Criterion C – Development

Introduction

This program is a database, made using Netbeans, using Java language. The database is a GUI, where the user inputs the scores during the game, and the program works with numbers. It sums, compares, displays and sorts them.

Summary List of All Techniques

- Private and public classes, attributes
- Nested loops
- Arrays
- Arraylists
- Method returning a value
- User defined objects made from an OOP "template" class
- Encapsulation of private methods that work on public attribute of a "template" class
- Making an array of objects
- Simple and compound selection (if/else)
- Sorting (bubble sort etc.), and in particular sorting an array of objects based on one key attribute
- Searching (linear search, binary search…)
- Parsing strings to ints
- GUI tabs
- GUI tables
- Use of a flag value (such as -999, or "not set yet")
- Overloaded constructors, which work differently depending on the parameters sent

Structure of the Program

This program is divided into different classes, because then the working with it is not so complex. The program is divided into 5 classes. The main class is called MainDossierGUI. It is a GUI class, consisting of 7 tabs. The four tabs are the same, as they display the scores for 4 teams. In these 4 tabs the scores are ready for user to be inputted and then calculated by program. The 5th tab is where the overall team scores are displayed, gained from the first 4 gaps. Also, in the 5th tab, it is stated whether did one pair beat the other, or that they are in withdraw.
The 6th tab is the table. It collects all the inputted information and processes it.
And the 7th tab is a way to improve a program. In another class I have the smartest bubble search for sorting the teams in the table. The last 3 tabs are used for giving attributes to the teams and their scores.

Data Structures Used

I have an array of teams. The array allows me to work with them together, as one group, not individually. I have not used any another function, for example, an array list, because I only have 4 teams, and the amount does not extend. There are four teams, prepared for the user. Each object in the array has 4 attributes. These represent the main information about the team: team name, overall points, number of games and full points, which are calculated after winning.

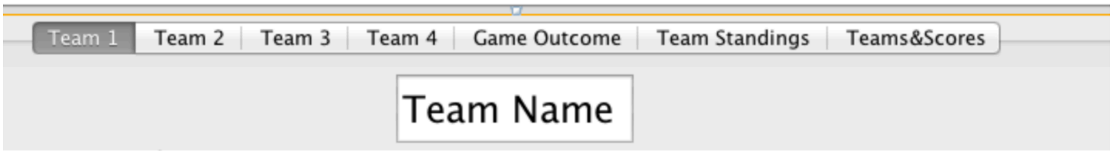
Also, I have used a file structure in order to save the scores of the teams. This is needed to work with the numbers, for example to sum and sort them.

Main Unique Algorithms

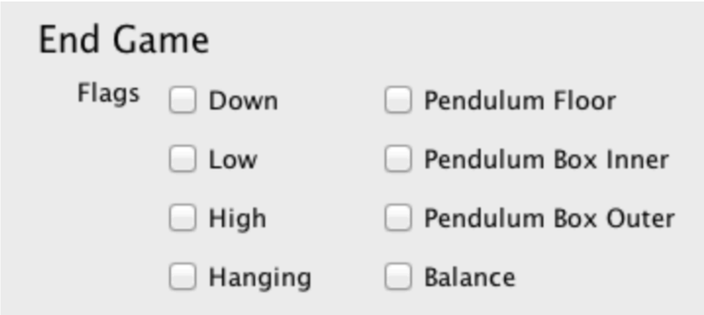
At the beginning, the user should input the scores. They can be inputted in any order, team 2 can be filled firstly, or half of team 4, there is no special order. However when a set of scores is input, the button "submit" should be clicked. It sums up the scores in each of the sections, and finally, it calculates the overall score for a team. After all of the scores are input, they are displayed in the 5th tab, after clicking "show results". The next thing to do is to click "plus extra points for winning/draw". The program divides 4 teams into 2 pairs and sums their scores. The pair, which has more points, wins 2 points per team. If the teams are in a draw, one point is being added to all of them. After this tab, the user should go to the table tab (Team Standings) and click the "Display" button. Then, the program sorts the teams, using smartest bubble sort, according to their scores, giving such details as team name, place, number of games played, overall score and final score.

User Interface/GUI Work

In my GUI I have used several Java Swing components. Java Swing is a easy-manageable Java GUI widget toolkit. For example, I have used different tabs. Each of them has their own function. The first four (Team 1,2,3,4) are used for working with team scores, particularly for inputting and summing them. The Game outcome tab shows the final scores for the teams. Team standing tab stands for viewing the most important information about the teams, and the last tab are left for developing. As you can see, the tabs are very helpful in organizing the functions, which your program should perform.



Another widget I have used is a Check box. I needed to assign certain values for individual functions. The scores differ, depending on which box is ticked. For example, "down" is 1 point, while "low" is 20.



End Game

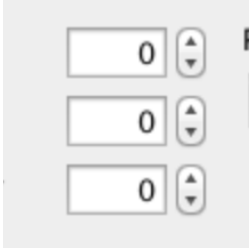
Flags □ Down □ Pendulum Floor

□ Low □ Pendulum Box Inner

□ High □ Pendulum Box Outer

□ Hanging □ Balance

One more widget I have used is spinner. It gives an opportunity of choosing one value out of several. For example, the amount of boxes lifted up by a robot.

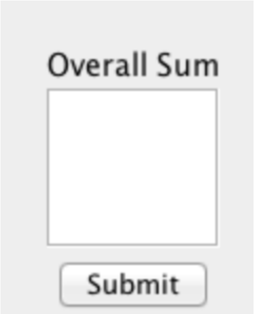


0

0

0

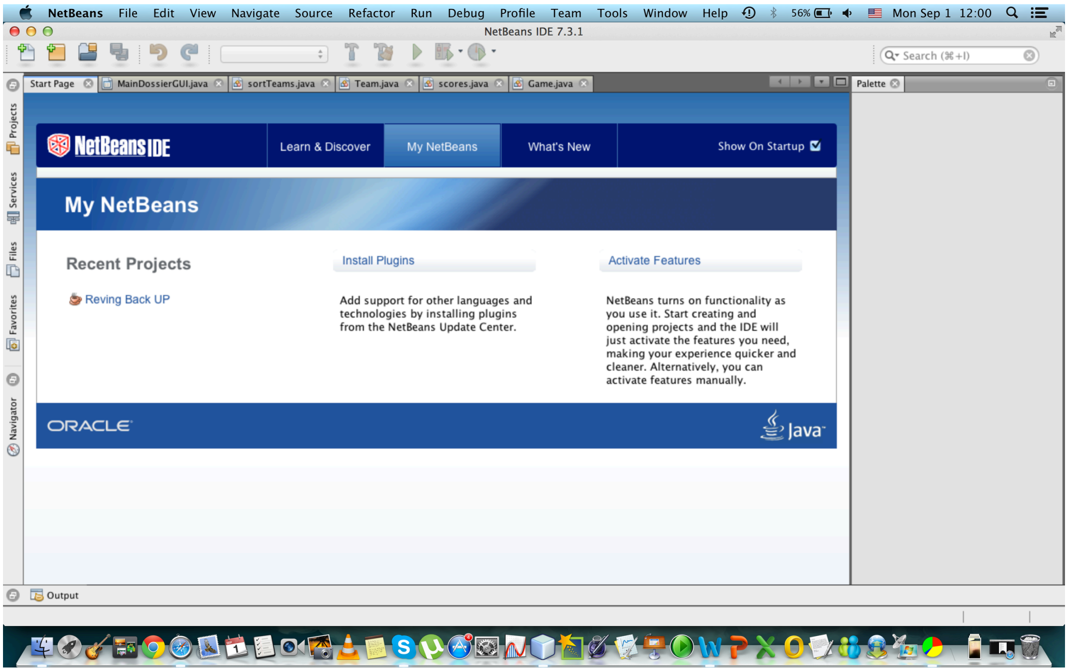
And another widget that I have used is Button. You can assign any function you want for a button. In my case one example is a submit (sending to other parts of the program) function.



Software Tools Used

The main software tool that I have used is Netbeans. Netbeans is based on Java, and allows a wide spectrum of opportunities. This software tool lets the user to create his own programs, for example databases. Netbeans is a user-friendly program, as it is easy to navigate it, to learn how it works and how to program on it.

Ivan Sirotkin



My NetBeans

Recent Projects

Reving Back UP

Install Plugins

Add support for other languages and technologies by installing plugins from the NetBeans Update Center.

Activate Features

NetBeans turns on functionality as you use it. Start creating and opening projects and the IDE will just activate the features you need, making your experience quicker and cleaner. Alternatively, you can activate features manually.

Another software tool that I have used for my work is Omnigraffle, which allows working with graphs of many different kinds. I needed it to show how my program works using algorithms, step-by-step.