Learning Team: Programming Improvement 1

Jason Garrick, Jonathon McNeil, Reuben Ellis

PRG/421

July 25, 2016

Edward Spear

Unique Lane Assignments for Racers Code Review

Whenever a new program begins, there is also a necessity to develop that program. Development is necessary to help software companies create programs that meet the needs of the users. As the computer world continues to grow by leaps and bounds, the need for software development also increases. From the business aspect, companies who use computers for daily work use, want to have their customized program that meets business needs while remaining cost efficient. By breaking the development process into phases, the challenges of creating new software are not only manageable for developer teams, but they allow the addressing of each issue before moving forward. Another reason for having a development plan is to create a current timetable. By having a structured process that needs following, teams move forward with the development process in a much faster time than before such standards were in place. A software company's name and reputation are also affected if a new product lacks development standards. Glitches in a program can lead to problems with a company using the product. For example, if a program were created to handle financial transactions and the program has errors, the company using the program could lose any amount of their money or mishandle customer money.

In upcoming weeks in project development, Team B will need to divide the project sooner in the week to ensure all aspects of the review are complete each week. Communication is also critical to ensure all team members are on track, so the team projects are complete.

We as a team decided to perform a code review of Jonathon’s program from week 1. When we extracted the java file from the Zip folder provided, we immediately noticed two public classes within the same file. Knowing we could not compile the project, we decided to split the classes into two separate class files named Race and Track. The reason we split the classes into two files is a Java program can only have one public class per file, and it is against object-oriented programming principles to make a class private. We then ran the project, and although each runner was unique, we were able to assign one lane to all eight runners. We made some changes to the Race class to match the Singleton design.

The first change we made was to add a final instance of the Race class so only one instance of the class is usable and the program cannot use two separate instances to assign the same lane. With only one instance, the user has to designate one lane at a time to ensure a unique lane assignment. We then added a single string set to hold the eight lanes. The constructor must be private, so only one instance occurs within the program. Within the private constructor, we made the Set of strings a HashSet, which is an unordered list which operates with time complexity 0(1) meaning constant time. The significance of having a constant time complexity is the HashSet won't be affected by the size or values within the set. A HashSet is much faster than a regular list and is useful when using the methods add, remove, or contain a list. A HashSet does not have sort capabilities, and there are no index values for the elements within the HashSet, so it is only for simple sets with one or two actions. We then added the lanes to the Set naming the lanes Lane1-Lane8 using the add method. The class already contained a method to return the instance so the program can access the instance within the main class. We also designated a Boolean method which removes a lane after the user assigns a value and the method accepts a Boolean of true. The last method was a Set of Strings which return the Lane numbers so the program can check to see if the lane is still available.

Within the Track class is the Main method which runs the entire program. We added an instance of the Race class through the public getInstance method within the Race class which returns an instance. The Main method only does two things, calls the pickTrack method which assigns a lane to each runner and displays the runners and their corresponding lanes once the lane allocation is complete. The pickTrack method created a new Scanner instance and a new racer value for each input for each track. We shortened the process by adding a for loop which starts at one since there are only eight runners and eight lanes. The for loop iterates through the runners with each runner identified by the variable "i" which assigns a 1-8 based on the for loop iteration. This approach allows for only one System.out.Println rather than eight separate statements. We added a do-while loop which continues the current iteration lane assignment if the user enters a number less than one, greater than eight, or if the lane is already in use. This approach forces the runner to receive track before the next runner can be assigned a lane. We also added a check which checks to see if the lane exists in the list by calling the list method from the Race class and if the lane exists then it can allocate to a racer and if it doesn't, the user will receive a message stating the lane is not available. If the lane does exist, another check is in place to ensure the lane is between one and eight and passes the lane name to the method within the Race class which removes the lane from the list. The lane also adds to an ArrayList, which is used in the Main method to display the value of the lane for each racer. We added a try-catch block within the pickTrack method which lets the user know if an exception occurs it is because a letter or symbol was entered instead of a number. The last step in the program is the Main method. We added a for loop which iterates through one System.out.println statement which displays the Runner number and the track which belongs to the runner. The for loop allows for only one statement instead of eight different statements, one for each runner.

References

C# - Define: What is a HashSet?. (2010). Retrieved from <http://stackoverflow.com/questions/4558754/define-what-is-a-hashset>

ComputerMoz.com. (2016). Retrieved from <http://www.computermoz.com/blog/why-is> software-development-important/

Sierra, K. & Bates, B. (2015). OCA/OCP Java SE 7 Programmer I & II Study Guide (Exams 1Z0 803 & 1Z0-804. New York: McGraw-Hill Education.

Streetdirectory.com. (2016). Retrieved from <http://www.streetdirectory.com/travel_guide/135488/software/software_development>

[\_process\_and\_its\_importance.html](http://www.streetdirectory.com/travel_guide/135488/software/software_development)