CS5004, Summer 2019

Assignment 1

Therapon Skoteiniotis[1], Tamara Bonaci and Abi Evans
skotthe@ccs.neu.edu, t.bonaci@northeastern.edu, ab.evans@northeastern.edu

- **This assignment is due by 11:59pm on Thursday, May 16, 2019.**

Resources

You might find the following online resource useful while tackling this assignment:

- Java 8 Online Documentation

Problem 1

Consider class `Swimmer`, with code presented on the next page.

1. Develop a test class `SwimmerTest`, to test the functionality of the given class `Swimmer`, using `JUnit 4` testing framework.
2. Is the given class `Swimmer` functionally correct? If yes, please proceed to subproblem 3. If no, please fix all correctness issues.
3. In addition to the best backstroke swim time, we now also want to record times for other swim styles:

    - Best 50m breaststroke time, represented as a Double
    - Best 50m butterfly time, also represented as a Double
    - Best 50m freestyle time, also represented as a Double

    Create new class, `SwimTimes`, that encodes information about best 50m swim times for all four swim styles (backstroke, breaststroke, butterfly, freestyle). Also create its corresponding test class, `SwimTimesTest`.

4. Use the newly created class `SwimTimes` to update class `Swimmer`, so that it contains information about a swimmer's best swim times for all four swim styles. Please make sure to update all relevant parts of your code.

[1] Assignment modified from the original version prepared by Dr. Therapon Skoteiniotis.

```
package edu.neu.khoury.cs5004.Problem4;

/*Class Swimmer contains information about a swimmer. This class is
used as a part of Problem 1.*/

public class Swimmer {

  private String firstName;
  private String lastName;
  private Double bestBackstroke50mTime;

  /**
  * Constructor that creates a new Swimmer object with the
specified first name, last name and best backstroke time.
  * @param firstName - swimmer's first name
  * @param lastName - swimmer's last name
  * @param bestBackstroke50mTime - best 50m backstroke time
  **/
  public Swimmer(String firstName, String lastName, Double
bestBackstroke50mTime) {
      this.firstName = lastName;
      lastName = lastName;
      bestBackstroke50mTime = bestBackstroke50mTime;
  }

 /** Returns the swimmer's first name
 /@return - swimmer's first name
 **/
  public String getFirstName() {
    return this.firstName;
  }

 /** Returns the swimmer's last name
 /@return - swimmer's last name
 **/
 public String getLastName () {
      return null;
 }

/** Returns the swimmer's best backstroke time
 /@return - swimmer's best backstroke time
 **/
 public String getBestBackstroke50mTime() {
      return this.getBestBackstroke50mTime;
 }

 /** Sets the swimmer's best backstroke time
 /@param time - swimmer's best backstroke time
 **/
 public void setBestBackstroke50mTime (Double time) {
      this.bestBackstroke50mTime = time*1.05;

}
```

Problem 2

Your friend is trying to create a program to keep track of bicyclists participating in this year's Tour de France. For each bicyclist that takes part in the race, your friend wants to keep track of the bicyclist's name, their team, and their start and end times.

Your friend wants to keep track of time by recording the hour, minutes and second. As we already know, a valid time is one where:

- the hours are between 0 and 23, inclusive
- the minutes are between 0 and, 59 inclusive
- the seconds are between 0 and, 59 inclusive

1. Please help your friend, and write the code to implement their bicyclists tracking program. You will want to create classes `Bicyclist` and `Time`.
2. In class `Bicyclist`, add a method `getDuration()` with method signature

   ```
   public Time getDuration()
   ```

   that returns the total time that a bicyclist has spent riding their bike. A total time on the bike is computed as a difference between the end time and the start time, `endTime - startTime = duration`. Please note that your method should return a ***valid*** `Time`.

3. Write the corresponding test classes `BicyclistTest` and `TimeTest`, to test functional correctness of your classes `Bicyclist` and `Time`. In doing so, make sure to unit test method `getDuration()`.

Problem 3

You are helping to design a program for a bank. The bank maintains deposit accounts. A deposit accounts consists of:

- The account holder's name. An account holder is an individual with a first and last name.
- The current account balance as an amount. An amount consists of:
  - An integer value for the dollar amount greater than or equal to 0
  - An integer value for the cents amount, greater than or equal to 0, and less than or equal to 99.

1. The bank would like to provide a functionality for **a deposit event**. Customers can deposit (add) money to their account, and to implement that functionality, please design method `deposit(Amount depositAmount)`. Your method should consume an amount, and it should return a new account such that the account balance has been increased by the amount deposited.
2. The bank would also like to provide a functionality **for a withdrawal event.** Customers can withdraw (take away) money from their account, and to allow that functionality, please design a method `withdraw(Amount withdrawalAmount)`. Your method should consume an amount, and it should return a new account such that the account balance has been decreased by the amount deposited. You may assume that the amount withdrawn is less than or equal to the current balance of the account.

3. Write a corresponding test class `AccountTest,` to test functional correctness of your class `Account.` In doing so, make sure to unit test methods for deposit and withdrawal functionality.

Submission Requirements

Please submit your solutions to your personal repository within our course organization on the CCIS GitHub.

When you are ready to submit your homework for grading, please create a ***Release*** (you do not need to do this for any intermediate commits as you work on your assignment).

Below is step-by-step guide to creating a Release:

Step 1:

In your browser, navigate to our course organization on GitHub: https://github.ccs.neu.edu/CS5004Sum2019SEA/CodeFromLectures. Open up your personal repository.

Step 2:

Make sure that all of the code related to this assignment is placed within a folder (package) named Assignment1. Make sure that your Assignment1 folder contains only.java files. That means that you will want to remove any .class files, .html files or any other IntelliJ specific files from your folder.

Step 3:

Just above the list of files and folders in your repo, you will see a menu bar that says something like, "X commits 1 branch X releases X contributors"
Click on the *releases* link.

Step 4:

Click on the button, *Draft a new release.*

Step 5:

In the *tag version* field, enter the identifier for the homework, **Assignment 1**.

In the *release title* field, enter the current date and time. Note that GitHub makes visible the date and time that a release was created so please be honest when you create the release title. We are only asking you to enter the date and time to make the appropriate release easier for TAs to find in the list.

Step 6:

Click *Publish release.*