CS-213: Assignment 1

Soumen Pradhan | 1912176

15 . 04 . 2021

1. Count number of vowels and consonants

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Q01_main {
    static Pattern vowReg =
        Pattern.compile("[a-zA-Z&&[aeiouAEIOU]]");
    static Pattern consReg =
        Pattern.compile("[a-zA-Z&&[^aeiouAEIOU]]");

    static BufferedReader bf = new BufferedReader(
        new InputStreamReader(System.in));

    public static void main(String[] args)
        throws IOException
    {
        String str = bf.readLine();
        int vow = letCount(str, true);
        int cons = letCount(str, false);

        System.out.printf(
            "Vowels: %d%nConsonants: %d%n", vow, cons
        );
    }

    static int letCount(String S, Boolean vowel) {
        if (S.isEmpty())
            return 0;

        Matcher match =
            vowel ? vowReg.matcher(S) : consReg.matcher(S);

        int count = 0;
        while (match.find())
            count++;

        return count;
    }
}
```

2. QuickSort Implementation

```java
public class Q02_main {
    public static void main(String[] args)
    {
        int[] arr = new int[15];
        fillRand(arr);

        printInt(arr);
        Sorter.quickSort(arr);
        printInt(arr);
    }

    static void fillRand(int[] arr) {
        for (int i = 0; i < arr.length; i++)
            arr[i] = (int)(Math.random() * 1000);
    }

    static void printInt(int[] arr) {
        for (int i = 0; i < arr.length; i++)
            System.out.print(arr[i] + " ");
        System.out.println();
    }
}

class Sorter {
    private static void swap(int[] arr, int i, int j) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }

    private static int partition(int[] arr, int lo, int hi) {
        int pivot = arr[hi], i = lo - 1;

```

```
            for (int j = lo; j < hi; j++) {
                if (arr[j] < pivot) {
                    i++;
                    swap(arr, i, j);
                }
            }
            swap(arr, i+1, hi);
            return i+1;
        }

        private static void quickSort(int[] arr, int lo, int hi) {
            if (lo < hi) {
                int pivot = partition(arr, lo, hi);
                quickSort(arr, lo, pivot-1);
                quickSort(arr, pivot+1, hi);
            }
        }

        static void quickSort(int[] arr) {
            quickSort(arr, 0, arr.length);
        }
    }
```

3. Area class inheritence

```
class Area {
    double dim1, dim2;

    double area() {
        return 0;
    }
}

class Circle extends Area {
    Circle(double radius) {
        this.dim1 = this.dim2 = radius;
    }

    double area() {
        return Math.PI * Math.pow(this.dim1, 2);
    }
}

class Rectangle extends Area {
```

```java
    Rectangle(double ht, double len) {
        this.dim1 = ht;
        this.dim2 = len;
    }

    double area() {
        return dim1 * dim2;
    }
}

class Triangle extends Area {
    Triangle(double ht, double base) {
        this.dim1 = ht;
        this.dim2 = base;
    }

    double area() {
        return dim1 * dim2 / 2;
    }
}
```

4. Interface error

```java
public interface SomethingIsWrong {
    void aMethod(int aValue);
}
```

5. The Ouput will be:

```
THIRD

SECOND

FIRST
```

The constructor of the child class `C` calls its parent class (`B` then `A`) constructors.

6. The code will not compile. Error: No member `j` is defined in class `A`. Hence, accessing `a.j` is illegal (`a` is an object of class `A`).

7. Given two classes,

```java
class ClassA {
    public void methodOne(int i) {}
    public void methodTwo(int i) {}
    public static void methodThree(int i) {}
    public static void methodFour(int i) {}
}

class ClassB extends ClassA {
    public static void methodOne(int i) {}
    public void methodTwo(int i) {}
    public void methodThree(int i) {}
    public static void methodFour(int i) {}
}
```

a) `methodTwo` overrides

b) `methodFour` hides

c) `methodOne` and `methodThree` produce compile errors

8. Count gross and dozen

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Q08_main {
    static BufferedReader bf = new BufferedReader(
        new InputStreamReader(System.in));

    public static void main(String[] args)
        throws IOException
    {
        int num = Integer.parseInt(bf.readLine());
        dozEgg(num);
        grossEgg(num);
    }

    static void dozEgg(int num) {
        int doz = num / 12;
        int rem = num % 12;

        System.out.printf("%d dozen and %d eggs.%n", doz, rem);
    }

    static void grossEgg(int num) {
        int gross = num / 144;
        int doz = (num % 144) / 12;
        int rem = num % 12;

        System.out.printf(
            "%d gross, %d dozen, and %d eggs.%n",
            gross, doz, rem
        );
    }
}
```

9. Find the number with largest number of divisors

```java
public class Q09_main {
    public static void main(String[] args)
    {
        int[] max = findMaxDiv(10000);
        System.out.printf(
            "%d: %d divisors%n", max[0], max[1]
        );
    }

    static int[] findMaxDiv(int num) {
        int[] divs = new int[num+1];

        for (int i = 1; i <= num; i++)
            for (int j = 1; j <= num/i; j++)
                divs[i * j]++;

        int maxAt = 0;
        for (int k = 0; k < divs.length; k++)
            maxAt = divs[k] > divs[maxAt] ? k : maxAt;

        return new int[]{maxAt, divs[maxAt]};
    }
}
```

10. Design 'Counter' class

```java
class Counter {
    private int count;

    Counter() { this.count = 0; }

    void increment() { this.count++; }
    int getValue() { return count; }
}
```

11. Print employees with more than 20 years' experience

```java
import java.util.ArrayList;

public class Q11_main {
    public static void main(String[] args)
    {
        ArrayList<Employee> employeeData;
        employeeData.removeIf(e -> e.yearsWithCompany >= 20);

        for (var e : employeeData)
            System.out.println(
                e.firstName + e.lastName + ": " + e.hourlyWage
            );
    }
}

class Employee {
    String lastName, firstName;
    double hourlyWage;
    int yearsWithCompany;
}
```

12. Split strings into groups of letters

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Q12_main {
    static BufferedReader bf = new BufferedReader(
        new InputStreamReader(System.in));

    public static void main(String[] args)
        throws IOException
    {
        String[] tokens = bf.readLine()
                              .split("[\\W]+");
        for (var s : tokens)
            System.out.println(s);
    }
}
```

13. Definition of Class 'room'

```java
class room {
    int room_no;
    int room_type;
    double room_area;
    boolean ACmachine;

    void set_data(int num, int type, double area, boolean ac) {
        this.room_no = num;
        this.room_type = type;
        this.room_area = area;
        this.ACmachine = ac;
    }

    void display_data() {
        System.out.printf(
            "Room No: %d%nRoom Type: %d%n" +
            "Room Area: %.2f%nAC in room: %b%n",
            room_no, room_type, room_area, ACmachine
        );
    }
}
```

14. Definition of Class ‘SimpleObject’

```java
class SimpleObject {
    SimpleObject() {
        System.out.println("SimpleObject constructed.");
    }
}
```

15. Illustrate use of ‘static’ keyword

```java
public class Q15_main {
    public static void main(String[] args) {
        var statRes = staticShowcase.mul(2.5, 2);

        var st = new staticShowcase();
        var nonStat = st.add(2.5, 2);

        System.out.println(statRes + "%n" + nonStat);
    }
}

class staticShowcase {
    static double mul(double a, double b) { return a * b; }
    double add(double a, double b) { return a + b; }
}
```

16. Inheritence of Class ‘shape’

```
class shape {
    void draw() {}
    void erase() {}
}

class circle extends shape {}
class square extends shape {}
class triangle extends shape {}
```

17. The Output will be:

```
I am a dog
```