代码6

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

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

## Unit 9:

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

IntersectingPoint.java:

package Unit\_9;

import java.util.Scanner;

/\*\*

\* 9.12

\*/

public class IntersectingPoint {

public static void main(String[] args) {

System.out.println("Please enter four points:");

/\*

y1 - (x1-x2)/(y1-y2)\*x1 + (x1-x2)/(y1-y2)\*x = y;

y1^2 - y1y2 - x1^2 + x1x2 + (x1-x2)x = (y1-y2)y;

(y1+x1)(y1-x1)

\*/

Scanner sc = new Scanner(System.in);

double x1 = sc.nextDouble(),y1 = sc.nextDouble();

double x2 = sc.nextDouble(),y2 = sc.nextDouble();

double x3 = sc.nextDouble(),y3 = sc.nextDouble();

double x4 = sc.nextDouble(),y4 = sc.nextDouble();

double k1 = (x1-x2)/(y1-y2), k2 = (x3-x4)/(y3-y4);

LinearEquation le = new LinearEquation(

k1, -1, y1-k1\*x1, k2, -1, y3-k2\*x3);

System.out.println(le.getX());

}

}

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

LinearEquation.java:

package Unit\_9;

import java.util.Scanner;

/\*\*

\* 9.11

\*/

public class LinearEquation {

private final double a, b, c, d, e, f, denominator;

public LinearEquation(double a, double b, double c, double d, double e, double f) {

this.a = a;

this.b = b;

this.c = c;

this.d = d;

this.e = e;

this.f = f;

denominator = a \* d - b \* c;

}

public double getA() {

return a;

}

public double getB() {

return b;

}

public double getC() {

return c;

}

public double getD() {

return d;

}

public double getE() {

return e;

}

public double getF() {

return f;

}

public boolean isSolvable() {

return denominator != 0;

}

public double getX() {

return (e \* d - b \* f) / denominator;

}

public double getY() {

return (a \* f - e \* c) / denominator;

}

}

class LinearEquationTest {

public static void main(String[] args) {

double a, b, c, d, e, f;

Scanner sc = new Scanner(System.in);

System.out.println("Please enter nums:");

a = sc.nextDouble();

b = sc.nextDouble();

c = sc.nextDouble();

d = sc.nextDouble();

e = sc.nextDouble();

f = sc.nextDouble();

LinearEquation le = new LinearEquation(a, b, c, d, e, f);

boolean solvable = le.isSolvable();

if (solvable) {

System.out.println(le.getX());

System.out.println(le.getY());

} else {

System.out.println("The equation has no solution.");

}

}

}

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

Location.java:

package Unit\_9;

import java.util.Scanner;

/\*\*

\* 9.13

\*/

public class Location {

public int row, column;

public double maxValue;

public Location(int row, int column, double maxValue) {

this.row = row;

this.column = column;

this.maxValue = maxValue;

}

public Location() {}

}

class LocationTest {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of rows and columns in the array: ");

int row = sc.nextInt(), column = sc.nextInt();

System.out.println("Enter the array:");

double[][] arr = new double[row][column];

for (int i = 0; i < row; i++) {

for (int j = 0; j < column; j++) {

arr[i][j] = sc.nextDouble();

}

}

Location res = locateLargest(arr);

System.out.println("The location of the largest element is " + res.maxValue + " at (" + res.row + ", " + res.column + ")");

}

public static Location locateLargest(double[][] a) {

Location loc = new Location();

loc.row = 0;

loc.column = 0;

loc.maxValue = a[0][0];

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

for (int j = 0; j < a[i].length; j++) {

if (loc.maxValue < a[i][j]) {

loc.row = i;

loc.column = j;

loc.maxValue = a[i][j];

}

}

}

return loc;

}

}

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

QuadraticEquation.java:

package Unit\_9;

import java.util.Scanner;

/\*\*

\* 9.10

\*/

public class QuadraticEquation {

private final long a, b, c, discriminant;

public QuadraticEquation(long a, long b, long c) {

this.a = a;

this.b = b;

this.c = c;

discriminant = b \* b - 4 \* a \* c;

}

public long getA() {

return a;

}

public long getB() {

return b;

}

public long getC() {

return c;

}

public long getDiscriminant() {

return discriminant;

}

public double getRoot1() {

if (discriminant < 0) return 0;

return (-b + Math.sqrt(discriminant) / 2 / a);

}

public double getRoot2() {

if (discriminant < 0) return 0;

return (-b - Math.sqrt(discriminant) / 2 / a);

}

}

class QuadraticEquationTest {

public static void main(String[] args) {

long a, b, c;

Scanner sc = new Scanner(System.in);

System.out.println("Please enter a, b, and c:");

a = sc.nextLong();

b = sc.nextLong();

c = sc.nextLong();

QuadraticEquation qe = new QuadraticEquation(a, b, c);

long disc = qe.getDiscriminant();

if (disc > 0) {

System.out.println(qe.getRoot1());

System.out.println(qe.getRoot2());

} else if (disc == 0) {

System.out.println(qe.getRoot1());

} else {

System.out.println("The equation has no roots.");

}

}

}

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

RegularPolygon.java:

package Unit\_9;

/\*\*

\* 9.9

\*/

public class RegularPolygon {

private int n = 3; // 多边形的变数

private double side = 1; // 边的长度

private double x = 0, y = 0; // 坐标

public RegularPolygon() {

}

public RegularPolygon(int n, double side) {

this.n = n;

this.side = side;

}

public RegularPolygon(int n, double side, double x, double y) {

this.n = n;

this.side = side;

this.x = x;

this.y = y;

}

public int getN() {

return n;

}

public void setN(int n) {

this.n = n;

}

public double getSide() {

return side;

}

public void setSide(double side) {

this.side = side;

}

public double getX() {

return x;

}

public void setX(double x) {

this.x = x;

}

public double getY() {

return y;

}

public void setY(double y) {

this.y = y;

}

public double getPerimeter() {

return n \* side;

}

public double getArea() {

return n \* side \* side / 4 / Math.tan(Math.PI / n);

}

}

class RegularPolygonTest {

public static void main(String[] args) {

// task 1

RegularPolygon rp1 = new RegularPolygon();

System.out.println(rp1.getPerimeter());

System.out.println(rp1.getArea());

// task 2

RegularPolygon rp2 = new RegularPolygon(6, 4);

System.out.println(rp2.getPerimeter());

System.out.println(rp2.getArea());

// task 3

RegularPolygon rp3 = new RegularPolygon(10, 4, 5.6, 7.8);

System.out.println(rp3.getPerimeter());

System.out.println(rp3.getArea());

}

}

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

StopWatch.java:

package Unit\_9;

import java.util.Date;

import java.util.Random;

/\*\*

\* 9.6

\*/

public class StopWatch {

private long startTime;

private long endTime;

public StopWatch() {

startTime = new Date().getTime();

}

public void start() {

startTime = new Date().getTime();

}

public void stop() {

endTime = new Date().getTime();

}

public long getElapsedTime() {

return (endTime - startTime);

}

}

class StopWatchTest {

public static void main(String[] args) {

int[] arr = new int[100000];

Random r = new Random();

for (int i = 0; i < 100000; i++) {

arr[i] = r.nextInt(1000);

}

StopWatch sw = new StopWatch();

selectSort(arr);

sw.stop();

System.out.println(sw.getElapsedTime());

}

private static void selectSort(int[] arr) {

for(int i = 0; i < arr.length-1; i++) {

int min = i;

for (int j = i + 1; j < arr.length; j++) {

if (arr[j] < arr[min]) {

min = j;

}

}

if (min != i) {

swap(arr, i, min);

}

}

}

private static void swap(int[] arr,int a,int b){

int temp = arr[a];

arr[a] = arr[b];

arr[b] = temp;

}

}

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

UsingDateClass.java:

package Unit\_9;

import java.util.Date;

/\*\*

\* 9.3

\*/

public class UsingDateClass {

public static void main(String[] args) {

Date date = new Date();

for (long i = 10000; i <= 100000000000L; i \*= 10) {

date.setTime(i);

System.out.println(date.toString());

}

}

}

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

UsingGregorianCalendarClass.java:

package Unit\_9;

import java.util.GregorianCalendar;

/\*\*

\* 9.5

\*/

public class UsingGregorianCalendarClass {

public static void main(String[] args) {

GregorianCalendar cal = new GregorianCalendar();

// task 1

System.out.println(cal.get(GregorianCalendar.YEAR));

System.out.println(cal.get(GregorianCalendar.MONTH) + 1);

System.out.println(cal.get(GregorianCalendar.DAY\_OF\_MONTH));

// task 2

cal.setTimeInMillis(1234567898765L);

System.out.println(cal.get(GregorianCalendar.YEAR));

System.out.println(cal.get(GregorianCalendar.MONTH) + 1);

System.out.println(cal.get(GregorianCalendar.DAY\_OF\_MONTH));

}

}

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

UsingRandomClass.java:

package Unit\_9;

import java.util.Random;

/\*\*

\* 9.4

\*/

public class UsingRandomClass {

public static void main(String[] args) {

Random rand = new Random(1000);

for (int i = 0; i < 50; i++) {

System.out.println(rand.nextInt(100));

}

}

}