Homework Assignment 6

STEP 1 – Create Superclass and Subclasses

Create an Eclipse project named JohnDoeHw7, and use the default package, i.e., no package name should be provided. Inside the project, create the parent class (superclass) **Player** as specified in the table below. Save it in a file ***Player.java***.

Create the two child classes (subclasses) shown below. Save each class in a separate file – ***BaseballPlayer.java,*** and***BasketballPlayer.java.***

|  |  |  |
| --- | --- | --- |
| **Class names** | | |
| **Player**  ***(superclass)*** | **BaseballPlayer *(subclass of Player)*** | **BasketballPlayer  *(subclass of Player)*** | |
| id | number of hits | number of shots made | |
| player name | number of times at bat | number of shots attempted | |
| team name |  |  | |
| position |  |  | |
| salary |  |  | |
| commission rate |  |  | |
| constructors | constructors | constructors | |
| getters | getters | getters | |
| setters | setters | setters | |
| calculate comission | calculate statistics | calculate statistics | |
| determine status | determine status | |

**Requirements:**

In class **Player**:

* id is of int type; player name, team name, position is of String type; salary, commission rate is of double type.
* There should be two constructors, one is the default constructor with no parameter, and the other has all six parameters to initialize the six private data fields, respectively.
* There should be a public getter and setter for each of the six private data fields.
* There is one effector method to calculate the commission by multiplying the salary with the commission rate. This method return a double type.

In class **BaseballPlayer**:

* The two private data fields are int type.
* There should be two constructors, one is the default constructor with no parameter, and the other has all eight parameters to initialize the six inherited data fields plus the two private data fields, respectively.
* There should be a public getter and setter for each of the two private data fields.
* There are two effector methods:
  + **calculate statistics** : calculate the player’s batting average by dividing “the number of hits” by “the number of times at bats”, it should return a double type
  + **determine status** : boolean return type. If the batting average is more than **0.25**, return true; o/w return false

In class **BasketballPlayer**:

* The two private data fields are int type.
* There should be two constructors, one is the default constructor with no parameter, and the other has all eight parameters to initialize the six inherited data fields plus the two private data fields, respectively.
* There should be a public getter and setter for each of the two private data fields.
* There are two effector methods:
  + **calculate statistics :** Calculate the player’s shot percentage by dividing “the number of shots made” by “the number of shots attempted” , it should return a double type
  + **determine status :** boolean return type.If the shot percentage is more than **0.32**, return true; o/w return false

The two constant values mentioned in the determine status method as thresholds should be defined as public static final variables in the two subclasses, respectively.

**DO NOT ADD or DELETE** any method from the requirements above, and you should strictly follow the requriements.

STEP 2 – Create an application class that uses the Player class family

In the same Eclipse project, create a new class file named *AppRunner.java*.

This application file has the main method, and inside the main method, create two instances of BaseballPlayer, and two instances of BasketballPlayer.

For each instance, you should use the constructor with 8 parameters, and then plug in all 8 parameters directly in the constructor. You can use arbitrary values for the 8 parameters in each instance, as long as they are reasonable, for example, in basketball player, number of shots made should be less than number of shots attempted.

Then for each player instance, you need to output the six inherited data fields from Player class, and its commissions, then output the two private date fields, and then the statistics of this player, and the “keeping status” of this player: true or false.

An example output for a baseball player is below:

player id: 20

name: Mike Schmidt

team: Phillies

position: Third Base

salary: 1000000.00

commission rate: 0.02

commission: 20000.00

number of hits: 2234

number of times at bat: 8352

statistics: .27

keeping status: true

The output precision requirement: for monetary items, two digits after the decimal point; for statistics, one digit after the decimal point. Use ***System.out.printf()*** method to format the String type and boolean type (%s), double type(%f), and int type (%d) output items, with format specifier %s, %f, and %d, respectively.

For double type, you need to specify the number of precisions after the decimal point. Generate similar output as above for all four players. When you hardcode parameters to initialize the four players, you can carefully design the parameters so that in each sport category, one player has “keeping status” as true, and the other player has it as false.

STEP 4 – CHECK YOUR WORK FOR ADHERENCE TO PROGRAM SPECIFICATIONS

#### NOTE – ADDITIONAL SPECIFICATIONS!!

#### You do NOT need to use any arrays in this homework.

#### The names of the classes must be exactly as stated in the table on page 1.

#### You can not CHANGE, ADD or DELETE any instance variable or method of the Player class and its subclasses. However, inside each method, you have the freedom to use and name any local variable that you need. Also in the main method of the application class, you have the freedom to name and use any local variable.

#### For class Player and its subclasses, you must provide a constructor without arguments and a constructor with a full list of arguments to initialize all inherited data fields (if applicable), and all private data fields.

#### In the main method, you must use constructors with actual parameters to create the required six objects, by hardcoding the arguments in the contructors. NO USER INPUT is needed in this homework!

#### For the determine statistics method in each subclass, the division should happen between two int type variables, and the result should be double.

#### Except for the three threshold values in the two subclasses, all class data must be private, and you must provide public getters and setters for each private instance data.

#### The methods to calculate statistics must have the SAME name in all classes.

#### The methods to determine the player’s status must have the SAME name in all classes and must all return a boolean.

#### You do NOT need to use any abstract class or abstract method in this homework.