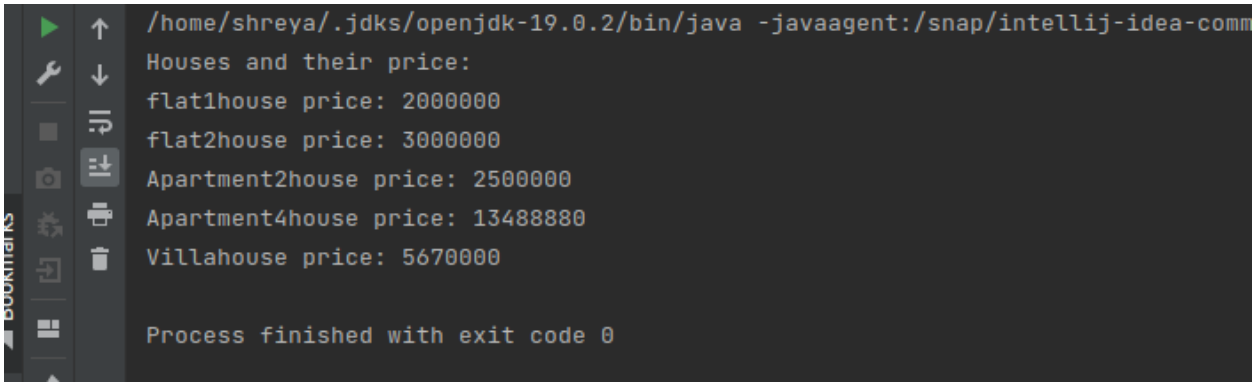
Introduction to java-3

Assignment

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

```
public class Enumexmp {

   public static enum house{

        flat1(2000000),

        flat2(3000000),

        Apartment2(2500000),

        Apartment4(13488880),

        Villa(5670000);

        private int price;

        house (int price){

            this.price=price;

        }

        private int getPrice(){

            return price;

        }

   }

   public static void main(String[] args){

        System.out.println("Houses and their price:");

        for(house h: house.values()){

            System.out.println(h+"house price: "+h.getPrice());

        }

   }

}
```



2) Create a User class with fields: firstname, lastname, age, phonenumber. Write a program which accepts values of user fields from command line, 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.

```
import java.io.*;

import java.util.*;



public class User {

   private String firstname,lastname;

   private int age;

   long pno;

   public User(String firstname,String lastname,int age,long pno){

        this.firstname=firstname;

        this.lastname=lastname;

        this.age=age;

        this.pno=pno;

   }

   public String toString(){

        return "First name: "+firstname+" Last name : "+lastname+" age : "+age+"
phone number : "+ pno;

   }
```

```
public static void main(String[] args) throws IOException {

        Scanner sc=new Scanner(System.in);

        FileWriter writer=null;

        try{

            writer= new FileWriter("src/Users.txt");

        }

        catch(IOException e){

            e.printStackTrace();

        }

        while(true){

            System.out.println("Enter first name: ");

            String firstname=sc.nextLine();

            System.out.println("Enter last name: ");

            String lastname=sc.nextLine();

            System.out.println("Enter age: ");

            int age=sc.nextInt();

            System.out.println("Enter phone number: ");

            long pno=sc.nextInt();

            User u=new User(firstname,lastname,age,pno);

            try{

                writer.write(u.toString()+"\n");

                writer.flush();

            }

            catch(IOException e){

                e.printStackTrace();

            }
```

```
            System.out.println("Do you want to continue creating users ?");

            System.out.println("if not, type QUIT");

            String ch=sc.next();

            if(ch.equalsIgnoreCase("quit")){

                break;

            }

        }

        try{

            writer.close();

        }

        catch(IOException e){

            e.printStackTrace();

        }

}

}
```

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

```
import java.util.*;

import java.io.*;

public class CountOccurences {

   public static void main(String[] args) {

       Scanner sc=new Scanner(System.in);

       System.out.println("Enter the File name : ");

       String filename="src/"+sc.nextLine();

       System.out.println("Enter the Word to be counted for occurrences");

       String word=sc.nextLine();

       String line;

       int count=0;

       try{

           FileReader fi;



           fi = new FileReader(filename);

           BufferedReader br=new BufferedReader(fi);
```

```
            while((line=br.readLine())!=null){
                String words[]=line.split(" ");
                for(String w:words){
                    if(w.equals(word)){
                        count++;
                    }
                }
            }
            fi.close();
            System.out.println(" Count : "+count);
        }
        catch(IOException e){
            System.out.println("File not found error.");
        }
    }
}
```

```
/home/shreya/.jdks/openjdk-19.0.2/bin/java -javaagent:/snap/intellij-idea-comm
Enter the File name :
Shreya.txt
Enter the Word to be counted for occurrences
for
 Count : 5

Process finished with exit code 0
```

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

```
public interface Mobile {
```

```
    String getModel();

    int getBatteryCapacity();

}
```

```
public class SamsungGalaxy implements Mobile {

    private String model;

    private int battery;

    public SamsungGalaxy(){

        this.model="Samsung Galaxy S21";

        this.battery=4000;

    }

    @Override

    public String getModel() {

        return model;

    }



    @Override

    public int getBatteryCapacity() {

        return battery;

    }

}
```

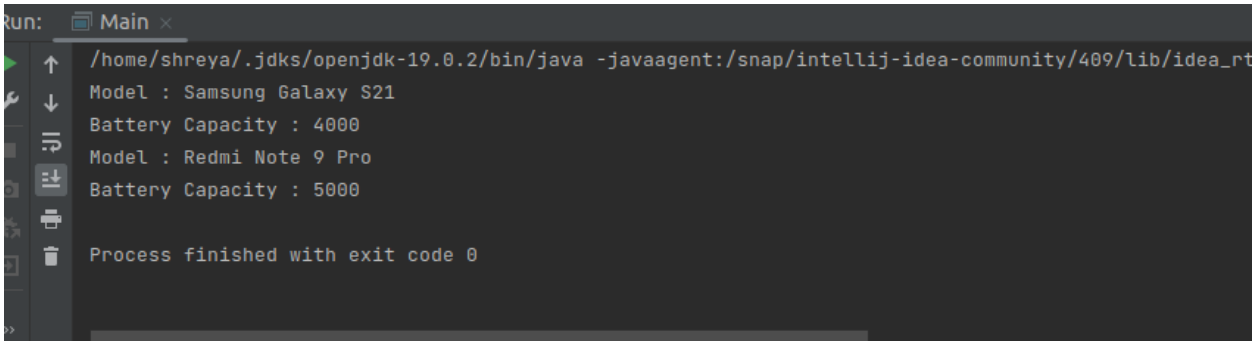
```
public class Redmi implements Mobile{

    private String model;

    private int battery;

    public Redmi(){
```

```
        this.model="Redmi Note 9 Pro";
        this.battery=5000;
    }
    @Override
    public String getModel() {
        return model;
    }

    @Override
    public int getBatteryCapacity() {
        return battery;
    }
}
```

```
public class MobileFactory {
    public static Mobile createMobile(String type){
        if(type.equalsIgnoreCase("Samsung")){
            return new SamsungGalaxy();
        }
        else if(type.equalsIgnoreCase("Redmi")){
            return new Redmi();
        }
        else{
            System.out.println("Invalid Choice");
            return null;
        }
```

```
    }

}
```

```
public class Main {

    public static void main(String[] args) {

        Mobile samsung=MobileFactory.createMobile("Samsung");

        Mobile redmi=MobileFactory.createMobile("Redmi");

        System.out.println("Model : "+samsung.getModel());

        System.out.println("Battery Capacity : "+samsung.getBatteryCapacity());

        System.out.println("Model : "+redmi.getModel());

        System.out.println("Battery Capacity : "+redmi.getBatteryCapacity());

    }

}
```



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

```
public class MobilePhone {

    private static MobilePhone instance;

    private MobilePhone(){}

    public static synchronized MobilePhone getInstance(){

        if(instance ==null){
```

```
            instance=new MobilePhone();

        }

        return instance;

    }

    public void makeCall(String number){

        System.out.println("Calling "+number);

    }

    public void sendText(String number,String message){

        System.out.println("Sending text to "+number+" "+message);

    }

}
```

```
public class Main {

    public static void main(String[] args) {

        MobilePhone phone= MobilePhone.getInstance();

        phone.makeCall("9843446443");

        phone.sendText("3454567565","Hello, how was your day ?");

    }

}
```