

Assignment - Introduction to Java - 2

Q1) Write a program to display values of enums using a constructor & getPrice() method (Example display house & their prices)

```
1 package Module_2.Q1;
2
3 public enum House { no usages
4     SMALL( price: 1000), no usages
5     MEDIUM( price: 10000), no usages
6     LARGE( price: 100000); no usages
7
8     private final int price; 2 usages
9
10    House(int price){ 6 usages
11        this.price = price;
12    }
13
14    public int getPrice(){ no usages
15        return price;
16    }
17
18 }
```

```
1 package Module_2.Q1;
2
3 public class Q1 { 2 usages
4     public static void ENUM() { 1 usage
5         House small = House.SMALL;
6         House medium = House.MEDIUM;
7         House large = House.LARGE;
8
9         System.out.println("House Type :"+ small.name() +" "+ small.getPrice());
10        System.out.println("House Type :"+ medium.name() +" "+medium.getPrice());
11        System.out.println("House Type :"+ large.name() +" "+large.getPrice());
12    }
13 }
```

Q2) Create a User class with fields: firstname, lastname, age, phonenumber. Write a program which accepts values of user fields from commandline, create object and append that to a text file. After every user creation the program should prompt: "Do you want to continue creating users? (Type QUIT to exit)" and keep on accepting values and writing to file until user quits.

```
package Module_2.Q2;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

class User{ 2 usages
    String firstName; 2 usages
    String lastName; 2 usages
    int age; 2 usages
    String PhoneNumber; 2 usages

    User(String firstName, String lastName, int age, String PhoneNumber){ 1 usage
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
        this.PhoneNumber = PhoneNumber;
    }

    @Override
    public String toString(){
        return "First Name: "+ firstName+" \n"+
            "Last Name: "+lastName+" \n"+
            "Age : "+age+" \n"+
            "Phone Number: "+PhoneNumber + "\n";
    }
}

public class Q2 { 2 usages
    public static void Creator() { 1 usage
        Scanner sc = new Scanner(System.in);
        String quit;

        try(FileWriter writer = new FileWriter("output.txt", append: true)){
            while(true){
                System.out.print("Enter First Name: ");
            }
        }
    }
}
```

```

public class Q2 { 2 usages
    public static void Creator() { 1 usage
        String quit;

        try(FileWriter writer = new FileWriter("output.txt", append: true)){
            while(true){
                System.out.print("Enter First Name: ");
                String firstName = sc.nextLine();
                System.out.print("Enter Last Name: ");
                String lastName = sc.nextLine();
                System.out.print("Enter Age: ");
                int age = Integer.parseInt(sc.nextLine());
                System.out.print("Enter Phone Number: ");
                String PhoneNumber = sc.nextLine();
                User firstUser = new User(firstName, lastName, age, PhoneNumber);

                writer.write(str: firstUser.toString() + "\n");
                writer.flush();

                System.out.print("User Saved in File Successfully \n");

                System.out.println("Do you want to Continue (Y/N): (Type QUIT to exit)");
                quit = sc.nextLine();
                if(quit.equals("QUIT")){
                    break;
                }
            }

            System.out.print("\n All Users in File are \n");
            try(BufferedReader br = new BufferedReader(new FileReader("output.txt"))){
                String line;
                while((line = br.readLine()) != null){
                    System.out.println(line);
                }
            }
        } catch (IOException e) {

```

```
public class Q2 { 2 usages
    public static void Creator() { 1 usage

        writer.write(str.firstUser.toString() + "\n");
        writer.flush();

        System.out.print("User Saved in File Successfully \n");

        System.out.println("Do you want to Continue (Y/N): (Type QUIT to exit)");
        quit = sc.nextLine();
        if(quit.equals("QUIT")){
            break;
        }
    }

    System.out.print("\n All Users in File are \n");
    try(BufferedReader br = new BufferedReader(new FileReader(fileName: "output.txt"))){
        String line;
        while((line = br.readLine()) != null){
            System.out.println(line);
        }
    }
} catch (IOException e) {
    throw new RuntimeException(e);
} finally {
    sc.close();
}
}
```

```
src
47
48
writer.write(" " + firstUser.toString() + "\n");
writer.flush();

Run Main x
G [ ] [ ] [ ] [ ] [ ]
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/akash/Downloads/idea-IU-251.26094.121/lib/idea_rt.jar=44043
Enter First Name: q
Enter Last Name: l
Enter Age: 23
Enter Phone Number: 2132142314
User Saved in File Successfully
Do you want to Continue (Y/N): (Type QUIT to exit)
QUIT

All Users in File are

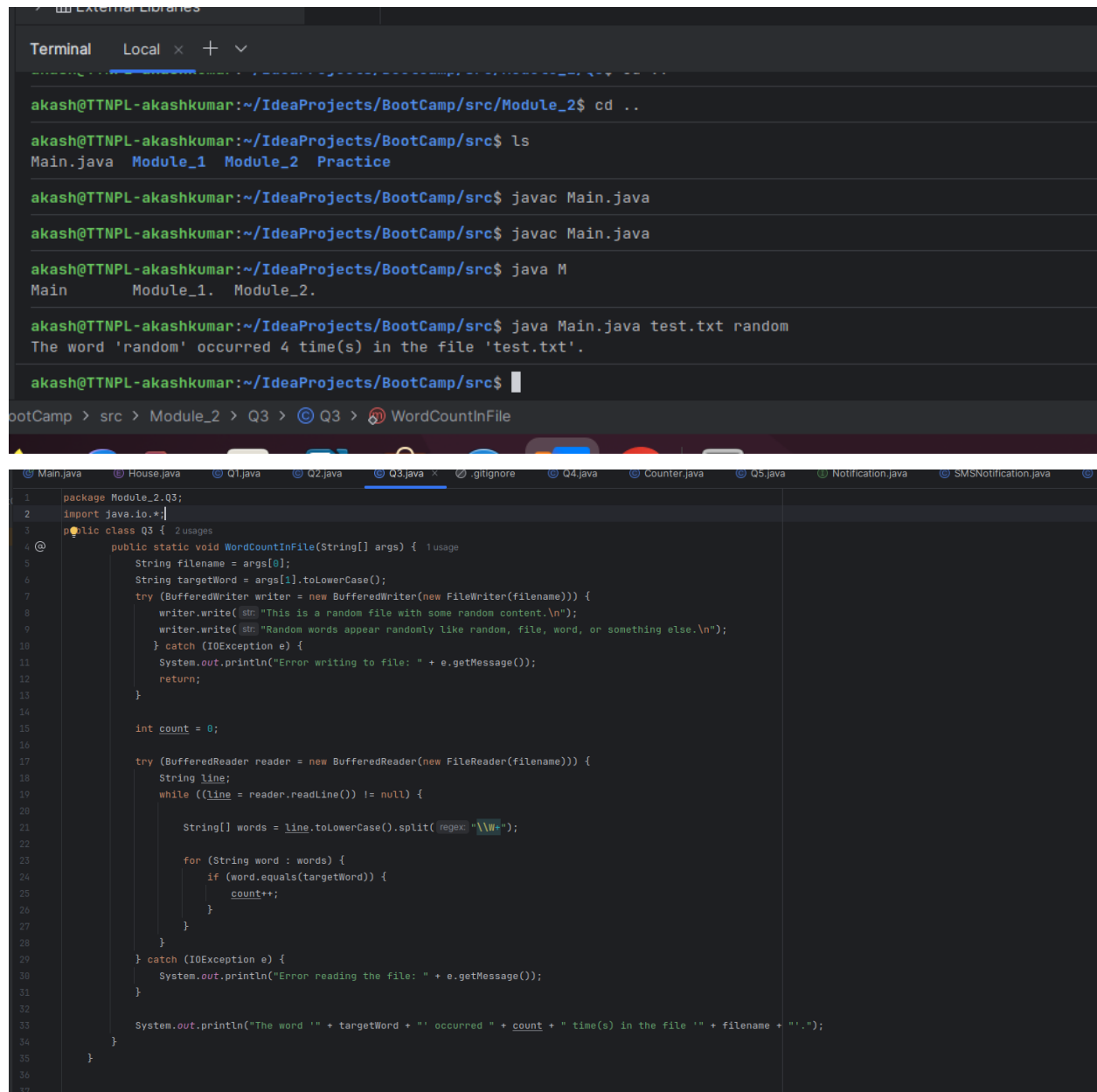
First Name: Akash
Last Name: Kumar
Age : 23
Phone Number: 12345

First Name: h
Last Name: w
Age : 1
Phone Number: 133232

First Name: q
Last Name: l
Age : 23
Phone Number: 2132142314

Process finished with exit code 0
```

Q3) Write a program to count number of occurrences of a word in a file. The file name and word should be supplied through commandline.



```
Terminal Local x + v
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src/Module_2$ cd ..
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src$ ls
Main.java Module_1 Module_2 Practice
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src$ javac Main.java
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src$ javac Main.java
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src$ java M
Main Module_1. Module_2.
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src$ java Main.java test.txt random
The word 'random' occurred 4 time(s) in the file 'test.txt'.
akash@TTNPL-akashkumar:~/IdeaProjects/BootCamp/src$

bootCamp > src > Module_2 > Q3 > Q3 > WordCountInFile

1 package Module_2.Q3;
2 import java.io.*;
3 public class Q3 { 2 usages
4 @ public static void WordCountInFile(String[] args) { 1 usage
5     String filename = args[0];
6     String targetWord = args[1].toLowerCase();
7     try (BufferedWriter writer = new BufferedWriter(new FileWriter(filename))) {
8         writer.write("This is a random file with some random content.\n");
9         writer.write("Random words appear randomly like random, file, word, or something else.\n");
10    } catch (IOException e) {
11        System.out.println("Error writing to file: " + e.getMessage());
12    }
13    return;
14
15    int count = 0;
16
17    try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
18        String line;
19        while ((line = reader.readLine()) != null) {
20
21            String[] words = line.toLowerCase().split("\\W+");
22
23            for (String word : words) {
24                if (word.equals(targetWord)) {
25                    count++;
26                }
27            }
28        }
29    } catch (IOException e) {
30        System.out.println("Error reading the file: " + e.getMessage());
31    }
32
33    System.out.println("The word '" + targetWord + "' occurred " + count + " time(s) in the file '" + filename + "'.");
34
35    }
36
37 }
```

Q4)Write a program to show application of Factory Design Pattern.

```
1 package Module_2.Q4;
2
3 public interface Notification { 5 usages 2 implementations
4     void notifyuser(); no usages 2 implementations
5 }
```

```
1 package Module_2.Q4;
2
3 public class EmailNotification implements Notification { 1 usage
4     @Override 1 usage
5     public void notifyuser() {
6         System.out.println("Email Notification");
7     }
8 }
9
```

```

1 package Module_2.Q4;
2
3 public class SMSNotification implements Notification { 1 usage
4     @Override 2 usages
5     public void notifyuser() {
6         System.out.println("SMS Notification");
7     }
8 }
9


```

```

1 package Module_2.Q4;
2
3 public class FactoryNotification { 2 usages
4     Change signature
5     public Notification createNotification(String type) { 2 usages
6         if(type == null || type.isEmpty())
7         {
8             return null;
9         }
10        if(type.equalsIgnoreCase( anotherString: "SMS")){
11            return new SMSNotification();
12        } else if(type.equalsIgnoreCase( anotherString: "Email")){
13            return new EmailNotification();
14        }
15        return null;
16    }
17 }
18

```



```
1 package Module_2.Q4;  
2  
3  
4   
5 public class Q4 { 1 usage  
6     public static void Factory(){ 1 usage  
7         FactoryNotification factory = new FactoryNotification();  
8         Notification not1 = factory.createNotification( type: "Email");  
9         not1.notifyuser();  
10        Notification not2 = factory.createNotification( type: "SMS");  
11        not2.notifyuser();  
12    }  
13 }  
14
```

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/akash/Downloads/idea-IU-251.26094.1  
Email Notification  
SMS Notification  
  
Process finished with exit code 0
```

Q5)Write a program to show application of Singleton Design Pattern.

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/akash/Downloads/idea-IU-251.26094.1
Count from c1: 3
Count from c2: 3
c1 and c2 are the same instance (Singleton).

Process finished with exit code 0
```

```
2
3 public class Counter { 7 usages
4     private static Counter instance; 3 usages
5
6     private int count = 0; 2 usages
7
8     private Counter() {} 1 usage
9
10    public static Counter getInstance() { 2 usages
11        if (instance == null) {
12            instance = new Counter();
13        }
14        return instance;
15    }
16
17    public void increment() { 3 usages
18        count++;
19    }
20
21    public int getCount() { 2 usages
22        return count;
23    }
24 }
25
```

```
1 package Module_2.Q5;
2
3 public class Q5 { 2 usages
4     public static void Singleton() { 1 usage
5         Counter c1 = Counter.getInstance();
6         Counter c2 = Counter.getInstance();
7
8         c1.increment();
9         c1.increment();
10
11         c2.increment();
12
13         System.out.println("Count from c1: " + c1.getCount());
14         System.out.println("Count from c2: " + c2.getCount());
15
16         if (c1 == c2) {
17             System.out.println("c1 and c2 are the same instance (Singleton).");
18         }
19     }
20
21 }
```