Bravo, Erick & Yu, Kelly

Lab 3 Design Document

P5.10 (Ch. 5)

Step 1: Find out which methods you are asked to supply.

- Convert a positive integer into the Roman numeral system.
- Get the resulting Roman numeral.

Step 2: Specify the public interface.

- public void conversionToRomanNumerals()
- public String getRomanNumeral()
- public RomanNumerals(int inputNumeral)

Step 3: Document the public interface.

```
/**
        A class that can convert a positive integer between 1 and 3,999 into a Roman numeral.
*/
public class RomanNumerals()
{
}
        /**
        Constructs an empty string for the Roman Numeral to be outputted, and intakes an integer input
        by the user.
        @param inputNumeral the integer that the user will input in; this integer will be converted into a
        Roman numeral
        */
        public RomanNumerals(int inputNumeral)
        {
        }

        /**
        Converts the given integer input into Roman numerals.
        */
        public void conversionToRomanNumerals()
        {
        }
        /**
        Returns the resulting Roman numeral.
        @return the Roman numeral string.
        */
        public String getRomanNumerals()
        {
        }
```

Step 4: Determine instance variables.

```
public class RomanNumerals()
{
        private String convertedNumeral;
        private int given;
}
```

Step 5: Implement constructors and methods.

```
public RomanNumerals(int inputNumeral)
{
        convertedNumeral = “”;
        given = inputNumeral;
}

public void conversionToRomanNumerals()
{
        while (given >= 1000) {
                convertedNumeral = convertedNumeral + "M";
                given = given - 1000;
        }
        while (given >= 900) {
                convertedNumeral += "CM";
                given = given - 900;
        }
        while (given >= 500) {
                convertedNumeral += "D";
                given = given - 500;
        }
        while (given >= 400) {
                convertedNumeral += "CD";
                given = given - 400;
        }
        while (given >= 100) {
                convertedNumeral += "C";
                given = given - 100;
        }
        while (given >= 90) {
                convertedNumeral += "XC";
                given = given - 90;
        }
        while (given >=50) {
                convertedNumeral += "L";
                given = given - 50;
        }
        while (given >=40) {
                convertedNumeral += "XL";
                given = given - 40;
        }
        while (given >= 10) {
                convertedNumeral += "X";
                given = given - 10;
        }
        while (given >= 9) {
                convertedNumeral += "IX";
                given = given - 9;
        }
        while (given >= 5) {
                convertedNumeral += "V";
                given = given - 5;
        }
        while (given >=4) {
                convertedNumeral += "IV";
                given = given - 4;
        }
        while (given >= 1) {
                convertedNumeral += "I";
                given = given - 1;
        }
}

public String getRomanNumerals()
{
        return convertedNumeral;
}
```

Step 6: Test your class. // Done in the code.

```
import java.util.Scanner;
public class RomanNumeralsTester
{
        public static void main (String[] args)
        {
                Scanner in = new Scanner(System.in);
                System.out.print("Please input a number: ");
                int input = in.nextInt();
                if (input < 1 || input > 3999)
                {
                        System.out.print("That is not within the range of Roman Numerals.");
                }
                else if (input >= 1 && input <= 3999)
                {
                        RomanNumerals number = new RomanNumerals(input);
                        number.conversionToRomanNumerals();
                        System.out.print("In Roman Numerals: ");
                        System.out.print(number.getRomanNumerals());
                }

        }
}
```

```
100

Enter a number from 0 to 3999: In Roman Numerals:  C

------------------------------------------------------------------------

BUILD SUCCESS

------------------------------------------------------------------------

Total time:  2.872 s

Finished at: 2020-06-26T13:55:23-07:00

------------------------------------------------------------------------

23

Enter a number from 0 to 3999: In Roman Numerals:  XXIII

------------------------------------------------------------------------

BUILD SUCCESS

------------------------------------------------------------------------

Total time:  1.853 s

Finished at: 2020-06-26T13:55:41-07:00

------------------------------------------------------------------------

3999

Please input a number: In Roman Numerals:  MMMCMXCIIIIIIIII
```

```
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  4.874 s
Finished at: 2020-06-26T14:01:57-07:00
------------------------------------------------------------------------
Please input a number between 1 and 3,999: 1978
In Roman Numerals: MCMLXXVIII
Please input a number between 1 and 3,999: 4000
That is not within the range of Roman Numerals.
------------------------------------------------------------------------
```

P6.18 (Ch. 6)

Step 1: Find out which methods you are asked to supply.

- Create an equation that shows half life once medicine is consumed by the patient

Step 2: Specify the public interface.

- Public class halfLife a place where the half life function and the

Step 3: Document the public interface.

- Scanner med = new Scanner(System.in);

Step 4: Determine instance variables.

- double Ao, int i

Step 5: Implement constructors and methods.

- System.out.println("A/Ao after hr " +i+ ": "+(Ao * Math.exp(-i * (Math.log(2) /6.0))));

Step 6: Test your class. // Done in the code.

-

```
Import java.util.*;
public class halfLife
{
        public static void main(string[] args)
        {
                // user input to declare how much radioactive medicine they took at T =0
                System.out.println("Please enter radioactive medicine given now: ");
                Scanner med = new Scanner(System.in);
                double Ao = med.nextDouble();
                System.out.println("A/Ao in a patient's body for the next 24 hours: ");
```

```
                              for (int i = 1; i <= 24; i++)
                              {
                              System.out.println("A/Ao after hr " +i+ ": "+(Ao * Math.exp(-i * (Math.log(2)
                              /6.0))));
                              }
                    }
          }

/*
Please enter radioactive medicine given now:
6
A/Ao in a patient's body for the next 24 hours:
A/Ao after hr 1: 5.345392308842036
A/Ao after hr 2: 4.762203155904599
A/Ao after hr 3: 4.242640687119286
A/Ao after hr 4: 3.7797631496846193
A/Ao after hr 5: 3.367386144928119
A/Ao after hr 6: 3.0
A/Ao after hr 7: 2.6726961544210184
A/Ao after hr 8: 2.3811015779522995
A/Ao after hr 9: 2.121320343559643
A/Ao after hr 10: 1.8898815748423097
A/Ao after hr 11: 1.68369307246406
A/Ao after hr 12: 1.5
A/Ao after hr 13: 1.3363480772105092
A/Ao after hr 14: 1.1905507889761497
A/Ao after hr 15: 1.0606601717798214
A/Ao after hr 16: 0.9449407874211551
A/Ao after hr 17: 0.8418465362320298
A/Ao after hr 18: 0.7500000000000002
A/Ao after hr 19: 0.6681740386052547
A/Ao after hr 20: 0.5952753944880749
A/Ao after hr 21: 0.5303300858899107
A/Ao after hr 22: 0.47247039371057753
A/Ao after hr 23: 0.42092326811601505
A/Ao after hr 24: 0.375
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  13.137 s
Finished at: 2020-06-26T12:17:26-07:00
------------------------------------------------------------------------
*/
```