Programming Project 1 Documentation

Jade Pearl

CMSC 215 Intermediate Programming

Professor Romerl Elizes

University of Maryland Global Campus

UML Class Diagrams

|  |
| --- |
| **Height** |
| -feet : int  -inches : int |
| +Height(int feet, int inches)  +toInches() : int  +toString() : String |

|  |
| --- |
| **Player** |
| -name : String  -height : Height  -age : int |
| +Player(String name, Height height, int age)  +getName() : String  +getHeight() : Height  +getAge() : Int  +toString() : String |



|  |
| --- |
| **Project1** |
|  |
| + main(String[] args) : void |

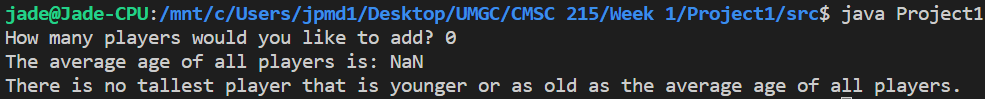
Approach Discussion

My approach to this project went step-by-step as follows:

1. First creating the UML diagrams of each class so that I can visually map out how I am going to build each class
2. Figuring out how to build the non-getter methods such as toInches or toString in order for the program to properly function
3. Planning out what I will be doing in the main method under Project1. I made sure that I first prompted the user for their inputs first before converting anything and adding to the ArrayList that was implemented
4. Compiling each class code to make sure there were no syntax or compilation errors.
5. Tested the Project1 program multiple times and input different scenarios as shown in the Test cases. Once everything was tested, I went back and edited my code to fix any issues. I had to double check the output to make sure all calculations were correctly made.

Test Plan

I presented my test cases in the form of sample outputs. The test cases are listed below:

* Test 1: Output for when the user chooses to enter zero players
  + 
* Test 2: Testing if 1 player input works
  + A screen shot of a computer

    Description automatically generated
* Test 3: Tests whether a double works when input anywhere in the height. (The variables were defined as integers, so a double should not work due to integer conversion being an issue). This result was expected!
  + A screen shot of a computer

    Description automatically generated
* Test 4: Testing more than 2 players.
  + A screenshot of a computer

    Description automatically generated
* Test 5: Tests if the toString in the Height class properly normalizes the height entered by the user (converting abnormal inches into feet and adding it to the total height of the player. For instance 4’ 22” would become 5’ 10”)
  + A screenshot of a computer

    Description automatically generated

Reflection

This project taught me how to use the ArrayList class. I have not used this class before doing this project and aside from reading about it in the textbook, I was not sure how to properly implement it in this case. This project was also a huge help in reviving my knowledge in programming with Java and how the ability to create the separate classes in Java. I had been used to C++ so returning to Java after a few years really helped thanks to this project. I have a deeper understanding of classes and the construction and use of immutable classes. I also had to remember how to do certain iterations through a loop for the ArrayList. In the program I had used a for loop as follows to iterate through the ArrayList: for (Player p : players) which iterates through each player object in the ArrayList and allows the program to evaluate each element in the list.