PART-3

Package & Interface

GitHub Repository Link: https://github.com/21ce114/JAVA-Practicals.git

Question 1:	Create an abstract class GeometricObject as the superclass for Circle and Rectangle. GeometricObject models common features of geometric objects. Both Circle and Rectangle contain the getArea() and getPerimeter() methods for computing the area and perimeter of a circle and a rectangle. Since you can compute areas and perimeters for all geometric objects, so define the getArea() and getPerimeter() methods in the GeometricObject class. Give implementation in the specific type of geometric object. Create TestGeometricObject class to display area and perimeter of Rectangle and Triangle, compare area of both and display results. Design of all classes are given in the following UML diagram.
Answer:	

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link:
https://github.com/21ce114/JAVA-Practicals.git
AIM : Create an abstract class GeometricObject as the superclass
for Circle and
Rectangle. GeometricObject models common features of geometric
objects. Both
Circle and Rectangle contain the getArea() and getPerimeter()
methods for
computing the area and perimeter of a circle and a rectangle.
Since you can
compute areas and perimeters for all geometric objects, so define
the getArea() and
getPerimeter() methods in the GeometricObject class. Give
implementation in the
specific type of geometric object. Create TestGeometricObject
class to display area
and perimeter of Rectangle and Triangle, compare area of both and
display results.
Design of all classes are given in the following UML diagram.
*/

import java.util.*;

abstract class GeometricObject{
    public abstract void getArea();
    public abstract void getPerimeter();

}
```

```
class Circle extends GeometricObject{
    int r=0;
    Circle(int r){
        this.r=r;
    }
    public void getArea() {
        System.out.println("The area of the circle
is:"+3.14*r*r);
    }
    public void getPerimeter() {
        System.out.println("The Perimeter of the circle is:"+
2*3.14*r);
    }
}
class Rectangle extends GeometricObject{
    int l,b;
    Rectangle(int l,int b){
        this.l=l;
        this.b=b;
    }
    public void getPerimeter() {
        System.out.println("The Perimeter of the Rectangle
is:"+2*(l+b));
    }
    public void getArea() {
        System.out.println("The area of the Rectangle is:"+ l*b);
    }
}
public class TestGeometricObject{

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the Radius of Circle:");
        int r=sc.nextInt();
        System.out.println("Enter the Length of Rectangle:");
        int l=sc.nextInt();
        System.out.println("Enter the breadth of Rectangle:");
        int b=sc.nextInt();

        Circle cir=new Circle(r);
        Rectangle rec=new Rectangle(l,b);

        cir.getArea();
        cir.getPerimeter();
        rec.getArea();
        rec.getPerimeter();
    }
```

```
}
Output:
 Enter the Radius of Circle:
3
 Enter the Length of Rectangle:
5
 Enter the breadth of Rectangle:
5
 The area of the circle is:28.259999999999998
 The Perimeter of the circle is:18.84
 The area of the Rectangle is:25
 The Perimeter of the Rectangle is:20
 PS C:\Users\HARSH>
```

Question 2:

Write a program to create a default method in an interface IPrinter. Create an interface IPrinter and IScanner. You can assume variables and methods for both interfaces. Create a concrete class to implement both the interfaces. Create 5 objects of the class, store it in Vector and display the result of the vector.

Answer:

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link:
https://github.com/21ce114/JAVA-Practicals.git
AIM : Write a program to create a default method in an interface
IPrinter. Create
an interface IPrinter and IScanner. You can assume variables and
methods for both
interfaces. Create a concrete class to implement both the
interfaces. Create 5 objects of the
class, store it in Vector and display the result of the vector.*/
import java.util.*;

interface IPrinter{
    default void show(Vector v1){
        System.out.println(v1);
    }

}
interface IScanner{
    default Vector get(){
        Vector v = new Vector(5);
        Scanner sc = new Scanner(System.in);
        int n;
        for(int i=0;i<5;i++){
        System.out.println("Enter your element(integer value):
");
        n=sc.nextInt();
        v.add(n);
```

```
        }
        return v;
    }
}
public class Practical_2 implements IPrinter, IScanner {
    public static void main(String[] args) {
        Vector vec = new Vector(5);
        IPrinter p = new Practical_2();
        IScanner s = new Practical_2();

        vec =s.get();
        p.show(vec);

    }
}
```

Output:

```
PS C:\Users\HARSH>  & 'C:\Program Fi
Enter your element(integer value):
1
Enter your element(integer value):
3
Enter your element(integer value):
2
Enter your element(integer value):
4
Enter your element(integer value):
5
[1, 3, 2, 4, 5]
PS C:\Users\HARSH>
```

Question 3:

WAP that illustrate the interface inheritance. Interface P is extended by P1 and P2 interfaces.1, 2 Interface P12 extends both P1 and P2. Each interface declares one method and one constant. Create one class that implements P12. By using the object of the class invokes each of its method and displays constant.

Answer:

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link:
https://github.com/21ce114/JAVA-Practicals.git
AIM : WAP that illustrate the interface inheritance. Interface P
is
extended by P1 and P2 interfaces.1, 2 Interface P12
extends both P1 and P2. Each interface declares one
method and one constant. Create one class that
implements P12. By using the object of the class
invokes each of its method and displays constant.*/

interface P
{
    int a=1;
    void displayP();
```

```
}
interface P1 extends P
{
    int b=2;
    void displayP1();
}
interface P2 extends P
{
    int c=3;
    void displayP2();

}
interface P12 extends P1,P2
{
    int d=4;
    void displayP12();
}
class imp implements P12
{
    public void displayP()
    {
        System.out.println("P called and constant:"+a);

    }
    public void displayP1()
    {
        System.out.println("P1 called and constant:"+b);
    }
    public void displayP2()
    {
        System.out.println("P2 called and constant:"+c);
    }
    public void displayP12()
    {
        System.out.println("P12 called and constant:"+d);
    }
}
class Practical3_3 extends imp
{
    public static void main(String[]args)
    {
        imp n=new imp();
        n.displayP();
        n.displayP1();
        n.displayP2();
        n.displayP12();
    }
}
```

Output:

```
PS C:\Users\HARSH>  & 'C:\Pr
P called and constant:1
P1 called and constant:2
P2 called and constant:3
P12 called and constant:4
PS C:\Users\HARSH>
```

Question 4:

Develop a Program that illustrate method overriding concept.

Answer:

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link:
https://github.com/21ce114/JAVA-Practicals.git
AIM : Develop a Program that illustrate method overriding
concept.*/

class parent{
    void show(){}
}
class child1 extends parent{
    //method overriding.
    void show(){
        System.out.println("This is child'1 class method.");
    }
}

class child2 extends parent{
    //method overriding.
    void show(){
        System.out.println("This is child'2 class method.");
    }
}
public class Practical3_4{

 public static void main(String[] args) {
    child1 obj1 = new child1();
    obj1.show();
    child2 obj2 = new child2();
    obj2.show();
  }
}
```

Output:

```
PS C:\Users\HARSH>  & 'C:\Program
This is child'1 class method.
This is child'2 class method.
PS C:\Users\HARSH>
```

Question 5:

Write a java program which shows importing of classes from other user define packages.

Answer:

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git
AIM :Write a java program which shows importing of classes from other
user define packages. */

package part_3;

public class Practical3_5_Package_file {
    public void print(){
        System.out.println("Code for Part 3. ");

    }
    public static void main(String[] args) {
        Practical3_5_Package_file pac = new Practical3_5_Package_file();
        pac.print();
    }
}
/*-------------------------------------------------------*/

import part_3.Practical3_5_Package_file;


public class Practical3_5_import {
    public static void main(String[] args) {
      Practical3_5_Package_file obj = new Practical3_5_Package_file();
        obj.print();
    }
}
```

Output:

Problems Javadoc Declaration
<terminated> Practical3_5_import [Java Ap
Code for Part 3.

Question: 6

Write a program that demonstrates use of packages & import statements.

Answer:

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link:
https://github.com/21ce114/JAVA-Practicals.git
AIM :Write a program that demonstrates use of
     packages & import statements. */
```

Package File:

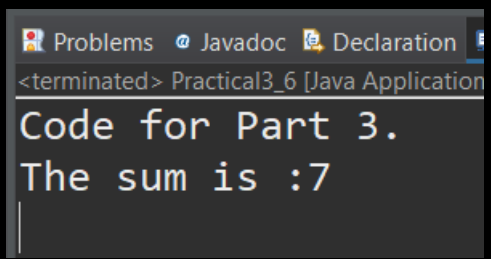
```
package part_3;

public class Practical3_5_Package_file {
    public void print(){
        System.out.println("Code for Part 3. ");

    }
    public static void main(String[] args) {
        Practical3_5_Package_file pac = new
Practical3_5_Package_file();
        pac.print();
    }
}
/*-------------------------------------------------------*/

package part_3;

public class PackageFileForPractical3_6 {
     public void add (int a,int b) {
          System.out.println("The sum is :"+(a+b));
     }

}
/*-------------------------------------------------------*/
```

Importing Package method:

```
// FIRST IMPORTING METHOD.
public class Practical3_6 {
     public static void main(String[] args) {
```

```
            // FIRST IMPORTING METHOD.
            //USING FULLY QUALIFIED NAME.
            part_3.Practical3_5_Package_file obj = new
part_3.Practical3_5_Package_file();
            obj.print();

            part_3.PackageFileForPractical3_6 obj1 = new
part_3.PackageFileForPractical3_6();
            obj1.add(5,2);

      }
}
/*-------------------------------------------------------*/
// Second IMPORTING METHOD.
// Using packagename.classname
import part_3.PackageFileForPractical3_6;
import part_3.Practical3_5_Package_file;
public class Practical3_6 {
      public static void main(String[] args) {

            Practical3_5_Package_file obj = new
Practical3_5_Package_file();
            obj.print();

            PackageFileForPractical3_6 obj1 = new
PackageFileForPractical3_6();
            obj1.add(5,2);

      }
}
/*-------------------------------------------------------*/

// Third IMPORTING METHOD.
// Using Using packagename.*  To access all the classes.
import part_3.*;

public class Practical3_6 {
      public static void main(String[] args) {

            Practical3_5_Package_file obj = new
Practical3_5_Package_file();
            obj.print();

            PackageFileForPractical3_6 obj1 = new
PackageFileForPractical3_6();
            obj1.add(5,2);

      }
}
```

Output:



Question: 7

Write a program that illustrates the significance of interface default method.

Answer:

```
/*ID: 21CE114
Name: Harsh Rana
Git Repository Link:
https://github.com/21ce114/JAVA-Practicals.git
AIM: Write a program that illustrates the significance of
interface default method.*/

interface TestInterface
{
    // abstract method
    public void abtmethod();

    // default method
    default void show()
    {
      System.out.println("Here Default Method Executed from the
interface.");
    }
}

class Practical3_7 implements TestInterface
{

    public void abtmethod()
    {
        System.out.println("This method's implimentation is in
class.");
    }

    public static void main(String args[])
    {
        Practical3_7 p = new Practical3_7();
        p.abtmethod();
```

```
        // default method
        p.show();
    }
}
```

Output:

```
PS C:\Users\HARSH>  & 'C:\Program Files\Java\jdk-1
This method's implimentation is in class.
Here Default Method Executed from the interface.
PS C:\Users\HARSH>
```