Homework 8

Classes

CS 5060 Intensive Programming, Fall 2012

74 points

Due: 3:59 pm November 19, 2012

Assignment description

In this assignment you will have to create classes, use inheritance and use polymorphism.

Computing the capacitance of a circuit (32 points)

Circuit classes (15 points): Create the following classes, similar to the ones for computing the resistance explained during lecture, to compute the total capacitance of a circuit with multiple capacitors:

- `CapacitorCircuit` (2 points),
- `SeriesCCircuit` as a subclass of `CapacitorCircuit` (3 points),
- `ParallelCCircuit` as a subclass of `CapacitorCircuit` (3 points),
- `Capacitor` (2 points).

The `CapacitorCircuit` must be an abstract class with the following methods:

- `getCapacitorCount()` which returns the total number of capacitors in the circuit (3 points), and
- `getCapacitance()` which returns the total capacitance of the circuit (2 points).[1]

Unit tests (15 points): Create classes:

- `CapacitorCircuitTest` (4 points),
- `SeriesCCircuitTest` (4 points),
- `ParallelCCircuitTest` (4 points),
- `CapacitorTest` (3 points)

to test each one of your classes. You should test all the public methods of your classes (remember to test circuits that have a combination of series and parallel sub-parts).

Packages (2 points):

- Put your classes in a package named: `cs5060.homework.hw08.circuit`.
- Put your test classes in a package named: `cs5060.homework.hw08.circuit.test`.

[1] Refer to `http://en.wikipedia.org/wiki/Series_and_parallel_circuits` for formulas.

Piggy bank (42 points)

Piggy bank classes (19 points): Create a `Coin` class with the following methods (2 points):

- constructor with a value parameter (1 point),
- `getValue()` method which returns the value of the coin (1 point).

Create a `PiggyBank` class with the following methods (17 points):

- constructor with capacity (maximum number of coins the piggy bank can hold) parameter (1 point),
- `isEmpty()` method (1 point),
- `isFull()` method (2 point),
- `getCapacity()` method (1 point),
- `addCoin(Coin)` method (3 points),
- `getNumberOfCoins()` method (2 points),
- `getTotalValue()` method (3 points),
- `isBroken()` method (1 point),
- `breakBank()` method (3 point).

Notes (4 points):

- `isEmpty` should return `true` when the piggy bank is broken.
- `isFull` should return `false` when the piggy bank is broken.
- `addCoin` should throw `FullPiggyBankException` (2 points) when the piggy bank is full.
- `addCoin` should throw `BrokenPiggyBankException` (2 points) when the piggy bank is broken.
- `getNumberOfCoins` and `getTotalValue` should return 0 when the piggy bank is broken.
- `breakBank` should return a list of all the coins, and throw a `BrokenPiggyBankException` if the piggy bank is already broken.
- Make `FullPiggyBankException` and `BrokenPiggyBankException` subclasses of `Exception`.

Unit tests (17 points): Create a `PiggyBankTest` class to test the methods of your `PiggyBank` class. Create a test method for each method you have to test. The tests for each method of the `PiggyBank` class are worth the same as the method.

Packages (2 points):

- Put your classes in a package named: `cs5060.homework.hw08.piggybank`.
- Put your test classes in a package named: `cs5060.homework.hw08.piggybank.test`.

Submission.

Submit a `zip` file with your code files. Make sure you include the package folders.

Include your name and A number at the top of each source file. Name the `zip` file `hw08_firstName_lastName.zip`. For example, if your name is John Smith, name the file `hw08_John_Smith.zip`.