

CS161: Introduction to Computer Science

Homework Assignment 5

due 2/21 by 11:59pm

Interacting Classes

So far we've focused on writing a single standalone Java class, e.g. the `Orca` class. In practice, however, we often need to create multiple classes that interact with one another to accomplish some goal. Two common ways in which classes interact are,

1. A method in one class can take as input a variable of the same class type or another class type. Similarly, a method in one class can return a variable of the same class type or another class type. An example of this is the `BankAccount` and `Bank` classes.
2. A class can have instance variables which themselves have class type. An example of this is the `Brain`, `Stomach`, and `Organism` classes.

Open BlueJ and create a new project named `hw5`.

Written Exercises

Double click on the sheet of notebook paper icon in BlueJ. This will open a README file where you can type your answers to the following exercises:

- Consider the following Java program:

```
/**
 * This class stores information about a person
 */
public class Person {
    // Instance variables
    private int age;
    private String name;

    // The constructor
    public Student (String theName, int theAge){
        age = theAge;
        theName = name;
    }

    public void introduceSelf(){
        System.out.println("Hi, my name is " + name + ".");
    }

    // Convert age to equivalent dog age
    public int dogAge(){
        int temp = age / 7.0;
        return temp;
    }
}
```

This class contains two errors: one compile-time error and one logical error. Identify both errors. Then correct both errors by re-writing the code.

Pair of Dice

In this week's assignment, you will write a class whose instance variables themselves have class type. In particular, this assignment asks you to write a Java class called `PairOfDice` that represents a pair of dice.



While I encourage you to write a separate class with a `main()` method to test your `PairOfDice` class – i.e. a driver or tester class – it is not required for this assignment so that you have more time to study for the exam.

1. You will first need to create a `Die` class. You can simply copy the `Die` class that I wrote *or* you can recreate the `Die` class without looking as practice for the exam.
2. Create a new Java class named `PairOfDice`. Add the following methods:
 - A default constructor (i.e. a constructor that takes no inputs). This constructor should create two 6-sided dice.
 - A constructor that takes a single `int` parameter representing the number of sides for both dice.
 - A constructor that takes two `int` parameters representing the number of sides for each die respectively. In this case, the two die can have a different number of sides.
 - A `roll()` method that rolls both dice. It is up to you whether this method returns a value or not.
 - A method to get the face value of the first die. A method to get the face value of the second die.
 - A method that returns the sum of the two face values.
 - A method to return the number of times the dice have been rolled.
 - A `toString()` method that builds and returns a string containing the face value of both dice as well as the number of times the dice have been rolled.
3. Finally, your code should have the appropriate Javadoc and inline (`//`) comments for full credit.

Below is an example of how someone might use your `PairOfDice` class:

```
public static void main(String[] args){
    // Create a 10-sided die and a 20-sided die
    System.out.println("Creating a 10-sided and 20-sided die...\n");
    PairOfDice pair = new PairOfDice(10, 20);

    // Roll and print the face values
    pair.roll();
    int first = pair.getFirstFaceValue();
    int second = pair.getSecondFaceValue();
    System.out.println("We rolled a " + first + " and a " + second);

    pair.roll();
    first = pair.getFirstFaceValue();
    second = pair.getSecondFaceValue();
    System.out.println("We rolled a " + first + " and a " + second);

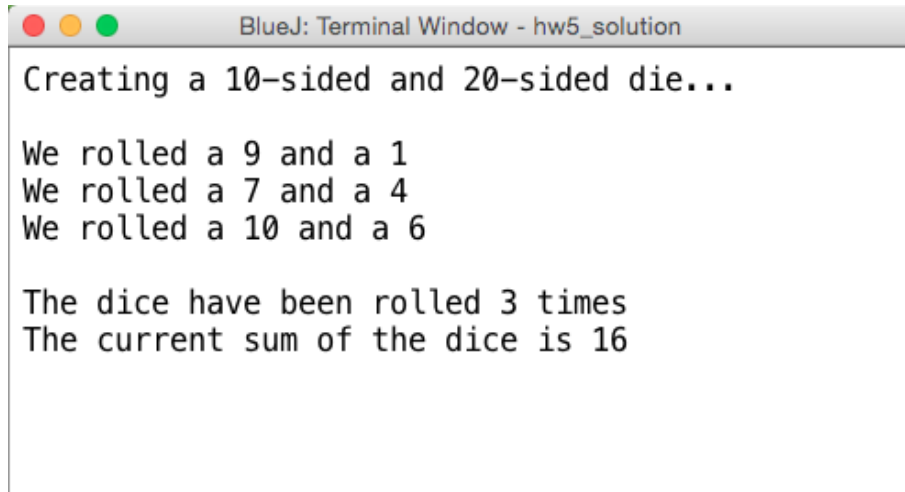
    pair.roll();
    first = pair.getFirstFaceValue();
    second = pair.getSecondFaceValue();
    System.out.println("We rolled a " + first + " and a " + second);
}
```

```
System.out.println();

// Get the number of rolls so far
int rolls = pair.getNumRolls();
System.out.println("The dice have been rolled " + rolls + " times");

// Get the sum of the dice
System.out.println("The current sum of the dice is " + pair.getSum());
}
```

Here is what would be printed to the screen:



```
BlueJ: Terminal Window - hw5_solution
Creating a 10-sided and 20-sided die...

We rolled a 9 and a 1
We rolled a 7 and a 4
We rolled a 10 and a 6

The dice have been rolled 3 times
The current sum of the dice is 16
```

Submitting your homework assignment

Submit your `hw5` folder with your `Die` and `PairOfDice` class inside. If you wrote a third class with a `main` method to test your `PairOfDice` class, feel free to include that as well.

Don't forget to rename your folder *before* zipping it and submitting it to Moodle.