Northeastern University
CSYE 6200 – Concepts of Object-Oriented Design
Fall 2022, Prof. Jones Yu

Lab 5

Due: 11:59 pm, Saturday, October 15

1. Problem 1

To help you understand wrapper classes, we want you to design your `Integer-like` wrapper class. Please design a class named `MyInteger`. The class contains:

- An `int` data field named `value` that stores the `int` value represented by this object.
- A constructor that creates a `MyInteger` object for the specified `int` value.
- A getter method that returns the `int` value.
- The methods `isEven()`, `isOdd()`, and `isPrime()` that return true if the value in this object is even, odd, or prime, respectively.
- The static methods `isEven(int)`, `isOdd(int)`, and `isPrime(int)` that return `true` if the specified value is even, odd, or prime, respectively.
- The static methods `isEven(MyInteger)`, `isOdd(MyInteger)`, and `isPrime(MyInteger)` that return `true` if the specified value is even, odd, or prime, respectively.
- The methods `equals(int)` and `equals(MyInteger)` that return `true` if the value in this object is equal to the specified value.
- A static method `parseInt(char[])` that converts an array of numeric characters to an `int` value.
- A static method `parseInt(String)` that converts a string into an `int` value.

Please write a program to test all methods in the class.

Expected results:

```
n1 is even? false                                          // Set n1 to 7
n1 is prime? true
15 is prime? false                                         // Verify number 15
parseInt(char[]) for { '4', '3', '7', '8' } = 4378
parseInt(String) for "4378" = 4378
n2 is odd? false                                           // Set n2 to 24
45 is odd? true                                            // Verify number 45
n1 is equal to n2? false
n1 is equal to 5? false
```

Note that texts marked with blue colors are the results of method calls. You need to find suitable method calls to output the expected results. As for texts marked with green colors, they remind you what to do to get the expected result on the left side.

2. Problem 2

We want you to create a class `RoomPeople` that can be used to record the number of people in the rooms of a building. The class has the attributes:

- `numberInRoom` - the number of people in a room
- `totalNumber` - the total number of people in all rooms as a `static` variable

The class has the following methods:

- `addOneToRoom` - adds a person to the room and increases the value of `totalNumber`
- `removeOneFromRoom` - removes a person from the room, ensuring that `numberInRoom` does not go below zero, and decreases the value of totalNumber as needed
- `getNumber` - returns the number of people in the room
- `getTotal` - a `static` method that returns the total number of people

Please write a program to test the class `RoomPeople`.

Expected results:

```
Add two to room a and three to room b                    //addOneToRoom()
Room a holds 2
Room b holds 3
Total in all rooms is 5
Remove two from both rooms                               //removeOneFromRoom()
Room a holds 0
Room b holds 1
Total in all rooms is 1
Remove two from room a (should not change the values)    //removeOneFromRoom()
Room a holds 0
Room b holds 1
Total in all rooms is 1
```

Note that texts marked with blue colors are the results of method calls. You need to find suitable method calls to output the expected results. As for texts marked with green colors, they remind you which method to call to perform the action, such as using `addOneToRomm()` for "Add two to room a and three to room b".

3. Submission Requirement:

This is an individual assignment, and each student needs to submit his/her solution to the Canvas. The submission needs to be a .zip file containing following data:

3.1 A document (either .doc or .pdf format) that describes:

- Problem description: a short description of the issue you are solving
- Analysis:
  - What design/solution/algorithm you use to solve the problem?
  - What are the difficulties you encounter?
  - …
- Source code: copy & paste your source code to the report (i.e., all .java files)
- Screenshots of sample runs: show that the code has been reasonably tested

3.2 The project with source codes:

- Source project: The Eclipse Java project that has all of your changes.