CMSC 215 Intermediate Programming
Programming Project 1

The first programming project involves writing a program to find the tallest basketball player whose age is less than or equal to the average of all the players. The program should contain three classes. The first class `Height` should contain two integer instance variables for the feet and inches. The class should be immutable, so it should have no setter methods. At a minimum it should contain the following methods:

- A constructor that accepts feet and inches constructs a `Height` object.
- A method `toInches` that returns the height in total inches.
- A method `toString` that returns the string representation of the height with a single quote following the feet and a double quote following the inches.

Regardless of what value for inches was supplied to the constructor the `toString` method should display the height normalized so the inches are less than 12.

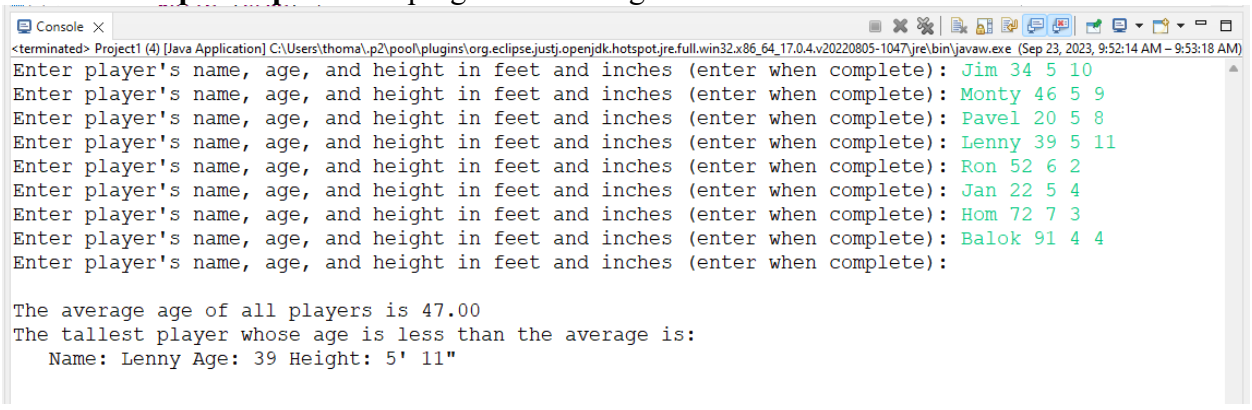
The second class `Player` should contain three instance variables that include the player's name, the player's height, which should be stored as the type `Height`, and the player's age. The class should be immutable, so it should have no setter methods It should contain the following methods:

- A constructor that accepts a player's name, height and age constructs a `Player` object.
- Getter methods for each of the instance variables
- A method `toString` that returns the string representation of a player with each field appropriately labeled.

The `toString` method of the `Player` class should call the `toString` method of the `Height` class.

The third class `Project1` should repeatedly prompt the user for the information for each of the players. It should create a `Player` object for each player and add the player to an `ArrayList`. As the players are read in, the total age of all players should be computed to enable the average to be calculated once all have been input. Once all player information has been inputted, the average age of all players should be output. Then the list of players should be traversed to find the tallest player whose age is less than or equal to the average age of all players.

Here is a **sample output** of the program for a single test case:



Documentation Requirements:
Follow the naming conventions previously provided in the course announcements. Please follow these requirements:

Make sure your Java program is using the recommended style such as:
• **Javadoc comment with your name as author, date, and brief purpose of the program**
• *Comments for variables and blocks of code to describe major functionality*
• *Meaningful variable names and prompts*
• Class names are written in upper CamelCase
• Constants are written in All Capitals
• Use proper spacing and empty lines to make your source code human readable

Deliverables:

You are to submit two files.

1. The first is a `.zip` file that contains all the source code for the project. The `.zip` file should contain only source code and nothing else, which means only the `.java` files. If you elect to use a package the `.java` files should be in a folder whose name is the package name. Every outer class should be in a separate `.java` file with the same name as the class name. Each file should include a comment block at the top containing your name, the project name, the date, and a short description of the class contained in that file.
2. The second is a Word document (PDF or RTF is also acceptable) that contains the documentation for the project, which should include the following:
    a. A UML class diagram that includes all classes you wrote. Do not include predefined classes.
    b. A test plan that includes test cases that you have created indicating what aspects of the program each one is testing. Include the **results of your testing** with screen captures clearly showing the output for each test case.
    c. A short paragraph on lessons learned from the project.