9.1

**package** demo2;

**public** **class** helloworld {

/\*\* Main method \*/

**public** **static** **void** main(String[] args) {

// Create a circle with radius 1

SimpleCircle circle1= **new** SimpleCircle();

System.***out***.println("The area of the circle of radius " + circle1.radius

+" is " + circle1.getArea());

// Create a circle with radius 25

SimpleCircle circle2= **new** SimpleCircle(25);

System.***out***.println("The area of the circle of radius " + circle2.radius

+ " is " + circle2.getArea());

// Create a circle with radius 125

SimpleCircle circle3 =**new** SimpleCircle(125);

System.***out***.println("The area of the circle of radius "+ circle3.radius

+ " is " + circle3.getArea());

// Modify circle radius

circle2.radius=100; // or circle2.setRadius(100)

System.***out***.println("The area of the circle of radius "

+ circle2.radius + " is " + circle2.getArea());

}

}

// Define the circle class with two constructors

**class** SimpleCircle {

**double** radius;

/\*\* Construct a circle with radius 1 \*/

SimpleCircle() {

radius=1;

}

/\*\* Construct a circle with a specified radius \*/

SimpleCircle(**double** newRadius) {

radius = newRadius;

}

/\*\* Return the area of this circle \*/

**double** getArea() {

**return** radius \*radius \* Math.***PI***;

}

**double** getPerimeter() {

**return** 2 \*radius \* Math.***PI***;

}

/\*\* Set a new radius for this circle \*/

**void** setRadius(**double** newRadius) {

radius = newRadius;

}

}

9.2

**package** demo2;

**public** **class** helloworld {

/\*\* Main method \*/

**public** **static** **void** main(String[] args) {

// Create a circle with radius 1

SimpleCircle circle1 = **new** SimpleCircle();

System.out.println("The area of the circle of radius " + circle1.radius

+ " is " + circle1.getArea());

// Create a circle with radius 25

SimpleCircle circle2 = **new** SimpleCircle(25);

System.out.println("The area of the circle of radius " + circle2.radius

+ " is " + circle2.getArea());

// Create a circle with radius 125

SimpleCircle circle3 = **new** SimpleCircle(125);

System.out.println("The area of the circle of radius " + circle3.radius

+ " is " + circle3.getArea());

// Modify circle radius

circle2.radius = 100; // or circle2.setRadius(100)

System.out.println("The area of the circle of radius "

+ circle2.radius + " is " + circle2.getArea());

**double** radius;

/\*\* Construct a circle with radius 1 \*/

SimpleCircle() {

radius = 1;

}

/\*\* Construct a circle with a specified radius \*/

SimpleCircle(**double** newRadius) {

radius = newRadius;

}

/\*\* Return the area of this circle \*/

**double** getArea() {

**return** radius \* radius \* Math.PI;

}

**double** getPerimeter() {

**return** 2 \* radius \* Math.PI;

}

/\*\* Set a new radius for this circle \*/

**void** setRadius(**double** newRadius) {

radius = newRadius;

}

}

}

9.3

**package** demo2;

**public** **class** helloworld {

**int** channel = 1; // Default channel is 1

**int** volumeLevel = 1; // Default volume level is 1

**boolean** on = **false**; // TV is off

**public** TV() {

}

**public** **void** turnOn() {

on = **true**;

}

**public** **void** turnOff() {

on= **false**;

}

**public** **void** setChannel(**int** newChannel) {

**if** (on && newChannel >= 1 && newChannel <= 120)

channel=newChannel ;

}

**public** **void** setVo1ume(**int** newVolumeLevel) {

**if** (on && newVolumeLevel >= 1 && newVolumeLevel <= 7)

volumeLevel = newVolumeLevel ;

}

**public** **void** channelUp() {

**if** (on && channel < 120)

channel ++;

}

**public** **void** channelDown() {

**if** (on && channel > 1)

channel--;

}

**public** **void** volumeUp() {

**if** (on && volumeLevel < 7)

volumeLevel ++;

}

**public** **void** volumeDown () {

**if** (on && volumeLevel > 1)

volumeLevel--;

}

}

9.4

**public** **static** **void** main(String[] args) {

TV tv1 = **new** TV();

tv1.turnOn();

tv1.setChannel (30);

tv1.setVolume(3);

TV tv2 = **new** TV();

tv2.turnOn ();

tv2 .channelUp();

tv2 .channelUp();

tv2.volumeUp();

System.***out***.println("tvl' s channel is " + tv1.channel

+ " and volume level is " + tv1.volumeLevel);

System.***out***.println("tv2's channel is " + tv2 .channel

+ " and volume level is " + tv2.volumeLevel);

}

9.5

**package** demo2;

**public** **class** helloworld {

**import** java.util.Scanner;

**import** javafx.geometry.Point2D;

**public** **class** TestPoint2D {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.***in***);

System.***out***.print("Enter pointl's x-t y-coordinates: ");

**double** xl = input.nextDouble();

**double** yl = input.nextDouble();

System.***out***.print("Enter point2' s x-t y-coordinates: ");

**double** x2 = input.nextDouble();

**double** y2 = input.nextDouble();

Point2D pi = **new** Point2D(x1, y1);

Point2D p2 = **new** Point2D(x2, y2);

System.***out***.println("p1 is " + p1.toString());

System.***out***.println("p2 is " + p2.toString());

System.***out***.println("The distance between pi and p2 is " +

p1.distance(p2));

}

}

}

9.6

public class CircleWithStaticMembers {

/\*\* The radius of the circle \*/

double radius;

/\*\* The number of objects created \*/

static int numberOfObjects=0;

/\*\* Construct a circle with radius 1 \*/

CircleWithStaticMembers(){

radius = 1;

numberOfObjects++;

}

/\*\* Construct a circle with a specified radius \*/

CircleWithStaticMembers(double newRadius) {

radius=newRadius;

numberOfObjects++ ;

}

/\*\* Return numberOfObjects \*/

static int getNumberOfObjectsC) {

return numberOfObjects;

}

/\*\* Return the area of this circle \*/

double getArea() {

return radius \* radius \* Math.PI;

}

}

9.7

**package** demo2;

**public** **class** helloworld {

**public** **static** **void** main(String[] args) {

System.out.println("Before creating objects");

System.out.println("The number of Circle objects is " +

CircleWithStaticMembers.numberOfObjects);

// Create c1

CircleWithStaticMembers c1 = **new** CircleWithStaticMembers();

// Display cl BEFORE c2 is created

System.out.println("\nAfter creating c1");

System.out.println("cl: radius (" + c1.radius +

") and number of Circle objects(" +

c1.numberOfObjects + ")");

// Create c2

CircleWithStaticMembers c2 = **new** CircleWithStaticMembers(5);

// Modify cl

cl.radius \* 9;

// Display cl and c2 AFTER c2 was created

System.out.println("\nAfter creating c2 and modifying cl");

System.out.println("c1: radius ( + c1.radius +

") and number of Circle objects(" +

c1.numberOfObjects + ")");

System.out.println("c2: radius (" + c2.radius +

") and number of Circle objects(" +

c2.number0f0bjects + ")");

}

}

9.8

public class CircleWithPrivateDataFields {

/\*The radius of the circle \*/

private double radius = 1;

/\*\* The number of objects created \*/

private static int numberOfObjects = 0;

/\*\* Construct a circle with radius =1 \*/

public CircleWithPrivateDataFields() {

numberOfObjects++;

}

/\*\* Construct a circle with a specified radius \*/

public CircleWithPrivateDataFields(double newRadius) {

radius\*

Radius;

numberOfObjects++;

}

/\*\* Return radius \*/

public double getRadiusO{

return radius;

}

/\*\* Set a new radius \*/

public void setRadius(double newRadius) {

radius = «(newRadius >= 0) ? newRadius : 0;

}

/\*\* Return numberOfObjects \*/

public static int getNumberOfObjectsO{

return numberOfObjects;

}

/\*\* Return the area of this circle \*/

public double getAreaO{

return radius \*

radius \* Math.PI;

}

}

9.9

public class TestCircleWithPrivateDataFields {

/\*\* Main method \*/

public static void main(String[] args) {

// Create a circle with radius 5.0

CircleWithPrivateDataFields myCircle =

new CircleWithPrivateDataFields(5.0);

System.out.println("The area of the circle of radius "

+ myCircle.getRadiusO + "is " + myCircle.getArea());

// Increase myCircle ’ s radius by 10%

myCircle.setRadius(myCircle.getRadiusO \* 1.1);

System.out.println("The area of the circle of radius "

+ myCircle.getRadiusO + " is " + myCircle.getArea());

System.out.println("The number of objects created is "

+ CircleWithPrivateDataFields.getNumberOfObjects())；

}

}

9.10

public class TestPassObject {

/\*\* Main method \*/

public static void main(String[] args) {

// Create a Circle object with radius 1

CircleWithPrivateDataFields myCircle = new CircleWithPrivateDataFields(1);

// Print areas for radius lt 2 V 3, 4, and S.

int n = 5;

printAreas(myCircle, n);

// See myCircle.radius and times

System.out.println("\n" + "Radius is " + myCircle.getRadius());

System, out.println("\n is " + n);

}

/\*\* Print a table of areas for radius \*/

public static void printAreas(

CircleWithPrivateDataFields c, int times) {

System.out.println("Radius \t\tArea");

while (times >= 1) {

System.out.println(c.getRadius() + "\t\t" + c.getArea());

c.setRadius(c.getRadius() + 1);

times--;

}

}

}

9.11

public class TotalArea {

/\* Main method \*/

public static void main(String[] args) {

// Declare circleArray

CircleWithPrivateDataFields[] circleArray;

// Create circleArray

circleArray = createCircleArray();

// Print circleArray and total areas of the circles

printCircleArray(circleArray);

}

- /\*\* Create an array of Circle objects \*/

public static CircleWithPrivateDataFields[] createCircleArray() {

CircleWithPrivateDataFields[] circleArray =

new CircleWithPrivateDataFields[5];

for (int i = 0;i < circleArray.length; i++) {

circleArray[i] =

new CircleWithPrivateDataFields(Math.random() \* 100);

}

// Return Circle array

return circleArray;

}

/\*\* Print an array of circles and their total area \*/

public static void printCircleArrayC

CircleWithPrivateDataFields[] circleArray){

System.out.printf("%-30s%-15s\n","Radius"，"Area");

for (int i = 0;i < circleArray.length;i++) {

System.out.printf("%-3Of%-15f\n", circleArray[i].getRadius(),

circleArray[i].getArea());

}

System.out.println("---------------------------------------------");

// Compute and display the result

System.out.printf("%-30s%-15f\n", "The total area of circles is" ,sum(circleArray));

}

/\*\*Add circle areas \*/

public static double sum(CircleWithPrivateDataFields[] circleArray) {

// Initialize sum

double sum \* 0;

// Add areas to sum

for (int i = 0; i < circleArray.length; i++)

sum +=circleArray[i].getArea()；

return sum;

}

}