

Marist IHSA Point Calculator

Final Write Up

Abstract

Calculating and keeping track of individual rider's points has become a tedious task for the Marist Equestrian team. Not only is determining how many points a rider has won from each competition difficult, but organizing all the data that has accumulated has become an almost impossible task. With the use of this software and database, a user can easily input, access, and manipulate rider's data that accumulates from horse showing. The software has been created through TextWrangler in java and connects to a mySQL database using JDBC and my computer's terminal.

Introduction

The goal of this project is to tackle the Marist Equestrian Team's problem of calculating individual rider's points. Being on the team myself, I have seen that there is not only a lack of organization in the recording of points, but also a lack of knowledge of rider's individual points. With the uses of this database, the Marist Equestrian team will be able to easily access information on rider's point and competition class, as well as able to easily manage and edit information. This will not only help with the organization of the team, but also with strategically planning who will ride during each show.

Following the prompted question makes the program user friendly to all individuals.

This database project uses the language mySQL to access and manipulate tables. It also uses the JDBC api that allows easy communication between mySQL and

my java code. With the use of the data collected from the user's answers to the question, various methods are called that invoke the JDBC and MySQL databases.

Looking at the structure and flow of the coded software, as well as other technology already available for the public, one can see the need for this simplistic yet extremely helpful tool. Not only is this product useful for use today, but it also is user-friendly.

Detailed System

The program is set up so the user is first prompted to type in one of five numbers, all corresponding to five different actions. The answer is then transported to a switch statement, which leads the user to more questions that will then be used to enact various methods. This process will then be repeated unless the user enacts option five, which stops the continuous loop of question and exits the program.

The first option the user can choose is if he or she wants to add a rider to the database. This case then asked the user for the rider's first name, last name, competition class, class placement, and incoming points. The class placement information gathered is then passed through the points method, which calculates the total points the rider won. In the IHSA world, different placements and ribbons correspond to different points. For example, if a rider won first place, he or she will receive 7 points. However, if the user gets a second place ribbon, he or she will get 5 points. This convoluted point system makes first place ribbons worth more but makes it confusing for riders to calculate their own points. The point method simply uses a switch method that takes the placement and matches it to its corresponding points. The points

are then return to the first case in the main method. Then, points are added to the incoming points, creating a variable that holds the total points. All of the information gathered is then passed through the NewRider method, which connects to the mySQL MaristIHSA databases and adds the rider.

The next choice the user has is to update the rider's points. This option is enacted when the user types in the number two, which leads to case two in the switch statement. There, it collects the rider's last name, competition class, and class placement. The class placement information is then passed through the point calculator method. Once the points have been calculated, the total points calculator method is enacted, and all the information collected is passed through. This method connects to the database and selects a points column using the rider's last name and competition class. This information is then used to add to the points earned from the competition, and then the method returns the total points. The total points, as well as last name and competition class, are passed through the update method. This method updates the rider's total points in the database, using his or her last name and competition class. Lastly, case two prints the points the rider won.

The third ability that the user is able to do is delete a rider from a database. This case collects the last name and competition class of the rider that the user wants to delete. These two pieces of information are used to invoke the "delete rider" method. This method connects to the date bases and deletes the row that has the corresponding rider's last name and competition class. This is useful when the user makes a mistake entering in information or when a rider graduates.

The fourth choice a user has is to view the database. This switch case asked the user for the last name of the rider. This is then used to invoke the “view data” method, which prints the data stored in the database under the rider’s name. Since some riders can be entered in more than one class, their names are listed twice in the database but have different competition class information. This method prints all the data for the rider whose last name is written.

The very last case the rider can choose is if he or she wants to exit the program. If the user enters five, the program enacts a return, which makes the while statement, which encapsulates the whole switch statement , become untrue, stopping the questions. However, an input of 1-4 in the first question will invoke the proper case statement but then return back to the original question, letting the user keep manipulating the database. This is useful so the user does not keep having to reload and rerun the program each time.

IhsaCalc
pathNum: int fName: String lName: String cClass: String place: int yesOrNo: String yesData: int points: int newRiderP: int tPoints: int
pointsCalc(int num) : int totalPoints(int earned, String last, String c) : int

```
Update(int total, String last, String c) : void  
NewRider(String first, String last, String c, int p) : void  
DeleteRider( String l, String c) : void  
ViewData() : void
```

Requirements

A requirement for the project includes the text editor, TextWrangler, which was necessary in the creation of the java code. The next requirement was a terminal and a computer. This will help in the compilation of the code, as well as the running of the code. Another requirement is the mySQL download, as well as the JDBC download. Lastly this program requires a user that inputs the proper information.

Literature Survey

One software already being utilized in the equestrian world is StartBox. This database software makes shows for the equestrian sport of Eventing run smoothly and lets competitors easily access riding times and scores. It makes the workload for host venues drastically decrease. The producers of this product are Scott Weber, Ron Smith, and Jennifer Weber, who share a love for computers as well as the equestrian community.

The equestrian sport for this software is catered to Eventing. This discipline requires the horse and rider to complete the three different events of Dressage: cross country jumping and stadium jumping. Once a rider decides to enter in an event, he or she chooses a level of competition, which automatically enters them into each of the three events. The host venue is then faced with the almost impossible task of scheduling time slots for sometimes hundreds of riders, all in different levels of

competitions. StartBox helps the host venue sort times out easily, making this once frustrating task a breeze. Startbox also stores all rider's information, making the secretarial part of the showing process easy as well. Prior to this system, the host location would have to create its own organizational database using a filing system or a spreadsheet, which always takes up so much time that could be spent more efficiently. This software reduces the drudgery by tapping into online databases and letting multiple people at a time edit the entries. All of the entries are connected to start times, as well as the scores the users accumulate from the shows. After a rider has completed, StartBox immediately updates the scores and posts them on the web. The leaderboard changes the listing, making it dynamic for the user to view his or her scores or other scores. Lastly, once the show is over, StartBox reports all the scores to big Eventing communities such as USEA, USEF, and FEI with just one click of a button. This software reduces the amount of work host venues spend on organization of rider entries by drastic amounts, letting them worry about more important things, such as horses.

Even though the style of riding is different than the Marist Equestrian team's style, the need for a software like this is still there. My own software, although not as complex, allows team members and trainers calculate rider's points from scores, letting them statistically plan how to improve individually as well as a team just as StartBox does but on a much smaller scale. StartBox is a successful software that proves that equestrian people have a need for an online organization based software.

User Manual

This program is accessed through the terminal with the proper mySQL database, JDBC files, and program file, all located in the same folder. First, the database needs to be opened in a terminal window with the ability to edit it. This is done with the simple input of use and then the databases name. On my computer, the database is named "javabook." The second step is the compiled file need to be run on a different window of the terminal using the input of "java -cp .:mysql-connector-java-5.1.41-bin.jar IhsaCalc". This calls the proper JDBC file and the proper compiled file of the software. This then brings up the first initial question of what the user wants to do to the database explained in the detailed system.

The user should be sure to input strings where it is apparent strings should be used and integers where it is obvious that integers should be used. Although capitals do not matter in mySQL, spelling does. The user should also note that if a number through 1-5 is not entered in the initial question, the program will give the user a default prompt of error: invalid statement.

Conclusion

The goal of this project is to fix the ongoing struggle of calculating and storing the Equestrian team's points from competitions. This user-friendly software makes it easy to input, edit, delete, and view individual rider's points that one accumulates from competing. Using textwrangler in java, mySQL, JDBC, and the terminal on my computer, this database is able to come to life to tackle the organizational problem the equestrian team faces. As seen with StartBox, an online database software is not only helpful, but in high demand. With the use of the Marist IHSA databases point calculator

software, the Marist equestrian team can conquer its organizational problem and focus instead on growing as riders.

Bibliography:

Laing, Daniel. *Introduction to Java Programming*. Pearson, 2015

"MySQL Community Downloads." *MySQL*. N.p., n.d. Web. 27 Apr. 2017.

Solutions, Jennifer Weber Solius Web. "StartBox Horse Show Management Software." *StartBox Scoring - Online Scoring Software for Horse Show Events*. N.p., 2008. Web. 29 Apr. 2017.