**Name**: Kharchenko, Vadym
**Project Name**: Programming Project 1 - Tallest Basketball Player
**Date**: 24 June, 2024
**Description**:

The goal of this program is to find the tallest player with an acceptable age.

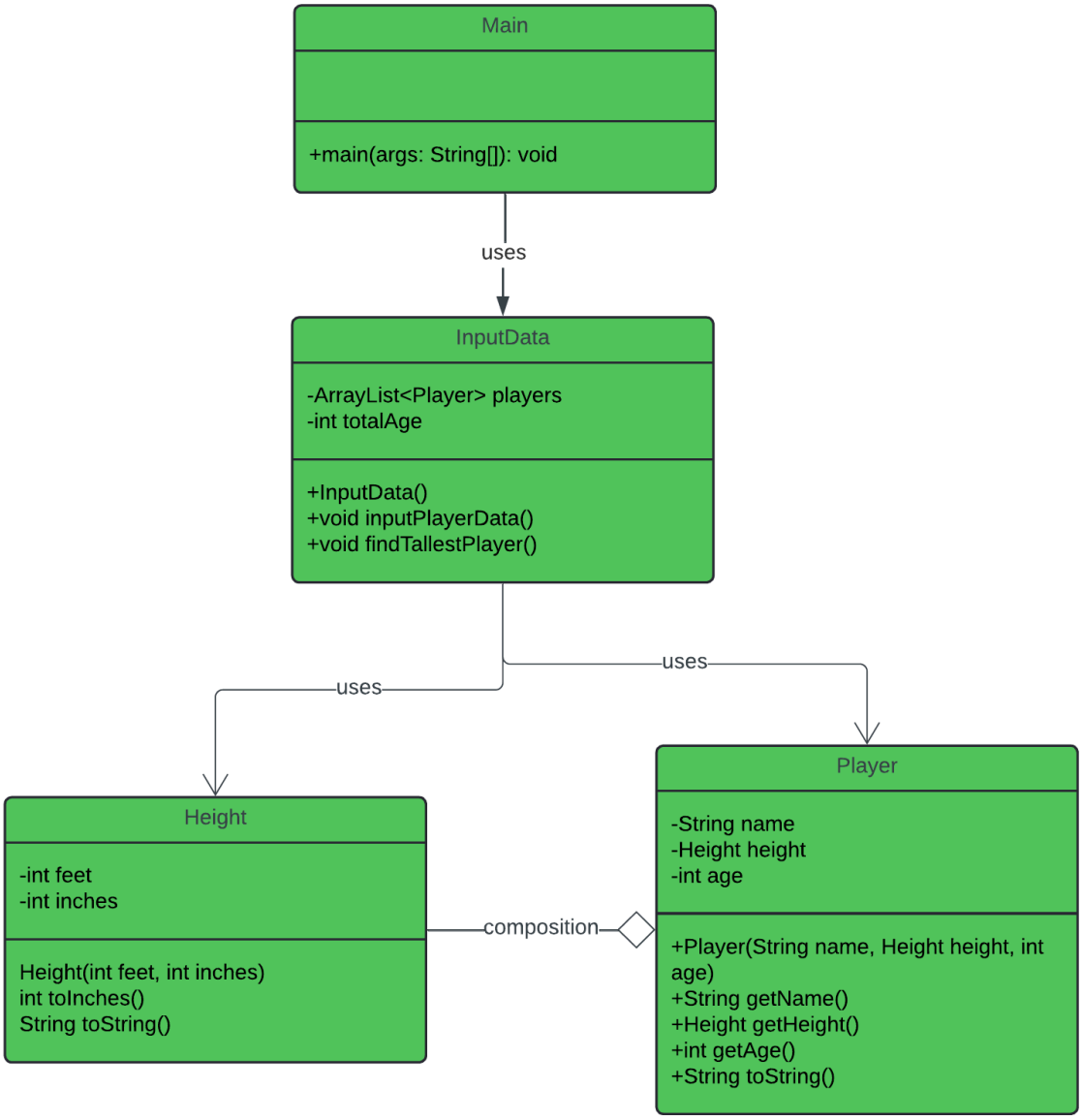
There are three classes.

The height class contains two integer instance variables for the feet and inches.

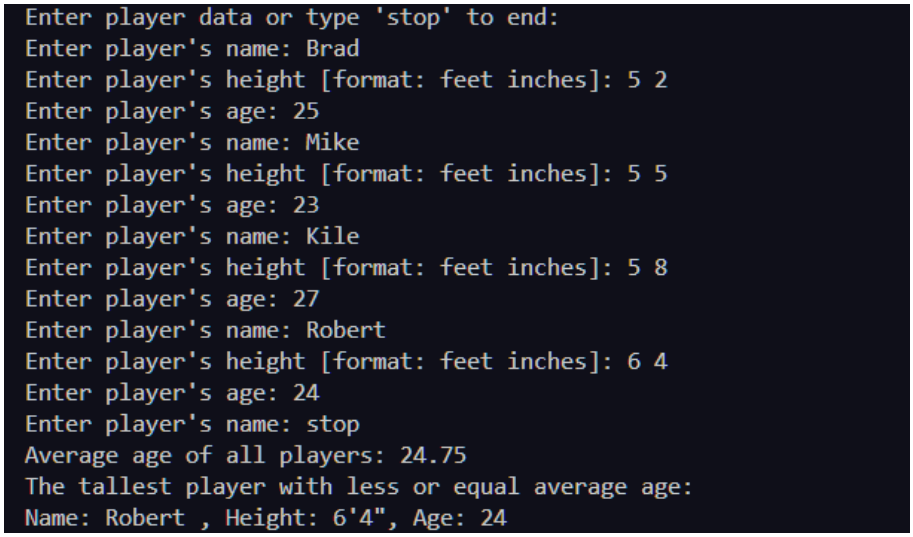
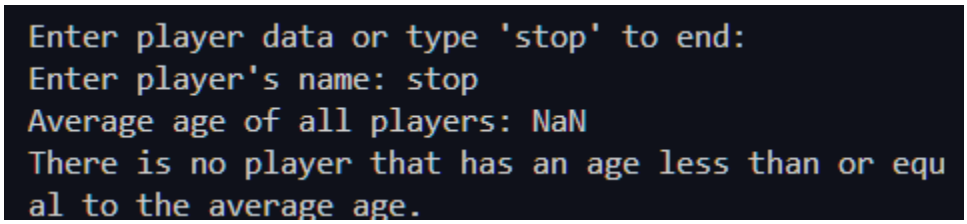
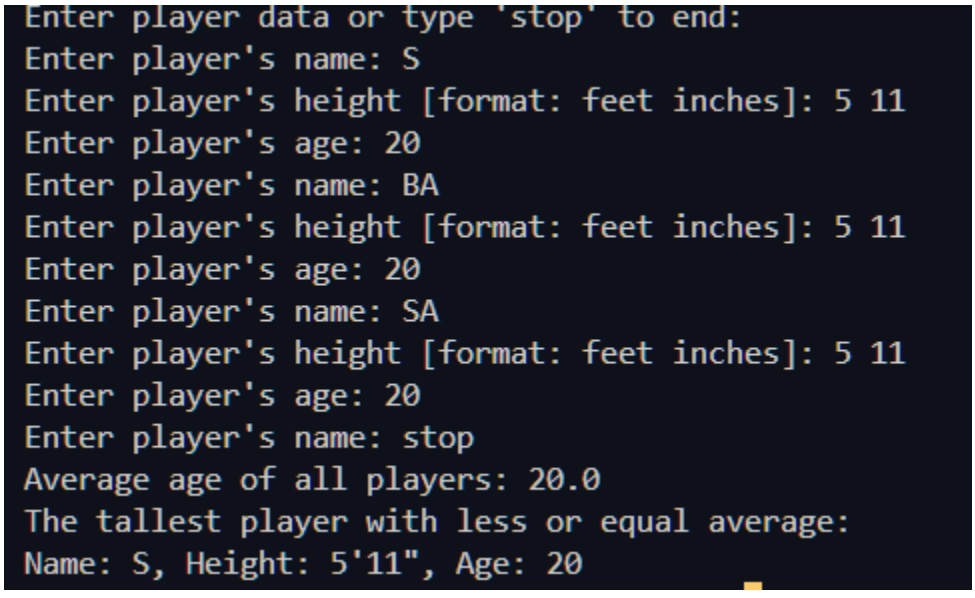
Player class contains three instance variables that include the player's name, the player's height, which is stored as the type of Height and the player's age.

InputData class prompts the user for the input and creates a player obj using the Height class calculates the average age and finds the tallest player with less or equal average age.

**UML Diagram**


Main
+main(args: String[]): void
uses
InputData
-ArrayList<Player> players
-int totalAge
+InputData()
+void inputPlayerData()
+void findTallestPlayer()
uses
uses
Height
-int feet
-int inches
Height(int feet, int inches)
int toInches()
String toString()
composition
Player
-String name
-Height height
-int age
+Player(String name, Height height, int age)
+String getName()
+Height getHeight()
+int getAge()
+String toString()

**Test Plan**

| | |
|---|---|
| Base functionality Test |  |
| Empty input |  |
| Handling similar Inputs |  |

**Lesson Learned**

This was a good practice on the general OOP principles. I used a modular concept, dividing the classes that can be reused elsewhere. I always struggled with error handling concepts, thus this time I tried implementing the bare bone error handling, which made it easier to troubleshoot. I faced a few problems with a general language syntax that took me a minute to figure out.