CS342: Software Design

Project #1

Due time: 9:00 pm on 1/26/2015
Submit using Blackboard web site
Total points: 60
Instructor: Ugo Buy

TA: Chihua Ma and Weixiang Shao

You are to code the following two classes in the Java language. For each class, make sure to implement all the fields and methods, whether static or non-static, according to the specifications below. Also, make sure to choose appropriate access levels, according to the principle of *information hiding*, for each field and method. Information hiding requires a class to prevent access to implementation of the class (i.e., the fields of the class) from clients of the class. Both classes should be implemented a single Eclipse project whose name is the concatenation of your last name and the word "Mp1". For instance, if your last name is "Smith", you should name your project "SmithMp1".

1. Define a public class *Point* modeling a point in a 2-dimensional reference system. Class *Point* defines two integer fields, *x* and *y*, with appropriate getter and setter methods.

   Class *Point* also defines two methods *slope()* and *distance()*. Method *slope()* takes as input a *Point* instance and returns a *double* number, which is the slope between the method receiver and the argument point. The slope is defined as the ratio of the difference between the *y* coordinates of the two points and the difference of their *x* coordinates. (Hint: Class *Double* defines static fields *POSITIVE_INFINITY* and *NEGATIVE_INFINITY*. Method *slope()* may return the constant *Double.POSITIVE_INFINITY* or *Double.NEGATIVE_INFINITY* when the slope is infinite.)

   Similarly, method *distance()* returns the Euclidean distance between the method receiver and the argument *Point* as a *double*. Use Pitagoras's theorem to compute this distance. (Hints: Static method *Math.sqrt()* returns the square root of its argument.)

   Write also a no-arg constructor for class *Point* and a constructor that takes two integers, which are used to initialize *x* and *y*.

   Finally, class *Point* defines a public static method *main()*, which creates a number of *Point* instances for the purpose of testing the class. In *main()* code three test cases in which the slope and distance functions are computed for the following pairs of points {(10, 10), (4, 2)}, {(2, 10), (-1, -2)}, and {(3, 3), (0, 6)}. Print the results of those computations on the Eclipse console using appropriate output statements.

2. Define a public class *Date*, which models dates in the Gregorian calendar system. Each date has three integer nonstatic fields, namely *day, month* and *year*. In addition, *Date* has a static field *daysInMonth*, an integer array with the length of the months of the year in days (e.g., { *31, 28, 31, 30, . . .* }).

   Class *Date* has just one constructor that takes as input three integer parameters that are used to initialize the three non-static fields. The constructor uses the static field and method *isLeapYear()* below to check that this is a valid day/month combination, for instance, printing an error message if someone is tries to define the date February 31, 2015. When an error is detected, the new instance is set to the default value of January 1, 2001 after the error message is printed.

Class *Date* defines a static method *isLeapYear()*, that takes as input a non-negative integer number and checks whether the argument number is a leap year. The method returns a boolean value depending on whether the argument is a leap year or not.

Finally, class *Date* defines a nonstatic method *prettyPrint()* that returns a string formatting the receiver with the full month name (e.g., as in “20 January 2015”).

Write five test cases in static method *main()*, constructing, formatting (with method *prettyPrint()*) and printing on the Eclipse console the following input cases 1/1/2001, 2/31/1998, 2/29/2016, 8/15/1945, and 12/31/1999. (Hint: Use a static field to hold the names of the months in class *Date*.)

You must work alone on this project. Submit this project using the submission link in the assignment’s page on Blackboard. No late submissions will be accepted. **Important: You are not allowed to use the predefined Java classes *Date* and *Point* for this project**.