Homework Assignment #5

CS5004 – Object-Oriented Design
Northeastern University – Silicon Valley
Summer 2019

Due 06/16 at 11:00pm PDT

Grading: Each programming problem is graded as follows

- A submission which does not compile gets 0.
- A submission which compiles but does something completely irrelevant gets 0.
- A submission which works (partially) correctly, gets (up to) %80 of the total credit.
- %20 is reserved for the coding style. Follow the coding style described in the book.

Formatting: Each class must reside in its own file. If you need to instantiate objects of class `A` in your class `B`, your `B.java` file must import `A.java` first. The names you choose for your classes should match the ones specified in each problem. Also, for each problem i, you must have a `Problem_i.java` file containing (only) the class which has the `main` method. Finally, you must put all files related to problem i into a folder named `problem_i`. The files *must be placed in the right sub-folder*. Make sure that your code complies. See the note in Problem 1 for an example.
Submission: For each problem i submit one `problem_i.zip` file to Blackboard. The zip file should maintain the structure of the sub-folders. See the note in Problem 1 for an example.

Problem 1 [50pts]. You are interested in keeping track of the team members and competition information for your school's annual entries in computer science programming competitions. Each team consists of exactly four team members. Every year your team competes in two competitions. As an initial start for your database, create a class named `Team` that has the following instance variables:

```
// Name for the team
String teamName;
// Names for each team members.
String name1, name2, name3, name4;
// Info on each competition
Competition competition1, competition2;
```

The class should also have a method that outputs the names of all team members and the competition information to the console. The `Competition` class contains variables to track the following:

```
String: Name of the competition,
String: Name of the winning team,
String: Name of the runner-up
Integer: Year of the competition
```

Implement the `Team` and `Competition` classes with appropriate constructor, accessor, and mutator methods. Write some tests.
In entering data for past competitions, you note that an entry is usually very similar to the previous year's entry. To help with the data entry, create a deep copy constructor for the Team class. Test your copy constructor by creating a copy of an existing team object, changing the competition information for the copy, and outputting the data for the original and the copy. The original object should be unchanged if your deep copy constructor is working properly.
Note: For this problem you must submit three files `Team.java`, `Competition.java` and `Problem_1.java`. The last one contains the `main` method (in public class `Problem_1`). Put the files in their correct subdirectories and compress them into `problem_1.zip` and submit to Blackboard. The same rules are to be followed for the remaining problems (in this and future assignments) unless otherwise specified.

Problem 2 [40pts]. Create a class to represent a container. The class Container should have the following properties

1. Maximum capacity of the container in liters.
2. Quantity of liquid at any given time in liters.

The following operations can be performed on the containers:

1. Completely fill a container.
2. Completely empty a container.
3. Transfer liquid from one container to another.

Define the class named `Container` that implements the properties and operations defined below. Create a constructor of the `Container` class that allows the user to specify the maximum capacity of the container in liters. Initially, assume that all the containers are empty.

Implement the following methods in the `Container` class.

- `quantity()` to return the current quantity of liquid at any given time in liters.
- `leftOver()` to return the quantity of liquid that can be filled in the current container before it is full.

- `full()` to fill the current container fully.
- `empty()` to make the container completely empty.
- `transfer()` to transfer a certain amount of liquid from one container to another.
- `displayQuantity()` to display the current quantity in liters.

Note: While transferring liquid from one container to another, check the maximum capacity of the container.

Problem 3 [50pts]. Define a class for rational numbers. A rational number is a number that can be represented as the quotient of two integers. For example, $\frac{1}{2}$, $\frac{3}{4}$, $\frac{64}{2}$, and so forth are all rational numbers. Represent rational numbers as two values of type `int`, one for the numerator and one for the denominator. Your class should have two instance variables of type `int`. Call the class `Rational`. Include a constructor with two arguments that can be used to set the instance variables of an object to any values. Also include a constructor that has only a single parameter of type `int`; call this single parameter `whole` and define the constructor so that the object will be initialized to the rational number $\frac{\texttt{whole}}{1}$. Also include a no-argument constructor that initializes an object to 0 (that is, to $\frac{0}{0}$). Note that the numerator, the denominator, or both may contain a minus sign. Define methods for addition, subtraction, multiplication, and division of objects of your class `Rational`. These methods should use a calling object and a single argument. For example, the `add` method (for addition) has a calling object and one argument. So `r1.add(r2)` returns the result of adding the rationals `r1` and `r2`. Define accessor and mutator methods as well as the methods `equals` and `toString`. You should include a method, `normalize`, to normalize the sign of the rational number so that the denominator is positive and the numerator is either positive or negative. For example, after normalization, $\frac{4}{-8}$ would be represented the same as $\frac{-4}{8}$. Also write a test program to test your class.