info~Text = ""
cb_editPlayer_Select~getSelectionModel()~select("-Bitte Ausw"'E4'x"hlen!-")

/**************************************************************************************************
************/

/**************************************************************************************************
************/
  /*                                        Routines                                                */

/**************************************************************************************************
************/
/**************************************************************************************************
************/


/**************************************************************************************************
************/
 /*                          GETTING THE LEADER OF AN EXISTING TEAM                                  */
/**************************************************************************************************
************/
::routine DBSearchLeader public
use arg ChoosenTeam

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (TeamName='"ChoosenTeam"')"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .true THEN leaderID = team~getString("TeamCaptain")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Players WHERE (idSpieler="leaderID")"
spieler = statement~executeQuery(sqlkomm)

check = spieler~First
IF check = .true THEN DO
        Vorname = spieler~getString("Vorname")
        Nachname = spieler~getString("Nachname")
        Name = Vorname Nachname
END

team~Close()
spieler~Close()
statement~Close()
conn~Close()

return Name


/**************************************************************************************************
************/
 /*                          GETTING THE TEAM OF AN EXISTING PLAYER                                  */
/**************************************************************************************************
************/
::routine DBSearchTeam public
use arg ChoosenPlayer

parse var ChoosenPlayer Vorname Nachname

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Players WHERE (Vorname='"Vorname"' AND Nachname='"Nachname"')"
player = statement~executeQuery(sqlkomm)
```

```
check = player~First
IF check = .true THEN teamID = player~getString("Team")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (idTeam="teamID")"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .true THEN TeamName = team~getString("TeamName")

player~Close()
team~Close()
statement~Close()
conn~Close()

return TeamName


/**************************************************************************************************
*************/
 /*                                GETTING THE PLAYERS TELEFONNR                                    */
/**************************************************************************************************
*************/
::routine getTelNr public
use arg slotDir

/*@get("tf_editPlayer_Surname")*/
/*@get("tf_editPlayer_Lastname")*/

Vorname = tf_editPlayer_Surname~Text
Nachname = tf_editPlayer_Lastname~Text

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Players WHERE (Vorname='"Vorname"' AND Nachname='"Nachname"')"
player = statement~executeQuery(sqlkomm)

check = player~First
IF check = .true THEN playerTel = player~getString("Telefonnummer")

player~Close()
statement~Close()
conn~Close()

return playerTel

/**************************************************************************************************
***************/

/**************************************************************************************************
*************/
 /*                                CHECK IF TEAM ALREADY EXISTS                                     */
/**************************************************************************************************
*************/
::routine DBTeamCheckName public
use arg EnteredName

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (TeamName = '"EnteredName"')"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .false THEN checkName = .true
```

```
ELSE checkName = .false

team~Close()
statement~Close()
conn~Close()

return checkName


/**************************************************************************************************
*************/
 /*                              CHECK IF PLAYER ALREADY EXISTS                                    */
/**************************************************************************************************
*************/
::routine DBPlayerCheckName public
use arg Info
parse var Info Vorname Nachname
checkName = .false

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Players WHERE (Vorname = '"Vorname"' AND Nachname = '"Nachname"')"
spieler = statement~executeQuery(sqlkomm)

check = spieler~First
IF check <> .true THEN checkName = .true

spieler~Close()
statement~Close()
conn~Close()

return checkName

/**************************************************************************************************
*****************/


/**************************************************************************************************
*************/
 /*                                  DATABASE SAVE EDIT TEAM                                       */
/**************************************************************************************************
*************/
::routine DBSaveEditTeam public
use arg Info

parse var Info Vorname Nachname
Leader = Vorname Nachname
TeamName = .environment~TName
mistake = .false

spielerID = DBPlayerID(Leader)

/* MySQL Verbindung erstellen */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (TeamName='".environment~SelectedTeam"')"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .true THEN teamID = team~getString("idTeam")

IF (.environment~Leader = Leader) THEN DO
        sqlkomm   =   "UPDATE   Team   SET   TeamName='"TeamName"',   TeamCaptain=   "spielerID"   WHERE
(idTeam="teamID")"
        update = statement~executeUpdate(sqlkomm)
        teamID = DBTeamID(TeamName)
```

```
        sqlkomm = "UPDATE Players SET Team="teamID" WHERE idSpieler="spielerID""
        update = statement~executeUpdate(sqlkomm)
END
ELSE DO
        sqlkomm = "SELECT * FROM Team WHERE (TeamCaptain="spielerID")"
        checkLeader = statement~executeQuery(sqlkomm)
        check = checkLeader~First
        IF check = .true THEN mistake = .true
        ELSE DO
                sqlkomm = "UPDATE Team SET TeamName='"TeamName"', TeamCaptain= "spielerID" WHERE
(idTeam="teamID")"
                update = statement~executeUpdate(sqlkomm)
                teamID = DBTeamID(TeamName)
                sqlkomm = "UPDATE Players SET Team="teamID" WHERE idSpieler="spielerID""
                update = statement~executeUpdate(sqlkomm)
        END
END

team~Close()
statement~Close()
conn~Close()
return mistake

/****************************************************************************************************
*************/
 /*                                  Datebase Get PlayerID                                       */
/****************************************************************************************************
*************/
::routine DBPlayerID public
use arg Leader
parse var Leader Vorname Nachname

/* MySQL Verbindung erstellen */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Players WHERE (Vorname='"Vorname"' AND Nachname='"Nachname"')"
spieler = statement~executeQuery(sqlkomm)

check = spieler~First
IF check = .true THEN spielerID = spieler~getString("idSpieler")

spieler~Close()
statement~Close()
conn~Close()

return spielerID


/****************************************************************************************************
*************/
 /*                                  Datebase Get PlayerID                                       */
/****************************************************************************************************
*************/
::routine DBTeamID public
use arg TeamName

/* MySQL Verbindung erstellen */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (TeamName = '"TeamName"')"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .true THEN teamID = team~getString("idTeam")

team~Close()
statement~Close()
```

```
conn~Close()

return teamID


/*************************************************************************************************
************/
 /*                                    CLEAR BLANK SPACES                                         */
/*************************************************************************************************
************/
::routine KeyReleaseCheck_tf_newTeamName public
use arg slotDir

/*@get("tf_newTeam_Name")*/

IF(tf_newTeam_Name~Text = " ") THEN tf_newTeam_Name~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_editTeamName public
use arg slotDir

/*@get("tf_editTeam_Name")*/

IF(tf_editTeam_Name~Text = " ") THEN tf_editTeam_Name~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_editPlayerSurname public
use arg slotDir

/*@get("tf_editPlayer_Surname")*/

IF(tf_editPlayer_Surname~Text = " ") THEN tf_editPlayer_Surname~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_editPlayerLastname public
use arg slotDir

/*@get("tf_editPlayer_Lastname")*/

IF(tf_editPlayer_Lastname~Text = " ") THEN tf_editPlayer_Lastname~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_editPlayerTelnr public
use arg slotDir

/*@get("tf_editPlayer_Telnr")*/

IF(tf_editPlayer_Telnr~Text = " ") THEN tf_editPlayer_Telnr~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_newPlayerSurname public
use arg slotDir

/*@get("tf_newPlayer_Surname")*/

IF(tf_newPlayer_Surname~Text = " ") THEN tf_newPlayer_Surname~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_newPlayerLastname public
use arg slotDir

/*@get("tf_newPlayer_Lastname")*/

IF(tf_newPlayer_Lastname~Text = " ") THEN tf_newPlayer_Lastname~Text = ""
/*************************************************************/
::routine KeyReleaseCheck_tf_newPlayerTelnr public
use arg slotDir

/*@get("tf_newPlayer_Telnr")*/

IF(tf_newPlayer_Telnr~Text = " ") THEN tf_newPlayer_Telnr~Text = ""
```

Team_Player_GUI.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.String?>
<?import javafx.collections.FXCollections?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ChoiceBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Separator?>
<?import javafx.scene.control.Tab?>
<?import javafx.scene.control.TabPane?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>
<?language rexx?>

<AnchorPane  maxHeight="-Infinity"  maxWidth="-Infinity"  minHeight="-Infinity"  minWidth="-Infinity"
prefHeight="423.0"             prefWidth="688.0"             xmlns="http://javafx.com/javafx/8.0.111"
xmlns:fx="http://javafx.com/fxml/1">

   <fx:script source="controller_team_player.rex" />

   <children>
      <TabPane layoutY="76.0" prefHeight="330.0" prefWidth="688.0" tabClosingPolicy="UNAVAILABLE">
        <tabs>
            <Tab fx:id="tab_newPlayer" text="Spieler Hinzufügen">
               <content>
                  <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
                     <children>
                        <TextField   fx:id="tf_newPlayer_Surname"   layoutX="108.0"   layoutY="34.0"
prefHeight="25.0"              prefWidth="181.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_newPlayerSurname slotDir;" />
                        <TextField   fx:id="tf_newPlayer_Lastname"   layoutX="107.0"   layoutY="87.0"
prefHeight="25.0"              prefWidth="182.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_newPlayerLastname slotDir;" />
                        <TextField    fx:id="tf_newPlayer_Telnr"    layoutX="107.0"    layoutY="140.0"
prefHeight="25.0"              prefWidth="183.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_newPlayerTelnr slotDir;" />
                        <ChoiceBox    fx:id="cb_newPlayer_Team"    layoutX="401.0"    layoutY="34.0"
prefHeight="25.0" prefWidth="217.0" value="-Bitte Daten Laden!-">
                                                            <items>
                                                                    <FXCollections
fx:factory="observableArrayList">
                                                                       <String   fx:value="-Bitte   Daten
Laden!-" />
                                                                    </FXCollections>
                                                            </items>
                                                        </ChoiceBox>
                        <Label     fx:id="lb_newPlayer_Surname"     layoutX="41.0"     layoutY="38.0"
text="Vorname" />
                        <Label     fx:id="lb_newPlayer_Team"     layoutX="323.0"     layoutY="38.0"
text="Mannschaft" />
                        <Button    fx:id="btn_newPlayer_Save"    layoutX="455.0"    layoutY="138.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickNewPlayerSaveAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Speichern" />
                        <Label    fx:id="lb_newPlayer_Lastname"    layoutX="37.0"    layoutY="91.0"
text="Nachname" />
                        <Label     fx:id="lb_newPlayer_Telnr"     layoutX="39.0"     layoutY="144.0"
text="Telefonnr." />
                        <Label     fx:id="lb_newPlayer_Info"     layoutX="37.0"     layoutY="186.0"
prefHeight="38.0" prefWidth="450.0" textFill="#e40808">
                           <font>
                              <Font name="System Bold" size="13.0" />
                           </font>
                        </Label>
                     </children>
                  </AnchorPane>
               </content>
            </Tab>
            <Tab fx:id="tab_editPlayer" text="Spieler Bearbeiten">
               <content>
```

```
                <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="277.0" prefWidth="669.0">
                   <children>
                      <ChoiceBox   fx:id="cb_editPlayer_Select"   layoutX="201.0"   layoutY="27.0"
prefHeight="25.0" prefWidth="226.0" value="-Bitte Daten Laden!-">
                                                          <items>
                                                                 <FXCollections
fx:factory="observableArrayList">
                                                                    <String   fx:value="-Bitte   Daten
Laden!-" />
                                                                 </FXCollections>
                                                          </items>
                                                    </ChoiceBox>
                      <Label     fx:id="lb_editPlayer_Select"     layoutX="55.0"     layoutY="30.0"
text="Spieler Auswählen">
                         <font>
                            <Font name="System Bold" size="13.0" />
                         </font>
                      </Label>
                      <Button      fx:id="btn_editPlayer"      layoutX="202.0"      layoutY="79.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditPlayerAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Bearbeiten" />
                      <Button    fx:id="btn_editPlayer_Delete"    layoutX="326.0"    layoutY="79.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditPlayerDeleteAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Löschen" />
                      <TextField   fx:id="tf_editPlayer_Surname"   layoutX="111.0"   layoutY="125.0"
prefHeight="25.0"              prefWidth="181.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_editPlayerSurname slotDir;" visible="false" />
                      <TextField  fx:id="tf_editPlayer_Lastname"  layoutX="408.0"  layoutY="125.0"
prefHeight="25.0"              prefWidth="216.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_editPlayerLastname slotDir;" visible="false" />
                      <TextField   fx:id="tf_editPlayer_Telnr"   layoutX="112.0"   layoutY="178.0"
prefHeight="25.0"              prefWidth="178.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_editPlayerTelnr slotDir;" visible="false" />
                      <ChoiceBox    fx:id="cb_editPlayer_Team"    layoutX="408.0"    layoutY="178.0"
prefHeight="25.0" prefWidth="217.0" visible="false">
                                                          <items>
                                                                 <FXCollections
fx:factory="observableArrayList">
                                                                    <String    fx:value="Bitte    Daten
Laden!" />
                                                                 </FXCollections>
                                                          </items>
                                                    </ChoiceBox>
                      <Button    fx:id="btn_editPlayer_Save"    layoutX="526.0"    layoutY="223.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditPlayerSaveAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Speichern" visible="false" />
                      <Button    fx:id="btn_editPlayer_Cancel"    layoutX="526.0"    layoutY="262.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditPlayerCancelAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Abbrechen" visible="false" />
                      <Label    fx:id="lb_editPlayer_Surname"    layoutX="51.0"    layoutY="129.0"
text="Vorname" visible="false" />
                      <Label     fx:id="lb_editPlayer_Team"     layoutX="326.0"     layoutY="182.0"
text="Mannschaft" visible="false" />
                      <Label    fx:id="lb_editPlayer_Lastname"    layoutX="328.0"    layoutY="129.0"
text="Nachname" visible="false" />
                      <Label     fx:id="lb_editPlayer_Telnr"     layoutX="49.0"     layoutY="182.0"
text="Telefonnr." visible="false" />
                      <Label     fx:id="lb_editPlayer_Info"     layoutX="67.0"     layoutY="229.0"
prefHeight="38.0" prefWidth="450.0" textFill="#e40808">
                         <font>
                            <Font name="System Bold" size="13.0" />
                         </font>
                      </Label>
                      <Button  fx:id="btn_editPlayer_Delete_Cancel"  layoutX="453.0"  layoutY="79.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditPlayerDeleteCancelAction slotDir /* now process the event, pass on slotDir
*/ ;" prefHeight="25.0" prefWidth="100.0" text="Abbrechen" visible="false" />
                   </children>
                </AnchorPane>
             </content>
          </Tab>
```

```
          <Tab fx:id="tab_newTeam" text="Mannschaft Hinzufügen">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
                     <children>
                        <TextField     fx:id="tf_newTeam_Name"     layoutX="153.0"     layoutY="41.0"
prefHeight="25.0"              prefWidth="224.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_newTeamName slotDir;" />
                        <Label      fx:id="lb_newTeam_Name"       layoutX="31.0"      layoutY="45.0"
text="Mannschaftsname" />
                        <ChoiceBox    fx:id="cb_newTeam_Leader"    layoutX="154.0"    layoutY="96.0"
prefHeight="25.0" prefWidth="222.0" value="-Bitte Daten Laden!-">
                                                            <items>
                                                                   <FXCollections
fx:factory="observableArrayList">
                                                                      <String   fx:value="-Bitte   Daten
Laden!-" />
                                                                   </FXCollections>
                                                            </items>
                                                      </ChoiceBox>
                        <Label     fx:id="lb_newTeam_Leader"     layoutX="30.0"     layoutY="100.0"
text="Mannschaftsführer" />
                        <Button     fx:id="btn_newTeam_Save"     layoutX="414.0"     layoutY="96.0"
mnemonicParsing="false"  onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickNewTeamSaveAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Speichern" />
                        <Label      fx:id="lb_newTeam_Info"      layoutX="30.0"      layoutY="151.0"
prefHeight="38.0" prefWidth="450.0" textFill="#e40808">
                           <font>
                              <Font name="System Bold" size="13.0" />
                           </font>
                        </Label>
                     </children></AnchorPane>
            </content>
          </Tab>
          <Tab fx:id="tab_editTeam" text="Mannschaft Bearbeiten">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="335.0" prefWidth="635.0">
                     <children>
                        <ChoiceBox    fx:id="cb_editTeam_Select"    layoutX="199.0"    layoutY="24.0"
prefHeight="25.0" prefWidth="226.0" value="-Bitte Daten Laden!-">
                                                            <items>
                                                                   <FXCollections
fx:factory="observableArrayList">
                                                                      <String   fx:value="-Bitte   Daten
Laden!-" />
                                                                   </FXCollections>
                                                            </items>
                                                      </ChoiceBox>
                        <Label     fx:id="lb_editTeam_Select"      layoutX="36.0"      layoutY="27.0"
text="Mannschaft Auswählen">
                           <font>
                              <Font name="System Bold" size="13.0" />
                           </font>
                        </Label>
                        <Button       fx:id="btn_editTeam"       layoutX="200.0"       layoutY="85.0"
mnemonicParsing="false"  onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditTeamAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Bearbeiten" />
                        <TextField    fx:id="tf_editTeam_Name"    layoutX="199.0"    layoutY="136.0"
prefHeight="25.0"              prefWidth="352.0"              onKeyReleased="slotDir=arg(arg());call
KeyReleaseCheck_tf_editTeamName slotDir;" visible="false" />
                        <Label      fx:id="lb_editTeam_Name"      layoutX="95.0"      layoutY="140.0"
text="Mannschaftsname" visible="false" />
                        <ChoiceBox    fx:id="cb_editTeam_Leader"    layoutX="200.0"    layoutY="183.0"
prefHeight="25.0" prefWidth="222.0" visible="false">
                                                            <items>
                                                                   <FXCollections
fx:factory="observableArrayList">
                                                                      <String   fx:value="-Bitte   Daten
Laden!-" />
                                                                   </FXCollections>
                                                            </items>
                                                      </ChoiceBox>
                        <Label     fx:id="lb_editTeam_Leader"     layoutX="94.0"     layoutY="187.0"
text="Mannschaftsführer" visible="false" />
```

```
                        <Button     fx:id="btn_editTeam_Save"     layoutX="451.0"     layoutY="183.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditTeamSaveAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Speichern" visible="false" />
                        <Button    fx:id="btn_editTeam_Delete"    layoutX="325.0"    layoutY="85.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditTeamDeleteAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Löschen" />
                        <Label      fx:id="lb_editTeam_Info"      layoutX="16.0"      layoutY="254.0"
prefHeight="38.0" prefWidth="450.0" textFill="#e40808">
                           <font>
                              <Font name="System Bold" size="13.0" />
                           </font></Label>
                        <Button    fx:id="btn_editTeam_Cancel"    layoutX="574.0"    layoutY="183.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditTeamCancelAction slotDir /* now process the event, pass on slotDir */ ;"
prefHeight="25.0" prefWidth="100.0" text="Abbrechen" visible="false" />
                        <Button  fx:id="btn_editTeam_Delete_Cancel"  layoutX="451.0"  layoutY="85.0"
mnemonicParsing="false" onAction="slotDir=arg(arg()) /* always supplied by BSF4ooRexx as the last
argument */ ;call klickEditTeamDeleteCancelAction slotDir /* now process the event, pass on slotDir
*/ ;" prefHeight="25.0" prefWidth="100.0" text="Abbrechen" visible="false" />
                     </children>
                  </AnchorPane>
            </content>
          </Tab>
        </tabs>
      </TabPane>
      <Separator layoutX="-2.0" layoutY="60.0" prefHeight="16.0" prefWidth="688.0" />
      <Label layoutX="359.0" layoutY="22.0" prefHeight="38.0" prefWidth="321.0" text="WinterCup im
Haas-Tenniscenter" textFill="#049ecd">
         <font>
            <Font name="System Bold" size="20.0" />
         </font>
      </Label>
      <Button    fx:id="btn_connectDB"    layoutX="26.0"    layoutY="29.0"    mnemonicParsing="false"
onAction="slotDir=arg(arg())  /*  always  supplied  by  BSF4ooRexx  as  the  last  argument  */  ;call
klickConnectDBAction slotDir /* now process the event, pass on slotDir */ ;" prefHeight="25.0"
prefWidth="200.0" text="Datenbank Verbindung herstellen" />
   </children>
</AnchorPane>
```

Main_spielbericht.rex

```
SAY "PROGRAM START!"
rexxHandler=.rxDocHandler~new

rxApp=BsfCreateRexxProxy(rexxHandler,,"javafx.application.Application")

rxApp~launch(rxApp~getClass, .nil)

::requires "BSF.CLS"

::class RxDocHandler

::method start
  use arg stage

  stage~setTitle("Spielbericht WinterCup im Haas-Tenniscenter")

  rootDocUrl=.bsf~new("java.net.URL", "file:Spielbericht_GUI.fxml")

  root=bsf.loadClass("javafx.fxml.FXMLLoader")~load( rootDocUrl  )
  scene=.bsf~new("javafx.scene.Scene", root)
  stage~setScene(scene)
  stage~show
```

Controller_spielbericht.rex

```
        /*****************************************************************************************************
*******************/

/*****************************************************************************************************
************/
  /*                                          Exportieren                                                */

/*****************************************************************************************************
************/
/*****************************************************************************************************
************/
/****************************************/
/*          RANKING EXPORT              */
/****************************************/
:: routine klickRankingExportAction public
use arg slotDir
/*@get("lb_Info_Export")*/

        lb_Info_Export~Text = ""
        "start /D" directory() "rexxj.cmd Results.rex"

/****************************************/
/*          ALL PLAYER EXPORT           */
/****************************************/
::routine klickAllPlayerExportAction public
use arg slotDir
/*@get("lb_Info_Export")*/

        lb_Info_Export~Text = ""
        "start /D" directory() "rexxj.cmd AlleSpieler.rex"

/****************************************/
/*          ALL TEAMS EXPORT            */
/****************************************/
:: routine klickAllTeamExportAction public
use arg slotDir

/*@get("lb_Info_Export")*/

        lb_Info_Export~Text = ""
        "start /D" directory() "rexxj.cmd AlleMannschaften.rex"

/*********************************************/
/*          RESULTAT SPIELER EXPORT          */
/*********************************************/
::routine klickPlayerExportAction public
use arg slotDir

/*@get("cb_Player_Export")*/
/*@get("btn_Player_Export")*/
/*@get("lb_Info_Export")*/

IF(cb_Player_Export~Value = "-DB Verbindung erstellen!-") THEN lb_Info_Export~Text = "Bitte zuerst
mit der Datenbank verbinden!"
ELSE IF (cb_Player_Export~Value = "Bitte Spieler ausw"'E4'x"hlen!") THEN lb_Info_Export~Text = "Bitte
Spieler ausw"'E4'x"hlen!"
ELSE DO
        lb_Info_Export~Text = ""
        SpielerID = DBPlayerID(cb_Player_Export~Value)
        "start /D" directory() "rexxj.cmd Spieler.rex" SpielerID
        cb_Player_Export~getSelectionModel()~Select("Bitte Spieler ausw"'E4'x"hlen!")
END

/*****************************************************/
/*              RESULTAT MANNSCHAFT EXPORT           */
/*****************************************************/
::routine klickTeamExportAction public
use arg slotDir

/*@get("cb_Team_Export")*/
/*@get("btn_Team_Export")*/
```

```
/*@get("lb_Info_Export")*/

IF(cb_Team_Export~Value = "-DB Verbindung erstellen!-") THEN lb_Info_Export~Text = "Bitte zuerst mit
der Datenbank verbinden!"
ELSE IF (cb_Team_Export~Value = "Bitte Mannschaft ausw"'E4'x"hlen!") THEN lb_Info_Export~Text = "Bitte
Mannschaft ausw"'E4'x"hlen!"
ELSE DO
        lb_Info_Export~Text = ""
        TeamID = DBgetTeamID(cb_Team_Export~Value)
        "start /D" directory() "rexxj.cmd Mannschaft.rex" TeamID
        cb_Team_Export~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
END

        /*****************************************************************************************
*******************/

/*****************************************************************************************
************/
  /*                                    DATABASE CONNECTION                                      */

/*****************************************************************************************
************/
/*****************************************************************************************
************/
:: routine klickConnectDBAction public
use arg slotDir

CALL DBConn(slotDir)

/*****************************************************************************************
************/

/*****************************************************************************************
************/
  /*                                  OPEN TEAM PLAYER PROGRAMM                                  */

/*****************************************************************************************
************/
/*****************************************************************************************
************/
::routine klickOpenTeamPlayerAction public
use arg slotDir
ADDRESS CMD "cd .."
ADDRESS CMD "cd Team_Player"
ADDRESS CMD "rexxj.cmd main_team_player.rex"


/*****************************************************************************************
************/

/*****************************************************************************************
************/
  /*                                       SELECTING TEAMS                                       */

/*****************************************************************************************
************/
/*****************************************************************************************
************/
:: routine klickSelectTeamAction public
use arg slotDir

/*@get("cb_homeTeam")*/
/*@get("cb_awayTeam")*/

/*@get("cb_1singleshome")*/
/*@get("cb_2singleshome")*/
/*@get("cb_3singleshome")*/
/*@get("cb_4singleshome")*/

/*@get("cb_1singlesaway")*/
/*@get("cb_2singlesaway")*/
/*@get("cb_3singlesaway")*/
/*@get("cb_4singlesaway")*/

/*@get("cb_1double1home")*/
```

```
/*@get("cb_1double2home")*/
/*@get("cb_1double1away")*/
/*@get("cb_1double2away")*/

/*@get("cb_2double1home")*/
/*@get("cb_2double2home")*/
/*@get("cb_2double1away")*/
/*@get("cb_2double2away")*/

/*@get("lb_Info")*/

/*@get("btn_selectTeam")*/
/*@get("btn_cancel")*/
/*@get("btn_save")*/

copy_slotDir = slotDir
CALL ClearAll(copy_slotDir)

IF (cb_homeTeam~Value = "-DB Verbindung erstellen!-" | cb_awayTeam~Value = "-DB Verbindung erstellen!-
") THEN lb_Info~Text = "Bitte zuerst mit der Datenbank verbinden!"
ELSE IF (cb_homeTeam~Value = "Bitte Mannschaft ausw"'E4'x"hlen!" | cb_awayTeam~Value = "Bitte
Mannschaft ausw"'E4'x"hlen!") THEN lb_Info~Text = "Bitte Mannschaft ausw"'E4'x"hlen!"
ELSE IF (cb_homeTeam~Value = cb_awayTeam~Value ) THEN lb_Info~Text = "Bitte unterschiedliche
Mannschaften ausw"'E4'x"hlen!"
ELSE DO
        lb_Info~Text = ""
        .environment~homeTeam = cb_homeTeam~Value
        .environment~awayTeam = cb_awayTeam~Value
        homeTeamID = DBgetTeamID(.environment~homeTeam)
        awayTeamID = DBgetTeamID(.environment~awayTeam)

/*****************************************************************************/
        cb_1singleshome~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2singleshome~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_3singleshome~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_4singleshome~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")

        cb_1double1home~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1double2home~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double1home~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double2home~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
/*****************************************************************************/

        /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")
/*****************************************************************************/
        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Players WHERE (Team="homeTeamID") ORDER BY Nachname ASC"
        playerHome = statement~executeQuery(sqlkomm)

        check = playerHome~First
        DO WHILE check = .true
                Vorname = playerHome~getString("Vorname")
                Nachname = playerHome~getString("Nachname")
                Fullname = Vorname Nachname

                cb_1singleshome~getItems~Add(Fullname)
                cb_2singleshome~getItems~Add(Fullname)
                cb_3singleshome~getItems~Add(Fullname)
                cb_4singleshome~getItems~Add(Fullname)

                cb_1double1home~getItems~Add(Fullname)
                cb_1double2home~getItems~Add(Fullname)
                cb_2double1home~getItems~Add(Fullname)
                cb_2double2home~getItems~Add(Fullname)
                check = playerHome~Next
        END

        cb_1singleshome~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1singleshome~getItems~Add("--------------------------------")
        cb_2singleshome~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2singleshome~getItems~Add("--------------------------------")
```

```
        cb_3singleshome~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_3singleshome~getItems~Add("--------------------------------")
        cb_4singleshome~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_4singleshome~getItems~Add("--------------------------------")

        cb_1double1home~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1double1home~getItems~Add("--------------------------------")
        cb_1double2home~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1double2home~getItems~Add("--------------------------------")
        cb_2double1home~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double1home~getItems~Add("--------------------------------")
        cb_2double2home~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double2home~getItems~Add("--------------------------------")
/*****************************************************************************/

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Players WHERE (Team!="homeTeamID") ORDER BY Nachname ASC"
        playerHome = statement~executeQuery(sqlkomm)

        check = playerHome~First
        DO WHILE check = .true
                Vorname = playerHome~getString("Vorname")
                Nachname = playerHome~getString("Nachname")
                Fullname = Vorname Nachname

                cb_1singleshome~getItems~Add(Fullname)
                cb_2singleshome~getItems~Add(Fullname)
                cb_3singleshome~getItems~Add(Fullname)
                cb_4singleshome~getItems~Add(Fullname)

                cb_1double1home~getItems~Add(Fullname)
                cb_1double2home~getItems~Add(Fullname)
                cb_2double1home~getItems~Add(Fullname)
                cb_2double2home~getItems~Add(Fullname)
                check = playerHome~Next
        END
/*****************************************************************************/
        cb_1singlesaway~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2singlesaway~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_3singlesaway~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_4singlesaway~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")

        cb_1double1away~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1double2away~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double1away~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double2away~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")
/*****************************************************************************/
        sqlkomm = "SELECT * FROM Players WHERE (Team="awayTeamID") ORDER BY Nachname ASC"
        playerAway = statement~executeQuery(sqlkomm)

        check = playerAway~First
        DO WHILE check = .true
                Vorname = playerAway~getString("Vorname")
                Nachname = playerAway~getString("Nachname")
                Fullname = Vorname Nachname

                cb_1singlesaway~getItems~Add(Fullname)
                cb_2singlesaway~getItems~Add(Fullname)
                cb_3singlesaway~getItems~Add(Fullname)
                cb_4singlesaway~getItems~Add(Fullname)

                cb_1double1away~getItems~Add(Fullname)
                cb_1double2away~getItems~Add(Fullname)
                cb_2double1away~getItems~Add(Fullname)
                cb_2double2away~getItems~Add(Fullname)
                check = playerAway~Next
        END

        cb_1singlesaway~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1singlesaway~getItems~Add("--------------------------------")
        cb_2singlesaway~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2singlesaway~getItems~Add("--------------------------------")
        cb_3singlesaway~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_3singlesaway~getItems~Add("--------------------------------")
        cb_4singlesaway~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
```

```
        cb_4singlesaway~getItems~Add("--------------------------------")

        cb_1double1away~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1double1away~getItems~Add("--------------------------------")
        cb_1double2away~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_1double2away~getItems~Add("--------------------------------")
        cb_2double1away~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double1away~getItems~Add("--------------------------------")
        cb_2double2away~getSelectionModel~Select("Bitte Spieler ausw"'E4'x"hlen!")
        cb_2double2away~getItems~Add("--------------------------------")
/*****************************************************************************/
sqlkomm = "SELECT * FROM Players WHERE (Team!="awayTeamID") ORDER BY Nachname ASC"
        playerAway = statement~executeQuery(sqlkomm)

        check = playerAway~First
        DO WHILE check = .true
                Vorname = playerAway~getString("Vorname")
                Nachname = playerAway~getString("Nachname")
                Fullname = Vorname Nachname

                cb_1singlesaway~getItems~Add(Fullname)
                cb_2singlesaway~getItems~Add(Fullname)
                cb_3singlesaway~getItems~Add(Fullname)
                cb_4singlesaway~getItems~Add(Fullname)

                cb_1double1away~getItems~Add(Fullname)
                cb_1double2away~getItems~Add(Fullname)
                cb_2double1away~getItems~Add(Fullname)
                cb_2double2away~getItems~Add(Fullname)
                check = playerAway~Next
        END
/*****************************************************************************/
        playerHome~Close()
        playerAway~Close()
        statement~Close()
        conn~Close()
        btn_selectTeam~setVisible(.false)
        btn_cancel~setVisible(.true)
        btn_save~setVisible(.true)
END


/*****************************************************************************************************
*************/

/*****************************************************************************************************
*************/
  /*                                              CANCEL                                               */

/*****************************************************************************************************
*************/
/*****************************************************************************************************
*************/
::routine klickCancelAction public
use arg slotDir

CALL ClearAll(slotDir)


/*****************************************************************************************************
*************/

/*****************************************************************************************************
*************/
  /*                                          SAVING FORMULAR                                          */

/*****************************************************************************************************
*************/
/*****************************************************************************************************
*************/
:: routine klickSaveAction public
use arg slotDir

/*@get("cb_homeTeam")*/
/*@get("cb_awayTeam")*/
```

```
/*@get("cb_1singleshome")*/
/*@get("cb_2singleshome")*/
/*@get("cb_3singleshome")*/
/*@get("cb_4singleshome")*/

/*@get("cb_1singlesaway")*/
/*@get("cb_2singlesaway")*/
/*@get("cb_3singlesaway")*/
/*@get("cb_4singlesaway")*/

/*@get("cb_1double1home")*/
/*@get("cb_1double2home")*/
/*@get("cb_1double1away")*/
/*@get("cb_1double2away")*/

/*@get("cb_2double1home")*/
/*@get("cb_2double2home")*/
/*@get("cb_2double1away")*/
/*@get("cb_2double2away")*/

/*@get("lb_Info")*/
/*@get("cb_round")*/

/*@get("btn_selectTeam")*/
/*@get("btn_cancel")*/
/*@get("btn_save")*/

/*@get("dp_date")*/

IF (cb_1singleshome~Value = "Bitte Spieler ausw"'E4'x"hlen!" | cb_2singleshome~Value = "Bitte Spieler
ausw"'E4'x"hlen!" | cb_3singleshome~Value = "Bitte Spieler ausw"'E4'x"hlen!" | cb_4singleshome~Value
= "Bitte Spieler ausw"'E4'x"hlen!") THEN lb_Info~Text = "Bitte Spieler ausw"'E4'x"hlen"
ELSE IF (cb_1singlesaway~Value = "Bitte Spieler ausw"'E4'x"hlen!" | cb_2singlesaway~Value = "Bitte
Spieler ausw"'E4'x"hlen!" | cb_3singlesaway~Value = "Bitte Spieler ausw"'E4'x"hlen!" |
cb_4singlesaway~Value = "Bitte Spieler ausw"'E4'x"hlen!") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1double1home~Value = "Bitte Spieler ausw"'E4'x"hlen!" | cb_1double2home~Value = "Bitte
Spieler ausw"'E4'x"hlen!" | cb_2double1home~Value = "Bitte Spieler ausw"'E4'x"hlen!" |
cb_2double2home~Value = "Bitte Spieler ausw"'E4'x"hlen!") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1double1away~Value = "Bitte Spieler ausw"'E4'x"hlen!" | cb_1double2away~Value = "Bitte
Spieler ausw"'E4'x"hlen!" | cb_2double1away~Value = "Bitte Spieler ausw"'E4'x"hlen!" |
cb_2double2away~Value = "Bitte Spieler ausw"'E4'x"hlen!") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1singleshome~Value = "--------------------------------" | cb_2singleshome~Value = "----
--------------------------------" | cb_3singleshome~Value = "--------------------------------" |
cb_4singleshome~Value = "--------------------------------") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1singlesaway~Value = "--------------------------------" | cb_2singlesaway~Value = "----
--------------------------------" | cb_3singlesaway~Value = "--------------------------------" |
cb_4singlesaway~Value = "--------------------------------") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1double1home~Value = "--------------------------------" | cb_1double2home~Value = "----
--------------------------------" | cb_2double1home~Value = "--------------------------------" |
cb_2double2home~Value = "--------------------------------") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1double1away~Value = "--------------------------------" | cb_1double2away~Value = "----
--------------------------------" | cb_2double1away~Value = "--------------------------------" |
cb_2double2away~Value = "--------------------------------") THEN lb_Info~Text = "Bitte Spieler
ausw"'E4'x"hlen"
ELSE IF (cb_1singleshome~Value = cb_1singlesaway~Value | cb_2singleshome~Value = cb_2singlesaway~Value
| cb_3singleshome~Value = cb_3singlesaway~Value | cb_4singleshome~Value = cb_4singlesaway~Value) THEN
lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1singleshome~Value = cb_2singleshome~Value | cb_1singleshome~Value = cb_3singleshome~Value
| cb_1singleshome~Value = cb_4singleshome~Value | cb_2singleshome~Value = cb_3singleshome~Value |
cb_2singleshome~Value = cb_4singleshome~Value | cb_3singleshome~Value = cb_4singleshome~Value) THEN
lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1singlesaway~Value = cb_2singlesaway~Value | cb_1singlesaway~Value = cb_3singlesaway~Value
| cb_1singlesaway~Value = cb_4singlesaway~Value | cb_2singlesaway~Value = cb_3singlesaway~Value |
cb_2singlesaway~Value = cb_4singlesaway~Value | cb_3singlesaway~Value = cb_4singlesaway~Value) THEN
lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1singleshome~Value = cb_2singlesaway~Value | cb_1singleshome~Value = cb_3singlesaway~Value
| cb_1singleshome~Value = cb_4singlesaway~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
```

```
ELSE IF (cb_2singleshome~Value = cb_1singlesaway~Value | cb_2singleshome~Value = cb_3singlesaway~Value
| cb_2singleshome~Value = cb_4singlesaway~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_3singleshome~Value = cb_1singlesaway~Value | cb_3singleshome~Value = cb_2singlesaway~Value
| cb_3singleshome~Value = cb_4singlesaway~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_4singleshome~Value = cb_1singlesaway~Value | cb_4singleshome~Value = cb_2singlesaway~Value
| cb_4singleshome~Value = cb_3singlesaway~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1double1home~Value = cb_1double2home~Value | cb_1double1home~Value = cb_2double1home~Value
| cb_1double1home~Value = cb_2double2home~Value | cb_1double1home~Value = cb_1double1away~Value |
cb_1double1home~Value = cb_1double2away~Value | cb_1double1home~Value = cb_2double1away~Value |
cb_1double1home~Value = cb_2double2away~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1double2home~Value = cb_2double1home~Value | cb_1double2home~Value = cb_2double2home~Value
| cb_1double2home~Value = cb_1double1away~Value | cb_1double2home~Value = cb_1double2away~Value |
cb_1double2home~Value = cb_2double1away~Value | cb_1double2home~Value = cb_2double2away~Value) THEN
lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_2double1home~Value = cb_2double2home~Value | cb_2double1home~Value = cb_1double1away~Value
| cb_2double1home~Value = cb_1double2away~Value | cb_2double1home~Value = cb_2double1away~Value |
cb_2double1home~Value = cb_2double2away~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_2double2home~Value = cb_1double1away~Value | cb_2double2home~Value = cb_1double2away~Value
| cb_2double2home~Value = cb_2double1away~Value | cb_2double2home~Value = cb_2double2away~Value) THEN
lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1double1away~Value = cb_1double2away~Value | cb_1double1away~Value = cb_2double1away~Value
| cb_1double1away~Value = cb_2double2away~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (cb_1double2away~Value = cb_2double1away~Value | cb_1double2away~Value = cb_2double2away~Value
| cb_2double1away~Value = cb_2double2away~Value) THEN lb_Info~Text = "Spieler doppelt ausgewaehlt!"
ELSE IF (dp_date~getValue = .Nil) THEN lb_Info~Text = "Bitte Datum Eingeben!"
ELSE DO
        copy_slotDir = slotDir
        CALL CheckInput(copy_slotDir)
END


/**************************************************************************************************
************/

/**************************************************************************************************
************/
  /*                                          Routines                                              */

/**************************************************************************************************
************/
/**************************************************************************************************
************/


/**************************************************************************************************
************/
 /*                                                CLEAR ALL                                        */
/**************************************************************************************************
************/

::routine ClearAll public
use arg slotDir

/*@get("cb_homeTeam")*/
/*@get("cb_awayTeam")*/

/*@get("cb_1singleshome")*/
/*@get("cb_2singleshome")*/
/*@get("cb_3singleshome")*/
/*@get("cb_4singleshome")*/

/*@get("cb_1singlesaway")*/
/*@get("cb_2singlesaway")*/
/*@get("cb_3singlesaway")*/
/*@get("cb_4singlesaway")*/

/*@get("cb_1double1home")*/
/*@get("cb_1double2home")*/
/*@get("cb_1double1away")*/
/*@get("cb_1double2away")*/

/*@get("cb_2double1home")*/
/*@get("cb_2double2home")*/
/*@get("cb_2double1away")*/
/*@get("cb_2double2away")*/
```

```
/*@get("lb_Info")*/
/*@get("cb_round")*/

/*@get("tf_1s1sh")*/
/*@get("tf_2s1sh")*/
/*@get("tf_3s1sh")*/
/*@get("tf_4s1sh")*/

/*@get("tf_1s2sh")*/
/*@get("tf_2s2sh")*/
/*@get("tf_3s2sh")*/
/*@get("tf_4s2sh")*/

/*@get("tf_1s3sh")*/
/*@get("tf_2s3sh")*/
/*@get("tf_3s3sh")*/
/*@get("tf_4s3sh")*/

/*@get("tf_1s1sa")*/
/*@get("tf_2s1sa")*/
/*@get("tf_3s1sa")*/
/*@get("tf_4s1sa")*/

/*@get("tf_1s2sa")*/
/*@get("tf_2s2sa")*/
/*@get("tf_3s2sa")*/
/*@get("tf_4s2sa")*/

/*@get("tf_1s3sa")*/
/*@get("tf_2s3sa")*/
/*@get("tf_3s3sa")*/
/*@get("tf_4s3sa")*/

/*@get("tf_1d1sh")*/
/*@get("tf_2d1sh")*/

/*@get("tf_1d2sh")*/
/*@get("tf_2d2sh")*/

/*@get("tf_1d3sh")*/
/*@get("tf_2d3sh")*/

/*@get("tf_1d1sa")*/
/*@get("tf_2d1sa")*/

/*@get("tf_1d2sa")*/
/*@get("tf_2d2sa")*/

/*@get("tf_1d3sa")*/
/*@get("tf_2d3sa")*/

/*@get("tf_winner")*/
/*@get("tf_result")*/

/*@get("btn_selectTeam")*/
/*@get("btn_cancel")*/
/*@get("btn_save")*/

btn_selectTeam~setVisible(.true)
btn_cancel~setVisible(.false)
btn_save~setVisible(.false)

cb_round~getSelectionModel()~Select(0)

cb_1singleshome~getItems~Clear()
cb_2singleshome~getItems~Clear()
cb_3singleshome~getItems~Clear()
cb_4singleshome~getItems~Clear()

cb_1singlesaway~getItems~Clear()
cb_2singlesaway~getItems~Clear()
cb_3singlesaway~getItems~Clear()
cb_4singlesaway~getItems~Clear()
```

```
cb_1double1home~getItems~Clear()
cb_1double2home~getItems~Clear()
cb_1double1away~getItems~Clear()
cb_1double2away~getItems~Clear()

cb_2double1home~getItems~Clear()
cb_2double2home~getItems~Clear()
cb_2double1away~getItems~Clear()
cb_2double2away~getItems~Clear()

lb_Info~Text = ""

tf_1s1sh~Text = ""
tf_2s1sh~Text = ""
tf_3s1sh~Text = ""
tf_4s1sh~Text = ""

tf_1s2sh~Text = ""
tf_2s2sh~Text = ""
tf_3s2sh~Text = ""
tf_4s2sh~Text = ""

tf_1s3sh~Text = ""
tf_2s3sh~Text = ""
tf_3s3sh~Text = ""
tf_4s3sh~Text = ""

tf_1s1sa~Text = ""
tf_2s1sa~Text = ""
tf_3s1sa~Text = ""
tf_4s1sa~Text = ""

tf_1s2sa~Text = ""
tf_2s2sa~Text = ""
tf_3s2sa~Text = ""
tf_4s2sa~Text = ""

tf_1s3sa~Text = ""
tf_2s3sa~Text = ""
tf_3s3sa~Text = ""
tf_4s3sa~Text = ""

tf_1d1sh~Text = ""
tf_2d1sh~Text = ""

tf_1d2sh~Text = ""
tf_2d2sh~Text = ""

tf_1d3sh~Text = ""
tf_2d3sh~Text = ""

tf_1d1sa~Text = ""
tf_2d1sa~Text = ""

tf_1d2sa~Text = ""
tf_2d2sa~Text = ""

tf_1d3sa~Text = ""
tf_2d3sa~Text = ""

tf_winner~Text = ""
tf_result~Text = ""


/*************************************************************************************************
************/
 /*                                    CHECK INPUT NUMBER                                          */
/*************************************************************************************************
************/
::routine CheckInput public
use arg slotDir

        /*@get("cb_homeTeam")*/
        /*@get("cb_awayTeam")*/
```

```
        /*@get("cb_1singleshome")*/
        /*@get("cb_2singleshome")*/
        /*@get("cb_3singleshome")*/
        /*@get("cb_4singleshome")*/

        /*@get("cb_1singlesaway")*/
        /*@get("cb_2singlesaway")*/
        /*@get("cb_3singlesaway")*/
        /*@get("cb_4singlesaway")*/

        /*@get("cb_1double1home")*/
        /*@get("cb_1double2home")*/
        /*@get("cb_1double1away")*/
        /*@get("cb_1double2away")*/

        /*@get("cb_2double1home")*/
        /*@get("cb_2double2home")*/
        /*@get("cb_2double1away")*/
        /*@get("cb_2double2away")*/

        /*@get("lb_Info")*/
        /*@get("cb_round")*/

        /*@get("tf_1s1sh")*/
        /*@get("tf_2s1sh")*/
        /*@get("tf_3s1sh")*/
        /*@get("tf_4s1sh")*/

        /*@get("tf_1s2sh")*/
        /*@get("tf_2s2sh")*/
        /*@get("tf_3s2sh")*/
        /*@get("tf_4s2sh")*/

        /*@get("tf_1s3sh")*/
        /*@get("tf_2s3sh")*/
        /*@get("tf_3s3sh")*/
        /*@get("tf_4s3sh")*/

        /*@get("tf_1s1sa")*/
        /*@get("tf_2s1sa")*/
        /*@get("tf_3s1sa")*/
        /*@get("tf_4s1sa")*/

        /*@get("tf_1s2sa")*/
        /*@get("tf_2s2sa")*/
        /*@get("tf_3s2sa")*/
        /*@get("tf_4s2sa")*/

        /*@get("tf_1s3sa")*/
        /*@get("tf_2s3sa")*/
        /*@get("tf_3s3sa")*/
        /*@get("tf_4s3sa")*/

        /*@get("tf_1d1sh")*/
        /*@get("tf_2d1sh")*/

        /*@get("tf_1d2sh")*/
        /*@get("tf_2d2sh")*/

        /*@get("tf_1d3sh")*/
        /*@get("tf_2d3sh")*/

        /*@get("tf_1d1sa")*/
        /*@get("tf_2d1sa")*/

        /*@get("tf_1d2sa")*/
        /*@get("tf_2d2sa")*/

        /*@get("tf_1d3sa")*/
        /*@get("tf_2d3sa")*/

        /*@get("tf_winner")*/
        /*@get("tf_result")*/

        /*@get("dp_date")*/
```

```
        /*@get("btn_selectTeam")*/
        /*@get("btn_cancel")*/
        /*@get("btn_save")*/

        V1s1sh = tf_1s1sh~Text
        V2s1sh = tf_2s1sh~Text
        V3s1sh = tf_3s1sh~Text
        V4s1sh = tf_4s1sh~Text

        V1s2sh = tf_1s2sh~Text
        V2s2sh = tf_2s2sh~Text
        V3s2sh = tf_3s2sh~Text
        V4s2sh = tf_4s2sh~Text

        V1s3sh = tf_1s3sh~Text
        V2s3sh = tf_2s3sh~Text
        V3s3sh = tf_3s3sh~Text
        V4s3sh = tf_4s3sh~Text

        V1s1sa = tf_1s1sa~Text
        V2s1sa = tf_2s1sa~Text
        V3s1sa = tf_3s1sa~Text
        V4s1sa = tf_4s1sa~Text

        V1s2sa = tf_1s2sa~Text
        V2s2sa = tf_2s2sa~Text
        V3s2sa = tf_3s2sa~Text
        V4s2sa = tf_4s2sa~Text

        V1s3sa = tf_1s3sa~Text
        V2s3sa = tf_2s3sa~Text
        V3s3sa = tf_3s3sa~Text
        V4s3sa = tf_4s3sa~Text

        V1d1sh = tf_1d1sh~Text
        V2d1sh = tf_2d1sh~Text

        V1d2sh = tf_1d2sh~Text
        V2d2sh = tf_2d2sh~Text

        V1d3sh = tf_1d3sh~Text
        V2d3sh = tf_2d3sh~Text

        V1d1sa = tf_1d1sa~Text
        V2d1sa = tf_2d1sa~Text

        V1d2sa = tf_1d2sa~Text
        V2d2sa = tf_2d2sa~Text

        V1d3sa = tf_1d3sa~Text
        V2d3sa = tf_2d3sa~Text

        lb_Info~Text = ""
        pointsOverallHome = 0
        pointsOverallAway = 0
        setsOverallHome = 0
        setsOverallAway = 0
        .environment~gamesOverallHome = 0
        .environment~gamesOverallAway = 0

        Set1HomeResult = 0
        Set1AwayResult = 0
        Set2HomeResult = 0
        Set2AwayResult = 0
        Set3HomeResult = 0
        Set3AwayResult = 0
        Set4HomeResult = 0
        Set4AwayResult = 0
        Set5HomeResult = 0
        Set5AwayResult = 0
        Set6HomeResult = 0
        Set6AwayResult = 0
```

```
        /*+++++++++++++++++++++++++++++++++++++++++++++++++++++++                1                SET
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
        pointsHomeSet = 0
        pointsAwaySet = 0
        Mistake = .false
        Mistake_1_Set = .false

        IF (V1s1sh = "" | V1s1sa = "" | V1s2sh = "" | V1s2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte
Felder des 1.Satzes ausf"'FC'x"llen! | "
        ELSE IF (V1s1sh = V1s1sa | V1s2sh = V1s2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des
1.Satzes sind falsch! | "
        ELSE DO
                Points = CheckCorrectNumberInput(V1s1sh V1s1sa)
                parse var Points pointsHome pointsAway Mistake

                IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 1.Satz! | "
                ELSE DO
                        pointsHomeSet = pointsHomeSet + pointsHome
                        pointsAwaySet = pointsAwaySet + pointsAway

                        Points = CheckCorrectNumberInput(V1s2sh V1s2sa)
                        parse var Points pointsHome pointsAway Mistake

                        IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 1.Satz! |
"
                        ELSE DO
                                pointsHomeSet = pointsHomeSet + pointsHome
                                pointsAwaySet = pointsAwaySet + pointsAway

                                IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
                                        IF(V1s3sh  =  ""  |  V1s3sa  =  "")  THEN  lb_Info~Text  =
lb_Info~Text"Bitte alle Felder des 1.Satzes ausf"'FC'x"llen! | "
                                        ELSE IF (V1s3sh = V1s3sa) THEN lb_Info~Text = lb_Info~Text"Die
Werte des 1.Satzes sind falsch! | "
                                        ELSE DO
                                                Points    =    CheckCorrectNumberInputTieBreak(V1s3sh
V1s3sa)
                                                parse var Points pointsHome pointsAway Mistake

                                                IF   (Mistake   =   .true)   THEN   lb_Info~Text   =
lb_Info~Text"Eingabefehler 1.Satz! | "
                                                ELSE DO
                                                        pointsHomeSet = pointsHomeSet + pointsHome
                                                        pointsAwaySet = pointsAwaySet + pointsAway
                                                END
                                        END
                                END
                                ELSE IF (V1s3sh <> "" | V1s3sa <> "") THEN DO
                                        lb_Info~Text = lb_Info~Text"Eingabefehler 1.Spiel 3.Satz! | "
                                        Mistake_1_Set = .true
                                END
                        END
                        IF(pointsHomeSet = 2 & Mistake_1_Set = .false) THEN DO
                                        pointsOverallHome = pointsOverallHome + 1
                                        Set1HomeResult = 1
                        END
                        ELSE IF(pointsAwaySet = 2 & Mistake_1_Set = .false) THEN DO
                                        pointsOverallAway = pointsOverallAway +1
                                        Set1AwayResult = 1
                        END
                END
        END

        setsOverallHome = setsOverallHome + pointsHomeSet
        setsOverallAway = setsOverallAway + pointsAwaySet

        /*+++++++++++++++++++++++++++++++++++++++++++++++++++++++                2                SET
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
        pointsHomeSet = 0
        pointsAwaySet = 0
        Mistake = .false
        Mistake_2_Set = .false

        IF (V2s1sh = "" | V2s1sa = "" | V2s2sh = "" | V2s2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte
Felder des 2.Satzes ausf"'FC'x"llen! | "
```

```
        ELSE IF (V2s1sh = V2s1sa | V2s2sh = V2s2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des
2.Satzes sind falsch! | "
        ELSE DO
                Points = CheckCorrectNumberInput(V2s1sh V2s1sa)
                parse var Points pointsHome pointsAway Mistake

                IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 2.Satz! | "
                ELSE DO
                        pointsHomeSet = pointsHomeSet + pointsHome
                        pointsAwaySet = pointsAwaySet + pointsAway

                        Points = CheckCorrectNumberInput(V2s2sh V2s2sa)
                        parse var Points pointsHome pointsAway Mistake

                        IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 2.Satz! |
"
                        ELSE DO
                                pointsHomeSet = pointsHomeSet + pointsHome
                                pointsAwaySet = pointsAwaySet + pointsAway

                                IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
                                        IF(V2s3sh  =  ""  |  V2s3sa  =  "")  THEN  lb_Info~Text  =
lb_Info~Text"Bitte alle Felder des 2.Satzes ausf"'FC'x"llen! | "
                                        ELSE IF (V2s3sh = V2s3sa) THEN lb_Info~Text = lb_Info~Text"Die
Werte des 2.Satzes sind falsch! | "
                                        ELSE DO
                                                Points    =    CheckCorrectNumberInputTieBreak(V2s3sh
V2s3sa)
                                                parse var Points pointsHome pointsAway Mistake

                                                IF   (Mistake   =   .true)   THEN   lb_Info~Text   =
lb_Info~Text"Eingabefehler 2.Satz! | "
                                                ELSE DO
                                                        pointsHomeSet = pointsHomeSet + pointsHome
                                                        pointsAwaySet = pointsAwaySet + pointsAway
                                                END
                                        END
                                END
                                ELSE IF (V2s3sh <> "" | V2s3sa <> "") THEN DO
                                        lb_Info~Text = lb_Info~Text"Eingabefehler 2.Spiel 3.Satz! | "
                                        Mistake_2_Set = .true
                                END
                        END
                        IF(pointsHomeSet = 2 & Mistake_2_Set = .false) THEN DO
                                        pointsOverallHome = pointsOverallHome + 1
                                        Set2HomeResult = 1
                        END
                        ELSE IF(pointsAwaySet = 2 & Mistake_2_Set = .false) THEN DO
                                        pointsOverallAway = pointsOverallAway +1
                                        Set2AwayResult = 1
                        END
                END
        END

        setsOverallHome = setsOverallHome + pointsHomeSet
        setsOverallAway = setsOverallAway + pointsAwaySet

        /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++                3                SET
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
        pointsHomeSet = 0
        pointsAwaySet = 0
        Mistake = .false
        Mistake_3_Set = .false

        IF (V3s1sh = "" | V3s1sa = "" | V3s2sh = "" | V3s2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte
Felder des 3.Satzes ausf"'FC'x"llen!"'0D'x""
        ELSE IF (V3s1sh = V3s1sa | V3s2sh = V3s2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des
3.Satzes sind falsch!"'0D'x""
        ELSE DO
                Points = CheckCorrectNumberInput(V3s1sh V3s1sa)
                parse var Points pointsHome pointsAway Mistake

                IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 3.Satz!1"'0D'x""
                ELSE DO
                        pointsHomeSet = pointsHomeSet + pointsHome
```

```
                        pointsAwaySet = pointsAwaySet + pointsAway

                        Points = CheckCorrectNumberInput(V3s2sh V3s2sa)
                        parse var Points pointsHome pointsAway Mistake

                        IF  (Mistake  =  .true)  THEN  lb_Info~Text  =  lb_Info~Text"Eingabefehler
3.Satz!"'0D'x""
                        ELSE DO
                                pointsHomeSet = pointsHomeSet + pointsHome
                                pointsAwaySet = pointsAwaySet + pointsAway

                                IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
                                        IF(V3s3sh  =  ""  |  V3s3sa  =  "")  THEN  lb_Info~Text  =
lb_Info~Text"Bitte alle Felder des 3.Satzes ausf"'FC'x"llen!"'0D'x""
                                        ELSE IF (V3s3sh = V3s3sa) THEN lb_Info~Text = lb_Info~Text"Die
Werte des 3.Satzes sind falsch!"'0D'x""
                                        ELSE DO
                                                Points   =   CheckCorrectNumberInputTieBreak(V3s3sh
V3s3sa)
                                                parse var Points pointsHome pointsAway Mistake

                                                IF   (Mistake   =   .true)   THEN   lb_Info~Text   =
lb_Info~Text"Eingabefehler 3.Satz!"'0D'x""
                                                ELSE DO
                                                        pointsHomeSet = pointsHomeSet + pointsHome
                                                        pointsAwaySet = pointsAwaySet + pointsAway
                                                END
                                        END
                                END
                                ELSE IF (V3s3sh <> "" | V3s3sa <> "") THEN DO
                                        lb_Info~Text    =    lb_Info~Text"Eingabefehler    3.Spiel
3.Satz!"'0D'x""
                                        Mistake_3_Set = .true
                                END
                        END
                        IF(pointsHomeSet = 2 & Mistake_3_Set = .false) THEN DO
                                        pointsOverallHome = pointsOverallHome + 1
                                        Set3HomeResult = 1
                        END
                        ELSE IF(pointsAwaySet = 2  & Mistake_3_Set = .false) THEN DO
                                        pointsOverallAway = pointsOverallAway +1
                                        Set3AwayResult = 1
                        END
                END
        END

        setsOverallHome = setsOverallHome + pointsHomeSet
        setsOverallAway = setsOverallAway + pointsAwaySet

        /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++                4                SET
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
        pointsHomeSet = 0
        pointsAwaySet = 0
        Mistake = .false
        Mistake_4_Set = .false

        IF (V4s1sh = "" | V4s1sa = "" | V4s2sh = "" | V4s2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte
Felder des 4.Satzes ausf"'FC'x"llen! | "
        ELSE IF (V4s1sh = V4s1sa | V4s2sh = V4s2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des
4.Satzes sind falsch! | "
        ELSE DO
                Points = CheckCorrectNumberInput(V4s1sh V4s1sa)
                parse var Points pointsHome pointsAway Mistake

                IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 4.Satz! | "
                ELSE DO
                        pointsHomeSet = pointsHomeSet + pointsHome
                        pointsAwaySet = pointsAwaySet + pointsAway

                        Points = CheckCorrectNumberInput(V4s2sh V4s2sa)
                        parse var Points pointsHome pointsAway Mistake

                        IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 4.Satz! |
"
                        ELSE DO
```

```
                                pointsHomeSet = pointsHomeSet + pointsHome
                                pointsAwaySet = pointsAwaySet + pointsAway

                                IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
                                        IF(V4s3sh  =  ""  |  V4s3sa  =  "")  THEN  lb_Info~Text  =
lb_Info~Text"Bitte alle Felder des 4.Satzes ausf"'FC'x"llen! | "
                                        ELSE IF (V4s3sh = V4s3sa) THEN lb_Info~Text = lb_Info~Text"Die
Werte des 4.Satzes sind falsch! | "
                                        ELSE DO
                                                Points   =   CheckCorrectNumberInputTieBreak(V4s3sh
V4s3sa)
                                                parse var Points pointsHome pointsAway Mistake

                                                IF   (Mistake   =   .true)   THEN   lb_Info~Text   =
lb_Info~Text"Eingabefehler 4.Satz! | "
                                                ELSE DO
                                                        pointsHomeSet = pointsHomeSet + pointsHome
                                                        pointsAwaySet = pointsAwaySet + pointsAway
                                                END
                                        END
                                END
                                ELSE IF (V4s3sh <> "" | V4s3sa <> "") THEN DO
                                        lb_Info~Text = lb_Info~Text"Eingabefehler 4.Spiel 3.Satz! | "
                                        Mistake_4_Set = .true
                                END
                        END
                        IF(pointsHomeSet = 2  & Mistake_4_Set = .false) THEN DO
                                        pointsOverallHome = pointsOverallHome + 1
                                        Set4HomeResult = 1
                                END
                        ELSE IF(pointsAwaySet = 2  & Mistake_4_Set = .false) THEN DO
                                        pointsOverallAway = pointsOverallAway +1
                                        Set4AwayResult = 1
                                END
                END
        END

        setsOverallHome = setsOverallHome + pointsHomeSet
        setsOverallAway = setsOverallAway + pointsAwaySet

        /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++                5                SET
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
        pointsHomeSet = 0
        pointsAwaySet = 0
        Mistake = .false
        Mistake_5_Set = .false

        IF (V1d1sh = "" | V1d1sa = "" | V1d2sh = "" | V1d2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte
Felder des 5.Satzes ausf"'FC'x"llen! | "
        ELSE IF (V1d1sh = V1d1sa | V1d2sh = V1d2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des
5.Satzes sind falsch! | "
        ELSE DO
                Points = CheckCorrectNumberInput(V1d1sh V1d1sa)
                parse var Points pointsHome pointsAway Mistake

                IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 5.Satz! | "
                ELSE DO
                        pointsHomeSet = pointsHomeSet + pointsHome
                        pointsAwaySet = pointsAwaySet + pointsAway

                        Points = CheckCorrectNumberInput(V1d2sh V1d2sa)
                        parse var Points pointsHome pointsAway Mistake

                        IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 5.Satz! |
"
                        ELSE DO
                                pointsHomeSet = pointsHomeSet + pointsHome
                                pointsAwaySet = pointsAwaySet + pointsAway

                                IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
                                        IF(V1d3sh  =  ""  |  V1d3sa  =  "")  THEN  lb_Info~Text  =
lb_Info~Text"Bitte alle Felder des 5.Satzes ausf"'FC'x"llen! | "
                                        ELSE IF (V1d3sh = V1d3sa) THEN lb_Info~Text = lb_Info~Text"Die
Werte des 5.Satzes sind falsch! | "
                                        ELSE DO
```

```
                                                Points    =    CheckCorrectNumberInputTieBreak(V1d3sh
V1d3sa)
                                                parse var Points pointsHome pointsAway Mistake

                                                IF   (Mistake   =   .true)   THEN   lb_Info~Text   =
lb_Info~Text"Eingabefehler 5.Satz! | "
                                                ELSE DO
                                                        pointsHomeSet = pointsHomeSet + pointsHome
                                                        pointsAwaySet = pointsAwaySet + pointsAway
                                                END
                                        END
                                ELSE IF (V1d3sh <> "" | V1d3sa <> "") THEN DO
                                        lb_Info~Text = lb_Info~Text"Eingabefehler 5.Spiel 3.Satz! | "
                                        Mistake_5_Set = .true
                                END
                        END
                        IF(pointsHomeSet = 2  & Mistake_5_Set = .false) THEN DO
                                        pointsOverallHome = pointsOverallHome + 1
                                        Set5HomeResult = 1
                        END
                        ELSE IF(pointsAwaySet = 2 & Mistake_5_Set = .false) THEN DO
                                        pointsOverallAway = pointsOverallAway +1
                                        Set5AwayResult = 1
                        END
                END
        END

        setsOverallHome = setsOverallHome + pointsHomeSet
        setsOverallAway = setsOverallAway + pointsAwaySet

        /*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++                6                SET
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
        pointsHomeSet = 0
        pointsAwaySet = 0
        Mistake = .false
        Mistake_6_Set = .false

        IF (V2d1sh = "" | V2d1sa = "" | V2d2sh = "" | V2d2sa = "") THEN lb_Info~Text = lb_Info~Text"Bitte
Felder des 6.Satzes ausf"'FC'x"llen!"'0D'x""
        ELSE IF (V2d1sh = V2d1sa | V2d2sh = V2d2sa) THEN lb_Info~Text = lb_Info~Text"Die Werte des
6.Satzes sind falsch!"'0D'x""
        ELSE DO
                Points = CheckCorrectNumberInput(V2d1sh V2d1sa)
                parse var Points pointsHome pointsAway Mistake

                IF (Mistake = .true) THEN lb_Info~Text = lb_Info~Text"Eingabefehler 6.Satz!"'0D'x""
                ELSE DO
                        pointsHomeSet = pointsHomeSet + pointsHome
                        pointsAwaySet = pointsAwaySet + pointsAway

                        Points = CheckCorrectNumberInput(V2d2sh V2d2sa)
                        parse var Points pointsHome pointsAway Mistake

                        IF  (Mistake  =  .true)  THEN  lb_Info~Text  =  lb_Info~Text"Eingabefehler
6.Satz!"'0D'x""
                        ELSE DO
                                pointsHomeSet = pointsHomeSet + pointsHome
                                pointsAwaySet = pointsAwaySet + pointsAway

                                IF ((pointsHomeSet - pointsAwaySet) = 0) THEN DO
                                        IF(V2d3sh  =  ""  |  V2d3sa  =  "")  THEN  lb_Info~Text  =
lb_Info~Text"Bitte alle Felder des 6.Satzes ausf"'FC'x"llen!"'0D'x""
                                        ELSE IF (V2d3sh = V2d3sa) THEN lb_Info~Text = lb_Info~Text"Die
Werte des 6.Satzes sind falsch!"'0D'x""
                                        ELSE DO
                                                Points    =    CheckCorrectNumberInputTieBreak(V2d3sh
V2d3sa)
                                                parse var Points pointsHome pointsAway Mistake

                                                IF   (Mistake   =   .true)   THEN   lb_Info~Text   =
lb_Info~Text"Eingabefehler 6.Satz!"'0D'x""
                                                ELSE DO
                                                        pointsHomeSet = pointsHomeSet + pointsHome
                                                        pointsAwaySet = pointsAwaySet + pointsAway
```

```
                                        END
                                END
                        END
                        ELSE IF (V2d3sh <> "" | V2d3sa <> "") THEN DO
                                lb_Info~Text     =     lb_Info~Text"Eingabefehler     6.Spiel
3.Satz!"'0D'x""
                                Mistake_6_Set = .true
                        END
                END
                IF(pointsHomeSet = 2 & Mistake_6_Set = .false) THEN DO
                        pointsOverallHome = pointsOverallHome + 1
                        Set6HomeResult = 1
                END
                ELSE IF(pointsAwaySet = 2 & Mistake_6_Set = .false) THEN DO
                        pointsOverallAway = pointsOverallAway +1
                        Set6AwayResult = 1
                END
        END
END

setsOverallHome = setsOverallHome + pointsHomeSet
setsOverallAway = setsOverallAway + pointsAwaySet

IF ((pointsOverallHome + pointsOverallAway) = 6) THEN DO

/***MySQL Conntection***/
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")
/*****************************/
statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (TeamName = '".environment~homeTeam"')"
team = statement~executeQuery(sqlkomm)
check = team~first
IF check = .true THEN DO
        IDHome = team~getString("idTeam")
        PunkteHome = team~getString("Punkte")
        GamesHome = team~getString("Games")
        SetsHome = team~getString("Sets")
END
statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE TeamName = '".environment~awayTeam"'"
team = statement~executeQuery(sqlkomm)
check = team~first
IF check = .true THEN DO
        IDAway = team~getString("idTeam")
        PunkteAway = team~getString("Punkte")
        GamesAway = team~getString("Games")
        SetsAway = team~getString("Sets")
END
team~Close()
/*******************************/
SetsHome = SetsHome + setsOverallHome
SetsAway = SetsAway + setsOverallAway


        IF ((pointsOverallHome = 3) & (pointsOverallAway = 3)) THEN DO
                tf_winner~Text = "Unentschieden!"
                tf_result~Text = "3  :  3"
                PunkteHome = PunkteHome + 1
                PunkteAway = PunkteAway + 1
                sqlkomm             =             "UPDATE             Team             SET
Punkte="PunkteHome",Games=".environment~gamesOverallHome",Sets="SetsHome" WHERE idTeam="IDHome""
                update = statement~executeUpdate(sqlkomm)
                sqlkomm             =             "UPDATE             Team             SET
Punkte="PunkteAway",Games=".environment~gamesOverallAway",Sets="SetsAway" WHERE idTeam="IDAway""
                update = statement~executeUpdate(sqlkomm)
        END
        ELSE IF (pointsOverallHome > pointsOverallAway) THEN DO
                tf_winner~Text = .environment~homeTeam
                tf_result~Text = pointsOverallHome"  :  "pointsOverallAway
                PunkteHome = PunkteHome + 2
                sqlkomm             =             "UPDATE             Team             SET
Punkte="PunkteHome",Games=".environment~gamesOverallHome",Sets="SetsHome" WHERE idTeam="IDHome""
```

```
                        update = statement~executeUpdate(sqlkomm)
                        sqlkomm             =             "UPDATE             Team             SET
Games=".environment~gamesOverallAway",Sets="SetsAway" WHERE idTeam="IDAway""
                        update = statement~executeUpdate(sqlkomm)
                END
                ELSE IF (pointsOverallHome < pointsOverallAway) THEN DO
                        tf_winner~Text = .environment~awayTeam
                        tf_result~Text = pointsOverallHome"  :  "pointsOverallAway
                        PunkteAway = PunkteAway + 2
                        sqlkomm             =             "UPDATE             Team             SET
Games=".environment~gamesOverallHome",Sets="SetsHome" WHERE idTeam="IDHome""
                        update = statement~executeUpdate(sqlkomm)
                        sqlkomm             =             "UPDATE             Team             SET
Punkte="PunkteAway",Games=".environment~gamesOverallAway",Sets="SetsAway" WHERE idTeam="IDAway""
                        update = statement~executeUpdate(sqlkomm)
                END

/*************************************************************************************************/
        spielerID11 = DBPlayerID(cb_1singleshome~Value)
        spielerID12 = DBPlayerID(cb_1singlesaway~Value)
        spielerID21 = DBPlayerID(cb_2singleshome~Value)
        spielerID22 = DBPlayerID(cb_2singlesaway~Value)
        spielerID31 = DBPlayerID(cb_3singleshome~Value)
        spielerID32 = DBPlayerID(cb_3singlesaway~Value)
        spielerID41 = DBPlayerID(cb_4singleshome~Value)
        spielerID42 = DBPlayerID(cb_4singlesaway~Value)

        teamID1 = DBgetTeamID(cb_homeTeam~Value)
        teamID2 = DBgetTeamID(cb_awayTeam~Value)

statement_Spiele = conn~createStatement

IF(V1s3sh = "" | V1s3sa = "") THEN DO
                sqlkomm_1Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)  Values
("spielerID11","teamID1","spielerID12","teamID2","V1s1sh","V1s1sa","V1s2sh","V1s2sa","cb_round~Value
",'"dp_date~getValue~toString"',"Set1HomeResult","Set1AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_1Satz)
        END
        ELSE DO
                sqlkomm_1Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)
Values
("spielerID11","teamID1","spielerID12","teamID2","V1s1sh","V1s1sa","V1s2sh","V1s2sa","V1s3sh","V1s3s
a","cb_round~Value",'"dp_date~getValue~toString"',"Set1HomeResult","Set1AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_1Satz)
        END
/*************************************************************************************************/
IF(V2s3sh = "" | V2s3sa = "") THEN DO
                sqlkomm_2Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)  Values
("spielerID21","teamID1","spielerID22","teamID2","V2s1sh","V2s1sa","V2s2sh","V2s2sa","cb_round~Value
",'"dp_date~getValue~toString"',"Set2HomeResult","Set2AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_2Satz)
        END
        ELSE DO
                sqlkomm_2Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)
Values
("spielerID21","teamID1","spielerID22","teamID2","V2s1sh","V2s1sa","V2s2sh","V2s2sa","V2s3sh","V2s3s
a","cb_round~Value",'"dp_date~getValue~toString"',"Set2HomeResult","Set2AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_2Satz)
        END
/*************************************************************************************************/

IF(V3s3sh = "" | V3s3sa = "") THEN DO
                sqlkomm_3Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)  Values
("spielerID31","teamID1","spielerID32","teamID2","V3s1sh","V3s1sa","V3s2sh","V3s2sa","cb_round~Value
",'"dp_date~getValue~toString"',"Set3HomeResult","Set3AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_3Satz)
        END
        ELSE DO
```

```
                sqlkomm_3Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)
Values
("spielerID31","teamID1","spielerID32","teamID2","V3s1sh","V3s1sa","V3s2sh","V3s2sa","V3s3sh","V3s3s
a","cb_round~Value",'"dp_date~getValue~toString"',"Set3HomeResult","Set3AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_3Satz)
        END
/*************************************************************************************************/

IF(V4s3sh = "" | V4s3sa = "") THEN DO
                sqlkomm_4Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2) Values
("spielerID41","teamID1","spielerID42","teamID2","V4s1sh","V4s1sa","V4s2sh","V4s2sa","cb_round~Value
",'"dp_date~getValue~toString"',"Set4HomeResult","Set4AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_4Satz)
        END
        ELSE DO
                sqlkomm_4Satz   =   "insert   into   4440_3.Singles   (`ID-Spieler1`,`ID-Team1`,`ID-
Spieler2`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)
Values
("spielerID41","teamID1","spielerID42","teamID2","V4s1sh","V4s1sa","V4s2sh","V4s2sa","V4s3sh","V4s3s
a","cb_round~Value",'"dp_date~getValue~toString"',"Set4HomeResult","Set4AwayResult")"
                singles = statement_Spiele~executeUpdate(sqlkomm_4Satz)
        END
/*************************************************************************************************/

        spielerID11 = DBPlayerID(cb_1double1home~Value)
        spielerID12 = DBPlayerID(cb_1double2home~Value)
        spielerID21 = DBPlayerID(cb_1double1away~Value)
        spielerID22 = DBPlayerID(cb_1double2away~Value)
        spielerID31 = DBPlayerID(cb_2double1home~Value)
        spielerID32 = DBPlayerID(cb_2double2home~Value)
        spielerID41 = DBPlayerID(cb_2double1away~Value)
        spielerID42 = DBPlayerID(cb_2double2away~Value)
/*************************************************************************************************/

IF(V1d3sh = "" | V1d3sa = "") THEN DO
                sqlkomm_5Satz  =  "insert  into  4440_3.Double    (`ID-Spieler1`,`ID-Spieler2`,`ID-
Team1`,`ID-Spieler3`,`ID-Spieler4`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)               Values
("spielerID11","spielerID12","teamID1","spielerID21","spielerID22","teamID2","V1d1sh","V1d1sa","V1d2
sh","V1d2sa","cb_round~Value",'"dp_date~getValue~toString"',"Set5HomeResult","Set5AwayResult")"
                double = statement_Spiele~executeUpdate(sqlkomm_5Satz)
        END
        ELSE DO
                sqlkomm_5Satz  =  "insert  into  4440_3.Double    (`ID-Spieler1`,`ID-Spieler2`,`ID-
Team1`,`ID-Spieler3`,`ID-Spieler4`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)
Values
("spielerID11","spielerID12","teamID1","spielerID21","spielerID22","teamID2","V1d1sh","V1d1sa","V1d2
sh","V1d2sa","V1d3sh","V1d3sa","cb_round~Value",'"dp_date~getValue~toString"',"Set5HomeResult","Set5
AwayResult")"
                double = statement_Spiele~executeUpdate(sqlkomm_5Satz)
        END
/*************************************************************************************************/

IF(V2d3sh = "" | V2d3sa = "") THEN DO
                sqlkomm_6Satz  =  "insert  into  4440_3.Double    (`ID-Spieler1`,`ID-Spieler2`,`ID-
Team1`,`ID-Spieler3`,`ID-Spieler4`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)               Values
("spielerID31","spielerID32","teamID1","spielerID41","spielerID42","teamID2","V2d1sh","V2d1sa","V2d2
sh","V2d2sa","cb_round~Value",'"dp_date~getValue~toString"',"Set5HomeResult","Set5AwayResult")"
                double = statement_Spiele~executeUpdate(sqlkomm_6Satz)
        END
        ELSE DO
                sqlkomm_6Satz  =  "insert  into  4440_3.Double    (`ID-Spieler1`,`ID-Spieler2`,`ID-
Team1`,`ID-Spieler3`,`ID-Spieler4`,`ID-
Team2`,Satz1SP1,Satz1SP2,Satz2SP1,Satz2SP2,Satz3SP1,Satz3SP2,Runde,Datum,ErgebnisSP1,ErgebnisSP2)
Values
("spielerID31","spielerID32","teamID1","spielerID41","spielerID42","teamID2","V2d1sh","V2d1sa","V2d2
sh","V2d2sa","V2d3sh","V2d3sa","cb_round~Value",'"dp_date~getValue~toString"',"Set5HomeResult","Set5
AwayResult")"
                double = statement_Spiele~executeUpdate(sqlkomm_6Satz)
```

```
        END
/*************************************************************************************************/
        statement_Spiele~Close()
        statement~Close()
        conn~Close()

        btn_selectTeam~setVisible(.true)
        btn_cancel~setVisible(.false)
        btn_save~setVisible(.false)
        lb_Info~Text = "Spiel erfolgreich gespeichert!"
END
ELSE lb_Info~Text = lb_Info~Text"Fehler bei der Spieleingabe!"


/*************************************************************************************************
*************/
 /*                                    CHECK CORRECT NUMBER INPUT                                 */
/*************************************************************************************************
*************/
:: routine CheckCorrectNumberInput public
use arg Points
parse Var Points Home Away
pointsHome = 0
pointsAway = 0
mistake = .false
IF(Home = 7 ) THEN DO
                IF(Away < 7) THEN pointsHome = 1
                ELSE mistake = .true
        END
ELSE IF(Away = 7 ) THEN DO
                IF(Home < 7) THEN pointsAway = 1
                ELSE mistake = .true
        END
ELSE IF (Home = 6) THEN DO
                IF (Away < 5) THEN pointsHome = 1
                ELSE mistake = .true
        END
ELSE IF (Away = 6) THEN DO
        IF (Home < 5) THEN pointsAway = 1
        ELSE mistake = .true
END
ELSE mistake = .true
.environment~gamesOverallHome = .environment~gamesOverallHome + Home
.environment~gamesOverallAway = .environment~gamesOverallAway + Away
return pointsHome pointsAway mistake


/*************************************************************************************************
*************/
 /*                              CHECK CORRECT NUMBER INPUT TIE BREAK                             */
/*************************************************************************************************
*************/
:: routine CheckCorrectNumberInputTieBreak public
use arg Points
parse Var Points Home Away
pointsHome = 0
pointsAway = 0
mistake = .false

IF((Home = 10) & (Away < 9)) THEN pointsHome = 1
ELSE IF ((Home > 10) & ((Home - Away) = 2)) THEN pointsHome = 1
ELSE IF((Away = 10) & (Home < 9)) THEN pointsAway = 1
ELSE IF((Away > 10) & ((Away - Home) = 2)) THEN pointsAway = 1
ELSE mistake = .true
.environment~gamesOverallHome = .environment~gamesOverallHome + Home
.environment~gamesOverallAway = .environment~gamesOverallAway + Away
return pointsHome pointsAway mistake


/*************************************************************************************************
*************/
 /*                                    GETTING THE ID OF A TEAM                                   */
/*************************************************************************************************
*************/
::routine DBgetTeamID public
```

```
use arg TeamName

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team WHERE (TeamName='"TeamName"')"
team = statement~executeQuery(sqlkomm)

check = team~First
IF check = .true THEN teamID = team~getInt("idTeam")

team~Close()
statement~Close()
conn~Close()

return teamID


/*************************************************************************************************
************/
 /*                                    DATABASE CONNECTION                                        */
/*************************************************************************************************
************/
::routine DBConn
use arg slotDir

/*@get("cb_homeTeam")*/
/*@get("cb_awayTeam")*/

/*@get("cb_Team_Export")*/
/*@get("cb_Player_Export")*/
/*@get("dp_date")*/
/*@get("cb_round")*/
/*@get("lb_Info_Export")*/

cb_awayTeam~getItems~Clear()
cb_homeTeam~getItems~Clear()
cb_Team_Export~getItems~Clear()
cb_Player_Export~getItems~Clear()
cb_round~getItems~Clear()

lb_Info_Export~Text = ""

cb_homeTeam~getItems~Add("Bitte Mannschaft ausw"'E4'x"hlen!")
cb_awayTeam~getItems~Add("Bitte Mannschaft ausw"'E4'x"hlen!")
cb_Team_Export~getItems~Add("Bitte Mannschaft ausw"'E4'x"hlen!")
cb_Player_Export~getItems~Add("Bitte Spieler ausw"'E4'x"hlen!")

/***MySQL Conntection***/
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
team = statement~executeQuery(sqlkomm)

check = .false
check = team~First
mannschaftenIndex = 0
DO WHILE check = .true
        IF (team~getString("TeamName") = "NO TEAM") THEN check = team~Next
        ELSE DO
                cb_homeTeam~getItems~Add(team~getString("TeamName"))
                cb_awayTeam~getItems~Add(team~getString("TeamName"))
                cb_Team_Export~getItems~Add(team~getString("TeamName"))
                mannschaftenIndex = mannschaftenIndex + 1
                check = team~Next
        END
END
```

```
sqlkomm = "SELECT * FROM Players ORDER BY Nachname ASC"
player = statement~executeQuery(sqlkomm)

check = player~First
DO WHILE check = .true
                Vorname = player~getString("Vorname")
                Nachname = player~getString("Nachname")
                Name = Vorname Nachname
                cb_Player_Export~getItems~Add(Name)
                check = player~Next
END

player~Close()
team~Close()
statement~Close()
conn~Close()
IF (mannschaftenIndex > 0) THEN DO
        DO i = 1 TO mannschaftenIndex -1
                cb_round~getItems~Add(i)
        END
        cb_round~getSelectionModel()~Select(0)
END
cb_homeTeam~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
cb_awayTeam~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
cb_Team_Export~getSelectionModel()~Select("Bitte Mannschaft ausw"'E4'x"hlen!")
cb_Player_Export~getSelectionModel()~Select("Bitte Spieler ausw"'E4'x"hlen!")


/*************************************************************************************************
************/
 /*                                    CHECK IF NUMBER BETWEEN 0-9                                  */
/*************************************************************************************************
************/
::routine CheckNum public
use arg Num
Check = .false

IF (Num >= 0 & Num <= 7) THEN Check = .true
ELSE DO
        Check = .false
END
return Check
/*++++++++++++++++++++++++++++++++++++++++*/
::routine CheckNum3Set public
use arg Num
Check = .false

IF (Num >= 0 & Num <= 20) THEN Check = .true
ELSE DO
        Check = .false
END
return Check
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1s1sh public
use arg slotDir
Check = .false

/*@get("tf_1s1sh")*/

Num = tf_1s1sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1s1sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1s1sa public
use arg slotDir
Check = .false

/*@get("tf_1s1sa")*/

Num = tf_1s1sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1s1sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
```

```
::routine KeyReleaseCheck_tf_1s2sh public
use arg slotDir
Check = .false

/*@get("tf_1s2sh")*/

Num = tf_1s2sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1s2sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1s2sa public
use arg slotDir
Check = .false

/*@get("tf_1s2sa")*/

Num = tf_1s2sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1s2sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1s3sh public
use arg slotDir
Check = .false

/*@get("tf_1s3sh")*/

Num = tf_1s3sh~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_1s3sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1s3sa public
use arg slotDir
Check = .false

/*@get("tf_1s3sa")*/

Num = tf_1s3sa~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_1s3sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2s1sh public
use arg slotDir
Check = .false

/*@get("tf_2s1sh")*/

Num = tf_2s1sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2s1sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2s1sa public
use arg slotDir
Check = .false

/*@get("tf_2s1sa")*/

Num = tf_2s1sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2s1sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2s2sh public
use arg slotDir
Check = .false

/*@get("tf_2s2sh")*/

Num = tf_2s2sh~Text
Check = CheckNum(Num)
```

```
IF Check = .false THEN tf_2s2sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2s2sa public
use arg slotDir
Check = .false

/*@get("tf_2s2sa")*/

Num = tf_2s2sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2s2sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2s3sh public
use arg slotDir
Check = .false

/*@get("tf_2s3sh")*/

Num = tf_2s3sh~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_2s3sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2s3sa public
use arg slotDir
Check = .false

/*@get("tf_2s3sa")*/

Num = tf_2s3sa~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_2s3sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_3s1sh public
use arg slotDir
Check = .false

/*@get("tf_3s1sh")*/

Num = tf_3s1sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_3s1sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_3s1sa public
use arg slotDir
Check = .false

/*@get("tf_3s1sa")*/

Num = tf_3s1sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_3s1sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_3s2sh public
use arg slotDir
Check = .false

/*@get("tf_3s2sh")*/

Num = tf_3s2sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_3s2sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_3s2sa public
use arg slotDir
Check = .false

/*@get("tf_3s2sa")*/

Num = tf_3s2sa~Text
```

```
Check = CheckNum(Num)

IF Check = .false THEN tf_3s2sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_3s3sh public
use arg slotDir
Check = .false

/*@get("tf_3s3sh")*/

Num = tf_3s3sh~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_3s3sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_3s3sa public
use arg slotDir
Check = .false

/*@get("tf_3s3sa")*/

Num = tf_3s3sa~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_3s3sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_4s1sh public
use arg slotDir
Check = .false

/*@get("tf_4s1sh")*/

Num = tf_4s1sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_4s1sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_4s1sa public
use arg slotDir
Check = .false

/*@get("tf_4s1sa")*/

Num = tf_4s1sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_4s1sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_4s2sh public
use arg slotDir
Check = .false

/*@get("tf_4s2sh")*/

Num = tf_4s2sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_4s2sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_4s2sa public
use arg slotDir
Check = .false

/*@get("tf_4s2sa")*/

Num = tf_4s2sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_4s2sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_4s3sh public
use arg slotDir
Check = .false

/*@get("tf_4s3sh")*/
```

```
Num = tf_4s3sh~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_4s3sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_4s3sa public
use arg slotDir
Check = .false

/*@get("tf_4s3sa")*/

Num = tf_4s3sa~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_4s3sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1d1sh public
use arg slotDir
Check = .false

/*@get("tf_1d1sh")*/

Num = tf_1d1sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1d1sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1d1sa public
use arg slotDir
Check = .false

/*@get("tf_1d1sa")*/

Num = tf_1d1sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1d1sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1d2sh public
use arg slotDir
Check = .false

/*@get("tf_1d2sh")*/

Num = tf_1d2sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1d2sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1d2sa public
use arg slotDir
Check = .false

/*@get("tf_1d2sa")*/

Num = tf_1d2sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_1d2sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1d3sh public
use arg slotDir
Check = .false

/*@get("tf_1d3sh")*/

Num = tf_1d3sh~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_1d3sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_1d3sa public
use arg slotDir
Check = .false
```

```
/*@get("tf_1d3sa")*/

Num = tf_1d3sa~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_1d3sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2d1sh public
use arg slotDir
Check = .false

/*@get("tf_2d1sh")*/

Num = tf_2d1sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2d1sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2d1sa public
use arg slotDir
Check = .false

/*@get("tf_2d1sa")*/

Num = tf_2d1sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2d1sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2d2sh public
use arg slotDir
Check = .false

/*@get("tf_2d2sh")*/

Num = tf_2d2sh~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2d2sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2d2sa public
use arg slotDir
Check = .false

/*@get("tf_2d2sa")*/

Num = tf_2d2sa~Text
Check = CheckNum(Num)

IF Check = .false THEN tf_2d2sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2d3sh public
use arg slotDir
Check = .false

/*@get("tf_2d3sh")*/

Num = tf_2d3sh~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_2d3sh~Text = ""
/*++++++++++++++++++++++++++++++++++++++++*/
::routine KeyReleaseCheck_tf_2d3sa public
use arg slotDir
Check = .false

/*@get("tf_2d3sa")*/

Num = tf_2d3sa~Text
Check = CheckNum3Set(Num)

IF Check = .false THEN tf_2d3sa~Text = ""
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*/
```

```
/*****************************************************************************************************
************/
 /*                                  Datebase Get PlayerID                                            */
/*****************************************************************************************************
************/
::routine DBPlayerID public
use arg Leader
parse var Leader Vorname Nachname

/* MySQL Verbindung erstellen */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM Players WHERE (Vorname='"Vorname"' AND Nachname='"Nachname"')"
spieler = statement~executeQuery(sqlkomm)

check = spieler~First
IF check = .true THEN spielerID = spieler~getString("idSpieler")

spieler~Close()
statement~Close()
conn~Close()

return spielerID


/*****************************************************************************************************
************/
 /*                                      FORMAT DATUM                                                 */
/*****************************************************************************************************
************/
::routine Datum public
use arg Datum

parse var Datum Jahr "-" Monat "-" Tag

Datum = Tag"."Monat"."Jahr

return Datum


/*****************************************************************************************************
************/
 /*                                     GET NAME OF PLAYER                                            */
/*****************************************************************************************************
************/
::routine GetPlayer public
 use arg ID
 /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM 4440_3.Players WHERE (idSpieler = "ID")"
        player = statement~executeQuery(sqlkomm)

        check = player~First
        DO WHILE check = .true
                Vorname = player~getString("Vorname")
                Nachname = player~getString("Nachname")
                Name = Vorname"/"Nachname
                check = player~Next
        END

        player~Close()
        statement~Close()
        conn~Close()
return Name
```

```
/*************************************************************************************************
************/
 /*                                        GET NAME OF TEAM                                       */
/*************************************************************************************************
************/
::routine GetTeam public
use arg ID
/* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")


        statement = conn~createStatement
        sqlkomm = "SELECT * FROM 4440_3.Team WHERE (idTeam = "ID")"
        team = statement~executeQuery(sqlkomm)

        check = team~First
        DO WHILE check = .true
                Name = team~getString("TeamName")
                check = team~Next
        END

        team~Close()
        statement~Close()
        conn~Close()
return Name

::requires BSF.CLS
::requires UNO.CLS
```

AlleMannschaften.rex

```
parse arg TeamID

oDesktop  = UNO.createDesktop()    -- get the UNO Desktop service object
xComponentLoader = oDesktop~XDesktop~XComponentLoader  -- get componentLoader
                                                       -- interface
/* open the blank *.sxc - file */
url = "private:factory/scalc"
xCalcComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)

/* get first sheet in spreadsheet */
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/*****************************************************************************/
/* Spalten Namen */
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"
CALL UNO.setCell xSheet, 0, 2, "Mannschaften"
CALL UNO.setCell xSheet, 1, 2, "Mannschaftsf"'FC'x"hrer"
CALL UNO.setCell xSheet, 2, 2, "Telefonnummer"
/*****************************************************************************/
        /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
        team = statement~executeQuery(sqlkomm)

        index = 3
        check = team~First
        DO WHILE check = .true
                IF (team~getString("TeamName") <> "NO TEAM") THEN DO
                        CALL UNO.setCell xSheet, 0 , index, team~getString("TeamName")

                        statement1 = conn~createStatement
                        sqlkomm    =    "SELECT    *    FROM    Players       WHERE    (idSpieler    =
"team~getString("TeamCaptain")")"
                        player = statement1~executeQuery(sqlkomm)
                        check_player = player~First
                        IF check_player = .true THEN DO
                                Vorname = player~getString("Vorname")
                                Nachname = player~getString("Nachname")
                                Leader = Vorname Nachname

        /*****************************************************************************/
                                CALL UNO.setCell xSheet, 1 , index, Leader
                                CALL        UNO.setCell        xSheet,        2        ,        index,
"'"player~getString("Telefonnummer")

        /*****************************************************************************/
                        END
                        index = index + 1
                END
                check = team~Next
        END

        player~Close()
        statement1~Close()
        team~Close()
        statement~Close()
        conn~Close()

/* Format Title & Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,2,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                          bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
 /* Changing Column Width */
xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns

/*Column 1 & 3*/
DO i = 0 to 3
```

```
  xCol = xCols~getByIndex(i) --fetch xCol
  props = xCol~XPropertySet --get access to its properties
  oldWidth = props~getPropertyValue("Width") --get current value
  newWidth = oldWidth + 1500
  props~setPropertyValue("Width",box("int",newWidth)) --set new Width
 END

 ::requires UNO.CLS
```

AlleSpieler.rex

```
parse arg TeamID

oDesktop  = UNO.createDesktop()    -- get the UNO Desktop service object
xComponentLoader = oDesktop~XDesktop~XComponentLoader  -- get componentLoader
                                                       -- interface
/* open the blank *.sxc - file */
url = "private:factory/scalc"
xCalcComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)

/* get first sheet in spreadsheet */
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/*****************************************************************************/
/* Spalten Namen */
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"
CALL UNO.setCell xSheet, 0, 2, "Vorname"
CALL UNO.setCell xSheet, 1, 2, "Nachname"
CALL UNO.setCell xSheet, 2, 2, "Telefonnummer"
CALL UNO.setCell xSheet, 3, 2, "Mannschaft"
/*****************************************************************************/
        /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Players ORDER BY Nachname ASC"
        player = statement~executeQuery(sqlkomm)

        index = 3
        check = player~First
        DO WHILE check = .true

        /*****************************************************************************/
                        CALL UNO.setCell xSheet, 0 , index, player~getString("Vorname")
                        CALL UNO.setCell xSheet, 1 , index, player~getString("Nachname")
                        CALL UNO.setCell xSheet, 2 , index, "'"player~getString("Telefonnummer")

        /*****************************************************************************/
                        statement1 = conn~createStatement
                        sqlkomm1 = "SELECT * FROM Team WHERE (idTeam = "player~getString("Team")")"
                        team = statement1~executeQuery(sqlkomm1)

                        check1 = team~First
                        IF   check1   =   .true   THEN   CALL   UNO.setCell   xSheet,   3   ,   index,
team~getString("TeamName")

                        check = player~Next
                        index += 1
        END


        team~Close()
        statement1~Close()
        player~Close()
        statement~Close()
        conn~Close()

/* Format Title & Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Mannschaft: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(3,2,3,index)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));


 /* Changing Column Width */
xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns
```

```
/*Column 1 & 3*/
DO i = 0 to 4
 xCol = xCols~getByIndex(i) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth + 1000
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
END

::requires UNO.CLS
```

Mannschaft.rex

```
parse arg TeamID

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM 4440_3.Singles WHERE (`ID-Team1` = "TeamID" OR `ID-Team2` = "TeamID") ORDER
BY Runde ASC, Datum ASC"
singles = statement~executeQuery(sqlkomm)

index = 3
count = 1
start = .true
title = .true

check = singles~First
IF(check = .true) THEN DO
DO WHILE check = .true
IF(start = .true) THEN DO
oDesktop  = UNO.createDesktop()    -- get the UNO Desktop service object
xComponentLoader = oDesktop~XDesktop~XComponentLoader  -- get componentLoader

/* open the blank *.sxc - file */
url = "private:factory/scalc"
xCalcComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)

/* get first sheet in spreadsheet */
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/*****************************************************************************/
/* Spalten Namen */
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"

/* Format Title: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,0,0)~XPropertySet
xPropSet~setPropertyValue("CharWeight",  box("float",  bsf.getConstant("com.sun.star.awt.FontWeight",
"BOLD")))
 /* Changing Column Width */
xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns

/*Column 1 & 8*/
DO i = 0 to 7
 xCol = xCols~getByIndex(i) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth + 1000
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
END
/*****************************************************************************/
start = .false
END
IF(count <= 4) THEN DO
IF(title = .true) THEN DO
        /*************************************************************************************/
        CALL UNO.setCell xSheet, 1, index, "Spieler"
        CALL UNO.setCell xSheet, 2, index, "Heim-Mannschaft"
        CALL UNO.setCell xSheet, 3, index, "Spieler"
        CALL UNO.setCell xSheet, 4, index, "Gast-Mannschaft"
        CALL UNO.setCell xSheet, 5, index, "1. Satz"
        CALL UNO.setCell xSheet, 6, index, "2. Satz"
        CALL UNO.setCell xSheet, 7, index, "3. Satz"
        /*************************************************************************************/
        /*************************************************************************************/
        CALL UNO.setCell xSheet, 0, index -1, "Runde:"
        CALL UNO.setCell xSheet, 1, index-1, singles~getString("Runde")
        CALL UNO.setCell xSheet, 2, index -1, "Datum:"
        CALL UNO.setCell xSheet, 3, index-1, "'"Datum(singles~getString("Datum"))
        /*************************************************************************************/
        /* Format Runde & Datum: Bold*/
```

```
        xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index -1,0,index - 1)~XPropertySet
        xPropSet~setPropertyValue("CharWeight",                                          box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
        xPropSet = xSheet~XCellRange~getCellRangeByPosition(2,index -1,2,index - 1)~XPropertySet
        xPropSet~setPropertyValue("CharWeight",                                          box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
        /* FormatTable Header: Bold*/
        xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index,7,index)~XPropertySet
        xPropSet~setPropertyValue("CharWeight",                                          box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

        /* Format Runde: Center*/
        xPropSet = xSheet~XCellRange~getCellRangeByPosition(1,index -1,1,index -1)~XPropertySet
        xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
        /* Format Datum: Center*/
        xPropSet = xSheet~XCellRange~getCellRangeByPosition(3,index -1,3,index -1)~XPropertySet
        xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
        index += 1
        title = .false
END
/*****************************************************************************/
        Spieler1 = GetPlayer(singles~getString("ID-Spieler1"))
        parse var Spieler1 Vorname1 "/" Nachname1
        Name1 =  Vorname1 Nachname1
        IF(Name1 = "NAME") THEN Name1 = "#NoPlayer"
        Spieler2 = GetPlayer(singles~getString("ID-Spieler2"))
        parse var Spieler2 Vorname2 "/" Nachname2
        Name2 =  Vorname2 Nachname2
        IF(Name2 = "NAME") THEN Name2 = "#NoPlayer"
        Team1 = GetTeam(singles~getString("ID-Team1"))
        Team2 = GetTeam(singles~getString("ID-Team2"))

        CALL UNO.setCell xSheet, 0, index, count".Einzel"
        CALL UNO.setCell xSheet, 1, index, Name1
        CALL UNO.setCell xSheet, 2, index, Team1
        CALL UNO.setCell xSheet, 3, index, Name2
        CALL UNO.setCell xSheet, 4, index, Team2
        Satz1 = "'"singles~getString("Satz1SP1") " : " singles~getString("Satz1SP2")
        CALL UNO.setCell xSheet, 5, index, Satz1
        Satz2 = "'"singles~getString("Satz2SP1") " : " singles~getString("Satz2SP2")
        CALL UNO.setCell xSheet, 6, index, Satz2
        IF(singles~getString("Satz3SP1") <> .Nil | singles~getString("Satz3SP2") <> .Nil) THEN DO
                Satz3 = "'"singles~getString("Satz3SP1") " : " singles~getString("Satz3SP2")
                CALL UNO.setCell xSheet, 7, index, Satz3
        END
/*****************************************************************************/
        index += 1
        count += 1
END
IF(count > 4) THEN DO
index += 7
count = 1
title = .true
END
check = singles~Next

/* Format Sätze: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(5,3,7,index)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
END

statement1 = conn~createStatement
sqlkomm1 = "SELECT * FROM 4440_3.Double WHERE (`ID-Team1` = "TeamID" OR `ID-Team2` = "TeamID") ORDER
BY Runde ASC, Datum ASC"
double = statement1~executeQuery(sqlkomm1)

count_doppel = 1
index = 9
check_double = double~First
DO WHILE (check_double = .true)
IF(count_doppel <= 2) THEN DO
/*****************************************************************************/
```

```
Spieler1 = GetPlayer(double~getString("ID-Spieler1"))
parse var Spieler1 Vorname1 "/" Nachname1
Name1 =  Vorname1 Nachname1
IF(Name1 = "NAME") THEN Name1 = "#NoPlayer"
Spieler2 = GetPlayer(double~getString("ID-Spieler2"))
parse var Spieler2 Vorname2 "/" Nachname2
Name2 =  Vorname2 Nachname2
IF(Name2 = "NAME") THEN Name2 = "#NoPlayer"

Spieler3 = GetPlayer(double~getString("ID-Spieler3"))
parse var Spieler3 Vorname3 "/" Nachname3
Name3 =  Vorname3 Nachname3
IF(Name3 = "NAME") THEN Name3 = "#NoPlayer"
Spieler4 = GetPlayer(double~getString("ID-Spieler4"))
parse var Spieler4 Vorname4 "/" Nachname4
Name4 =  Vorname4 Nachname4
IF(Name4 = "NAME") THEN Name4 = "#NoPlayer"

Team1 = GetTeam(double~getString("ID-Team1"))
Team2 = GetTeam(double~getString("ID-Team2"))

CALL UNO.setCell xSheet, 0, index, count_doppel".Doppel"
CALL UNO.setCell xSheet, 1, index, Name1
CALL UNO.setCell xSheet, 1, index +1, Name2
CALL UNO.setCell xSheet, 2, index, Team1
CALL UNO.setCell xSheet, 3, index, Name3
CALL UNO.setCell xSheet, 3, index +1, Name4
CALL UNO.setCell xSheet, 4, index, Team2
Satz1 = "'"double~getString("Satz1SP1") " : " double~getString("Satz1SP2")
CALL UNO.setCell xSheet, 5, index, Satz1
Satz2 = "'"double~getString("Satz2SP1") " : " double~getString("Satz2SP2")
CALL UNO.setCell xSheet, 6, index, Satz2
IF(double~getString("Satz3SP1") <> .Nil | double~getString("Satz3SP2") <> .Nil) THEN DO
        Satz3 = "'"double~getString("Satz3SP1") " : " double~getString("Satz3SP2")
        CALL UNO.setCell xSheet, 7, index, Satz3
END
/*****************************************************************************/
count_doppel += 1
index + = 2
END
IF(count_doppel > 2) THEN DO
index += 8
count_doppel = 1
END
check_double = double~Next
END
double~Close()
statement1~Close()
END
ELSE DO
RxMessageBox("Diese Mannschaft hat noch kein Spiel bestritten!", , "OK", "ERROR")
END

singles~Close()
statement~Close()
conn~Close()

ADDRESS CMD "EXIT"

::routine GetPlayer public
use arg ID
/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM 4440_3.Players WHERE (idSpieler = "ID")"
player = statement~executeQuery(sqlkomm)

check = player~First
DO WHILE check = .true
Vorname = player~getString("Vorname")
Nachname = player~getString("Nachname")
```

```
Name = Vorname"/"Nachname
check = player~Next
END

player~Close()
statement~Close()
conn~Close()
return Name

::routine GetTeam public
use arg ID
/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")


statement = conn~createStatement
sqlkomm = "SELECT * FROM 4440_3.Team WHERE (idTeam = "ID")"
team = statement~executeQuery(sqlkomm)

check = team~First
DO WHILE check = .true
Name = team~getString("TeamName")
check = team~Next
END

team~Close()
statement~Close()
conn~Close()
return Name

::routine Datum public
use arg Datum

parse var Datum Jahr "-" Monat "-" Tag

Datum = Tag"."Monat"."Jahr

return Datum


::requires BSF.CLS
::requires UNO.CLS
```

Spieler.rex

```
parse arg SpielerID

/* MySQL Verbindung erstellen  */
driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&password=Haas-
Tenniscenter")

statement = conn~createStatement
sqlkomm = "SELECT * FROM 4440_3.Singles WHERE (`ID-Spieler1` = "SpielerID" OR `ID-Spieler2` =
"SpielerID") ORDER BY Runde ASC, Datum ASC"
singles = statement~executeQuery(sqlkomm)

statement1 = conn~createStatement
sqlkomm1 = "SELECT * FROM 4440_3.Double WHERE (`ID-Spieler1` = "SpielerID" OR `ID-Spieler2` =
"SpielerID" OR `ID-Spieler3` = "SpielerID" OR `ID-Spieler4` = "SpielerID") ORDER BY Runde ASC, Datum
ASC"
double = statement1~executeQuery(sqlkomm1)

index = 3
start = .true

check = singles~First
check_double = double~First
IF(check = .true | check_double = .true) THEN DO
        IF(start = .true) THEN DO
                oDesktop  = UNO.createDesktop()    -- get the UNO Desktop service object
                xComponentLoader = oDesktop~XDesktop~XComponentLoader  -- get componentLoader

                /* open the blank *.sxc - file */
                url = "private:factory/scalc"
                xCalcComponent    =    xComponentLoader~loadComponentFromURL(url,    "_blank",    0,
.UNO~noProps)

                /* get first sheet in spreadsheet */
                xSheet                                                                          =
xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
                /*****************************************************************************/
                /* Spalten Namen */
                CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"

                /* Format Title: Bold*/
                xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,0,0)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                 box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
                 /* Changing Column Width */
                xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns

                /*Column 1 & 8*/
                DO i = 0 to 7
                 xCol = xCols~getByIndex(i) --fetch xCol
                 props = xCol~XPropertySet --get access to its properties
                 oldWidth = props~getPropertyValue("Width") --get current value
                 newWidth = oldWidth + 1000
                 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
                END
                /*****************************************************************************/
                start = .false
        END

        DO WHILE check = .true
                /*****************************************************************************/
                CALL UNO.setCell xSheet, 1, index, "Spieler"
                CALL UNO.setCell xSheet, 2, index, "Heim-Mannschaft"
                CALL UNO.setCell xSheet, 3, index, "Spieler"
                CALL UNO.setCell xSheet, 4, index, "Gast-Mannschaft"
                CALL UNO.setCell xSheet, 5, index, "1. Satz"
                CALL UNO.setCell xSheet, 6, index, "2. Satz"
                CALL UNO.setCell xSheet, 7, index, "3. Satz"
                /*****************************************************************************/
                /*****************************************************************************/
```

```
                CALL UNO.setCell xSheet, 0, index -1, "Runde:"
                CALL UNO.setCell xSheet, 1, index-1, singles~getString("Runde")
                CALL UNO.setCell xSheet, 2, index -1, "Datum:"
                CALL UNO.setCell xSheet, 3, index-1, "'"Datum(singles~getString("Datum"))
                /*****************************************************************************/
                /* Format Runde & Datum: Bold*/
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(0,index    -1,0,index    -
1)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                    box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(2,index    -1,2,index    -
1)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                    box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
                /* FormatTable Header: Bold*/
                xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index,7,index)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                    box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

                /* Format Runde: Center*/
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(1,index    -1,1,index    -
1)~XPropertySet
                xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
                /* Format Datum: Center*/
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(3,index    -1,3,index    -
1)~XPropertySet
                xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
                index += 1
                /*****************************************************************************/
                Spieler1 = GetPlayer(singles~getString("ID-Spieler1"))
                parse var Spieler1 Vorname1 "/" Nachname1
                Name1 =  Vorname1 Nachname1
                IF(Name1 = "NAME") THEN Name1 = "#NoPlayer"
                Spieler2 = GetPlayer(singles~getString("ID-Spieler2"))
                parse var Spieler2 Vorname2 "/" Nachname2
                Name2 =  Vorname2 Nachname2
                IF(Name2 = "NAME") THEN Name2 = "#NoPlayer"
                Team1 = GetTeam(singles~getString("ID-Team1"))
                Team2 = GetTeam(singles~getString("ID-Team2"))

                CALL UNO.setCell xSheet, 0, index, "Einzel"
                CALL UNO.setCell xSheet, 1, index, Name1
                CALL UNO.setCell xSheet, 2, index, Team1
                CALL UNO.setCell xSheet, 3, index, Name2
                CALL UNO.setCell xSheet, 4, index, Team2
                Satz1 = "'"singles~getString("Satz1SP1") " : " singles~getString("Satz1SP2")
                CALL UNO.setCell xSheet, 5, index, Satz1
                Satz2 = "'"singles~getString("Satz2SP1") " : " singles~getString("Satz2SP2")
                CALL UNO.setCell xSheet, 6, index, Satz2
                IF(singles~getString("Satz3SP1") <> .Nil | singles~getString("Satz3SP2") <> .Nil) THEN
DO
                        Satz3 = "'"singles~getString("Satz3SP1") " : " singles~getString("Satz3SP2")
                        CALL UNO.setCell xSheet, 7, index, Satz3
                END
                /*****************************************************************************/
                index += 3
                check = singles~Next
        END

        DO WHILE (check_double = .true)
                /*****************************************************************************/
                CALL UNO.setCell xSheet, 1, index, "Spieler"
                CALL UNO.setCell xSheet, 2, index, "Heim-Mannschaft"
                CALL UNO.setCell xSheet, 3, index, "Spieler"
                CALL UNO.setCell xSheet, 4, index, "Gast-Mannschaft"
                CALL UNO.setCell xSheet, 5, index, "1. Satz"
                CALL UNO.setCell xSheet, 6, index, "2. Satz"
                CALL UNO.setCell xSheet, 7, index, "3. Satz"
                /*****************************************************************************/
                /*****************************************************************************/
                CALL UNO.setCell xSheet, 0, index -1, "Runde:"
                CALL UNO.setCell xSheet, 1, index-1, double~getString("Runde")
                CALL UNO.setCell xSheet, 2, index -1, "Datum:"
```

```
                CALL UNO.setCell xSheet, 3, index-1, "'"Datum(double~getString("Datum"))
                /*****************************************************************************/
                /* Format Runde & Datum: Bold*/
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(0,index    -1,0,index    -
1)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                    box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(2,index    -1,2,index    -
1)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                    box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
                /* FormatTable Header: Bold*/
                xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index,7,index)~XPropertySet
                xPropSet~setPropertyValue("CharWeight",                                    box("float",
bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

                /* Format Runde: Center*/
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(1,index    -1,1,index    -
1)~XPropertySet
                xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
                /* Format Datum: Center*/
                xPropSet    =    xSheet~XCellRange~getCellRangeByPosition(3,index    -1,3,index    -
1)~XPropertySet
                xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
                index += 1
                /*****************************************************************************/
                Spieler1 = GetPlayer(double~getString("ID-Spieler1"))
                parse var Spieler1 Vorname1 "/" Nachname1
                Name1 =  Vorname1 Nachname1
                IF(Name1 = "NAME") THEN Name1 = "#NoPlayer"
                Spieler2 = GetPlayer(double~getString("ID-Spieler2"))
                parse var Spieler2 Vorname2 "/" Nachname2
                Name2 =  Vorname2 Nachname2
                IF(Name2 = "NAME") THEN Name2 = "#NoPlayer"

                Spieler3 = GetPlayer(double~getString("ID-Spieler3"))
                parse var Spieler3 Vorname3 "/" Nachname3
                Name3 =  Vorname3 Nachname3
                IF(Name3 = "NAME") THEN Name3 = "#NoPlayer"
                Spieler4 = GetPlayer(double~getString("ID-Spieler4"))
                parse var Spieler4 Vorname4 "/" Nachname4
                Name4 =  Vorname4 Nachname4
                IF(Name4 = "NAME") THEN Name4 = "#NoPlayer"

                Team1 = GetTeam(double~getString("ID-Team1"))
                Team2 = GetTeam(double~getString("ID-Team2"))

                CALL UNO.setCell xSheet, 0, index, "Doppel"
                CALL UNO.setCell xSheet, 1, index, Name1
                CALL UNO.setCell xSheet, 1, index +1, Name2
                CALL UNO.setCell xSheet, 2, index, Team1
                CALL UNO.setCell xSheet, 3, index, Name3
                CALL UNO.setCell xSheet, 3, index +1, Name4
                CALL UNO.setCell xSheet, 4, index, Team2
                Satz1 = "'"double~getString("Satz1SP1") " : " double~getString("Satz1SP2")
                CALL UNO.setCell xSheet, 5, index, Satz1
                Satz2 = "'"double~getString("Satz2SP1") " : " double~getString("Satz2SP2")
                CALL UNO.setCell xSheet, 6, index, Satz2
                IF(double~getString("Satz3SP1") <> .Nil | double~getString("Satz3SP2") <> .Nil) THEN
DO
                        Satz3 = "'"double~getString("Satz3SP1") " : " double~getString("Satz3SP2")
                        CALL UNO.setCell xSheet, 7, index, Satz3
                END
                /*****************************************************************************/
                index + = 4

                check_double = double~Next
        END
        /* Format Sätze: Center*/
        xPropSet = xSheet~XCellRange~getCellRangeByPosition(5,3,7,index)~XPropertySet
        xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));
END
```

```
ELSE DO
        RxMessageBox("Dieser Spieler hat noch kein Spiel bestritten!", , "OK", "ERROR")
END
double~Close()
statement1~Close()
singles~Close()
statement~Close()
conn~Close()

ADDRESS CMD "EXIT"

::routine GetPlayer public
 use arg ID
 /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM 4440_3.Players WHERE (idSpieler = "ID")"
        player = statement~executeQuery(sqlkomm)

        check = player~First
        DO WHILE check = .true
                Vorname = player~getString("Vorname")
                Nachname = player~getString("Nachname")
                Name = Vorname"/"Nachname
                check = player~Next
        END

        player~Close()
        statement~Close()
        conn~Close()
return Name

::routine GetTeam public
use arg ID
/* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")


        statement = conn~createStatement
        sqlkomm = "SELECT * FROM 4440_3.Team WHERE (idTeam = "ID")"
        team = statement~executeQuery(sqlkomm)

        check = team~First
        DO WHILE check = .true
                Name = team~getString("TeamName")
                check = team~Next
        END

        team~Close()
        statement~Close()
        conn~Close()
return Name

::routine Datum public
use arg Datum

parse var Datum Jahr "-" Monat "-" Tag

Datum = Tag"."Monat"."Jahr

return Datum

::requires BSF.CLS
::requires UNO.CLS
```

Results.rex

```
oDesktop  = UNO.createDesktop()    -- get the UNO Desktop service object
xComponentLoader = oDesktop~XDesktop~XComponentLoader  -- get componentLoader
                                                       -- interface
/* open the blank *.sxc - file */
url = "private:factory/scalc"
xCalcComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)

/* get first sheet in spreadsheet */
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/*****************************************************************************/
/* Spalten Namen */
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"
CALL UNO.setCell xSheet, 1, 2, "Mannschaften"
CALL UNO.setCell xSheet, 2, 2, "Mannschaftf"'FC'x"hrer"
CALL UNO.setCell xSheet, 3, 2, "Telefonnummer"
/*****************************************************************************/
        /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
        team = statement~executeQuery(sqlkomm)

        index = 3
        check = team~First
        DO WHILE check = .true
                IF (team~getString("TeamName") <> "NO TEAM") THEN DO
                        CALL UNO.setCell xSheet, 1 , index, team~getString("TeamName")

                        statement1 = conn~createStatement
                        sqlkomm    =    "SELECT    *    FROM    Players       WHERE    (idSpieler    =
"team~getString("TeamCaptain")")"
                        player = statement1~executeQuery(sqlkomm)
                        check_player = player~First
                        IF check_player = .true THEN DO
                                Vorname = player~getString("Vorname")
                                Nachname = player~getString("Nachname")
                                Leader = Vorname Nachname

        /*****************************************************************************/
                                CALL UNO.setCell xSheet, 2 , index, Leader
                                CALL        UNO.setCell        xSheet,        3        ,        index,
"'"player~getString("Telefonnummer")

        /*****************************************************************************/
                        END
                        index = index + 1
                END
                check = team~Next
        END

        player~Close()
        statement1~Close()
        team~Close()
        statement~Close()
        conn~Close()
/*****************************************************************************/
indexJ = index + 3
/*****************************************************************************/
CALL UNO.setCell xSheet, 0, indexJ, "Rang"
CALL UNO.setCell xSheet, 1, indexJ, "Mannschaften"
CALL UNO.setCell xSheet, 2, indexJ, "Punkte"
CALL UNO.setCell xSheet, 3, indexJ, "Games"
CALL UNO.setCell xSheet, 4, indexJ, "Sets"
/*****************************************************************************/
indexJ = indexJ + 1

        /* MySQL Verbindung erstellen  */
```

```
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Team ORDER BY Punkte DESC, Games DESC, Sets DESC"
        team = statement~executeQuery(sqlkomm)

        Rang = 1
        check = team~First
        DO WHILE check = .true
                IF (team~getString("TeamName") <> "NO TEAM") THEN DO

        /*****************************************************************************/
                        CALL UNO.setCell xSheet, 0, indexJ, Rang
                        CALL UNO.setCell xSheet, 1, indexJ, team~getString("TeamName")
                        CALL UNO.setCell xSheet, 2, indexJ, team~getString("Punkte")
                        CALL UNO.setCell xSheet, 3, indexJ, team~getString("Games")
                        CALL UNO.setCell xSheet, 4, indexJ, team~getString("Sets")

        /*****************************************************************************/

                        indexJ = indexJ + 1
                        Rang = Rang + 1
                END
                check = team~Next
        END

        team~Close()
        statement~Close()
        conn~Close()

/* Format Title & Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Telefonnummer: Right*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index + 3,4,index + 3)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
/* Format Table: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index + 3,0,indexJ)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));

/* Format Table: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(2,index + 3,4,indexJ)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));

/* Changing Column Width */
xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns

/*Column 2 & 4*/
DO i = 1 to 4
 xCol = xCols~getByIndex(i) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth + 1000
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
END

/*Column 1*/
 xCol = xCols~getByIndex(1) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth - 200
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
```

```
::requires UNO.CLS
```

Spielbericht_GUI.fxml

```
oDesktop  = UNO.createDesktop()    -- get the UNO Desktop service object
xComponentLoader = oDesktop~XDesktop~XComponentLoader  -- get componentLoader
                                                       -- interface
/* open the blank *.sxc - file */
url = "private:factory/scalc"
xCalcComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)

/* get first sheet in spreadsheet */
xSheet = xCalcComponent~XSpreadSheetDocument~getSheets~XIndexAccess~getByIndex(0)~XSpreadSheet
/*****************************************************************************/
/* Spalten Namen */
CALL UNO.setCell xSheet, 0, 0, "WinterCup im Haas-Tenniscenter"
CALL UNO.setCell xSheet, 1, 2, "Mannschaften"
CALL UNO.setCell xSheet, 2, 2, "Mannschaftf"'FC'x"hrer"
CALL UNO.setCell xSheet, 3, 2, "Telefonnummer"
/*****************************************************************************/
        /* MySQL Verbindung erstellen  */
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Team ORDER BY TeamName ASC"
        team = statement~executeQuery(sqlkomm)

        index = 3
        check = team~First
        DO WHILE check = .true
                IF (team~getString("TeamName") <> "NO TEAM") THEN DO
                        CALL UNO.setCell xSheet, 1 , index, team~getString("TeamName")

                        statement1 = conn~createStatement
                        sqlkomm    =    "SELECT    *    FROM    Players       WHERE    (idSpieler    =
"team~getString("TeamCaptain")")"
                        player = statement1~executeQuery(sqlkomm)
                        check_player = player~First
                        IF check_player = .true THEN DO
                                Vorname = player~getString("Vorname")
                                Nachname = player~getString("Nachname")
                                Leader = Vorname Nachname

        /*****************************************************************************/
                                CALL UNO.setCell xSheet, 2 , index, Leader
                                CALL        UNO.setCell        xSheet,        3        ,        index,
"'"player~getString("Telefonnummer")

        /*****************************************************************************/
                        END
                        index = index + 1
                END
                check = team~Next
        END

        player~Close()
        statement1~Close()
        team~Close()
        statement~Close()
        conn~Close()
/*****************************************************************************/
indexJ = index + 3
/*****************************************************************************/
CALL UNO.setCell xSheet, 0, indexJ, "Rang"
CALL UNO.setCell xSheet, 1, indexJ, "Mannschaften"
CALL UNO.setCell xSheet, 2, indexJ, "Punkte"
CALL UNO.setCell xSheet, 3, indexJ, "Games"
CALL UNO.setCell xSheet, 4, indexJ, "Sets"
/*****************************************************************************/
indexJ = indexJ + 1

        /* MySQL Verbindung erstellen  */
```

```
        driverMgr = bsf.loadClass("java.sql.DriverManager") -- load Java class
        bsf.loadClass('com.mysql.jdbc.Driver')~newinstance
        conn=driverMgr~getConnection("jdbc:mysql://db02.easyserver.at:3306/4440_3?user=4440_3&passwo
rd=Haas-Tenniscenter")

        statement = conn~createStatement
        sqlkomm = "SELECT * FROM Team ORDER BY Punkte DESC, Games DESC, Sets DESC"
        team = statement~executeQuery(sqlkomm)

        Rang = 1
        check = team~First
        DO WHILE check = .true
                IF (team~getString("TeamName") <> "NO TEAM") THEN DO

        /*****************************************************************************/
                        CALL UNO.setCell xSheet, 0, indexJ, Rang
                        CALL UNO.setCell xSheet, 1, indexJ, team~getString("TeamName")
                        CALL UNO.setCell xSheet, 2, indexJ, team~getString("Punkte")
                        CALL UNO.setCell xSheet, 3, indexJ, team~getString("Games")
                        CALL UNO.setCell xSheet, 4, indexJ, team~getString("Sets")

        /*****************************************************************************/

                        indexJ = indexJ + 1
                        Rang = Rang + 1
                END
                check = team~Next
        END

        team~Close()
        statement~Close()
        conn~Close()

/* Format Title & Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Telefonnummer: Right*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,0,3,2)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))

/* Format Table Header: Bold*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index + 3,4,index + 3)~XPropertySet
xPropSet~setPropertyValue("CharWeight", box("float", -
                        bsf.getConstant("com.sun.star.awt.FontWeight", "BOLD")))
/* Format Table: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(0,index + 3,0,indexJ)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));

/* Format Table: Center*/
xPropSet = xSheet~XCellRange~getCellRangeByPosition(2,index + 3,4,indexJ)~XPropertySet
xPropSet~XPropertySet~setPropertyValue("HoriJustify",
bsf.getConstant("com.sun.star.table.CellHoriJustify", "CENTER"));

/* Changing Column Width */
xCols=xSheet~XColumnRowRange~getColumns --get XTableColumns

/*Column 2 & 4*/
DO i = 1 to 4
 xCol = xCols~getByIndex(i) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth + 1000
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
END

/*Column 1*/
 xCol = xCols~getByIndex(1) --fetch xCol
 props = xCol~XPropertySet --get access to its properties
 oldWidth = props~getPropertyValue("Width") --get current value
 newWidth = oldWidth - 200
 props~setPropertyValue("Width",box("int",newWidth)) --set new Width
```

```
::requires UNO.CLS
```