代码5

GitHub地址：https://github.com/HuaZhouyang/Course\_JavaProgramming

———————————————————————————————————————

**Unit 9:**

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***AlternativeCircle.java:***

package Unit\_9;

/\*\*

\* Listing 9.2

\*/

public class AlternativeCircle {

/\*\*

\* Main method

\*/

public static void main(String[] args) {

// Create a circle with radius 1

AlternativeCircle circle1 = new AlternativeCircle();

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

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

// Create a circle with radius 25

AlternativeCircle circle2 = new AlternativeCircle(25);

System.out.println("The area of the circle of radius " + circle2.radius + " is " + circle2.getArea());

// Create a circle with radius 125

AlternativeCircle circle3 = new AlternativeCircle(125);

System.out.println("The area of the circle of radius " + circle3.radius + " is " + circle3.getArea());

// Modify circle radius

circle2.radius = 100;

System.out.println("The area of the circle of radius " + circle2.radius + " is " + circle2.getArea());

}

double radius;

/\*\*

\* Construct a circle with radius 1

\*/

AlternativeCircle() {

radius = 1;

}

/\*\*

\* Construct a circle with a specified radius

\*/

AlternativeCircle(double newRadius) {

radius = newRadius;

}

/\*\*

\* Return the area of this circle

\*/

double getArea() {

return radius \* radius \* Math.PI;

}

/\*\*

\* Return the perimeter of this circle

\*/

double getPerimeter() {

return 2 \* radius \* Math.PI;

}

/\*\*

\* Set a new radius for this circle

\*/

void setRadius(double newRadius) {

radius = newRadius;

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***CircleWithPrivateDataFields.java:***

package Unit\_9;

/\*\*

\* Listing 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 = newRadius;

numberOfObjects++;

}

/\*\*

\* Return radius

\*/

public double getRadius() {

return radius;

}

/\*\*

\* Set a new radius

\*/

public void setRadius(double newRadius) {

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

}

/\*\*

\* Return numberOfObjects

\*/

public static int getNumberOfObjects() {

return numberOfObjects;

}

/\*\*

\* Return the area of this circle

\*/

public double getArea() {

return radius \* radius \* Math.PI;

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***CircleWithStaticMembers.java:***

package Unit\_9;

/\*\*

\* Listing 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 getNumberOfObjects() {

return numberOfObjects;

}

/\*\*

\* Return the area of this circle

\*/

double getArea() {

return radius \* radius \* Math.PI;

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TestCircle.java:***

package Unit\_9;

/\*\*

\* Listing 9.1

\*/

public class TestCircle {

/\*\*

\* Main method

\*/

public static void main(String[] args) {

// Create a circle with radius 1

Circle circle1 = new Circle();

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

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

// Create a circle with radius 25

Circle circle2 = new Circle(25);

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

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

// Create a circle with radius 125

Circle circle3 = new Circle(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 Circle {

double radius;

/\*\*

\* Construct a circle with radius 1

\*/

Circle() {

radius = 1;

}

/\*\*

\* Construct a circle with a specified radius

\*/

Circle(double newRadius) {

radius = newRadius;

}

/\*\*

\* Return the area of this circle

\*/

double getArea() {

return radius \* radius \* Math.PI;

}

/\*\*

\* Return the perimeter of this circle

\*/

double getPerimeter() {

return 2 \* radius \* Math.PI;

}

/\*\*

\* Set a new radius for this circle

\*/

void setRadius(double newRadius) {

radius = newRadius;

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TestCircleWithPrivateDataFields.java:***

package Unit\_9;

/\*\*

\* Listing 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.getRadius() + " is " + myCircle.getArea());

// Increase myCircle's radius by 10%

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

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

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

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

+ CircleWithPrivateDataFields.getNumberOfObjects());

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TestCircleWithStaticMembers.java:***

package Unit\_9;

/\*\*

\* Listing 9.7

\*/

public class TestCircleWithStaticMembers {

/\*\*

\* Main method

\*/

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(); // Use the Circle class in Listing 9.6

// Display c1 BEFORE c2 is created

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

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

") and number of Circle objects (" +

c1.numberOfObjects + ")");

// Create c2

CircleWithStaticMembers c2 = new CircleWithStaticMembers(5);

// Modify c1

c1.radius = 9;

// Display c1 and c2 AFTER c2 was created

System.out.println("\nAfter creating c2 and modifying c1");

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.numberOfObjects + ")");

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TestPassObject.java:***

package Unit\_9;

/\*\*

\* Listing 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); // Use the Circle class in Listing 9.8

// Print areas for radius 1, 2, 3, 4, and 5.

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--;

}

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TestPoint2D.java:***

package Unit\_9;

import java.util.Scanner;

//import javafx.geometry.Point2D;

/\*\*

\* Listing 9.5

\*/

public class TestPoint2D {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter point1's x-, y–coordinates: ");

double x1 = input.nextDouble();

double y1 = input.nextDouble();

System.out.print("Enter point2's x-, y–coordinates: ");

double x2 = input.nextDouble();

double y2 = input.nextDouble();

Point2D p1 = 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 p1 and p2 is " +

p1.distance(p2));

System.out.println("The midpoint between p1 and p2 is " +

p1.midpoint(p2).toString());

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TestTV.java:***

package Unit\_9;

/\*\*

\* Listing 9.4

\*/

public class TestTV {

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("tv1'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);

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TotalArea.java:***

package Unit\_9;

/\*\*

\* Listing 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 printCircleArray(CircleWithPrivateDataFields[] circleArray) {

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

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

System.out.printf("%–30f%–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;

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***TV.java:***

package Unit\_9;

/\*\*

\* Listing 9.3

\*/

public class TV {

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 setVolume(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--;

}

}