1st program

import java.util.Scanner;

public class Pj1{

public static void main(String[] args) {

Scanner sc1 = new Scanner(System.in);

System.out.println("Enter size of matrix");

int n = sc1.nextInt();

int[][] matrix1 = new int[n][n];

int[][] matrix2 = new int[n][n];

int[][] matrix3 = new int[n][n];

System.out.println("Enter elements of Matrix1");

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

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

matrix1[i][j] = sc1.nextInt();

}

}

System.out.println("Enter elements of Matrix2");

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

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

matrix2[i][j] = sc1.nextInt();

}

}

for (int i = 0; n > i; i++) {

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

matrix3[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

System.out.println("Resultant Matrix:");

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

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

System.out.print(matrix3[i][j] + "\t");

}

System.out.println();

}

sc1.close();

}

}

2nd program

import java.util.Scanner;

class Stack {

int top;

int s1[] = new int[3];

int len1 = s1.length;

Stack() {

top = -1;

}

void push(int item) {

if (len1 - 1 == top) {

System.out.println("Stack overflow");

} else {

top = top + 1;

s1[top] = item;

}

}

void pop() {

if (top == -1) {

System.out.println("Stack underflow");

} else {

int item = s1[top];

System.out.println(item + " has been popped");

top = top - 1;

}

}

void display() {

if (top == -1) {

System.out.println("Stack is empty. No items to display");

} else {

System.out.println("Items in the Stack are ");

for (int i = top; i >= 0; i--) {

System.out.println(s1[i]);

}

}

}

}

public class Pj2 {

public static void main(String[] args) {

System.out.println("Stack Program");

int ch;

int item;

Stack stk = new Stack();

Scanner sc1 = new Scanner(System.in);

while (true) {

System.out.println("Enter 1, 2, 3, 4 for PUSH, POP, DISPLAY, EXIT");

ch = sc1.nextInt();

switch (ch) {

case 1:

System.out.println("Enter the item to be pushed");

item = sc1.nextInt();

stk.push(item);

break;

case 2:

stk.pop();

break;

case 3:

stk.display();

break;

case 4:

System.exit(0);

default:

System.out.println("Enter a valid choice");

}

}

}

}

3rd program

package pkg2;

class Employee {

private int id;

private String name;

private double salary;

public Employee(int id, String name, double salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

public void raiseSalary(double percent) {

if (percent > 0) {

double increment = salary \* (percent / 100);

salary = salary + increment;

System.out.println("New Salary: " + salary);

} else {

System.out.println("Invalid percentage. Salary remains unchanged.");

}

}

public void showEmployeeDetails() {

System.out.println("Employee ID: " + id);

System.out.println("Employee Name: " + name);

System.out.println("Employee Salary: " + salary);

}

}

public class Pj3 {

public static void main(String[] args) {

Employee Suresh = new Employee(1, "Suresh", 10000);

Suresh.showEmployeeDetails();

Suresh.raiseSalary(20);

Suresh.showEmployeeDetails();

}

}

4th program

package pkg2;

class MyPoint {

private int x;

private int y;

public MyPoint() {

this.x = 0;

this.y = 0;

}

public MyPoint(int x1, int y1) {

this.x = x1;

this.y = y1;

}

public void setXY(int x1, int y1) {

this.x = x1;

this.y = y1;

}

public int[] getXY() {

int[] coordinates = { x, y };

return coordinates;

}

public double distance(int x1, int y1) {

int xDiff = this.x - x1;

int yDiff = this.y - y1;

return Math.sqrt(xDiff \* xDiff + yDiff \* yDiff);

}

public double distance(MyPoint another) {

int xDiff = this.x - another.x;

int yDiff = this.y - another.y;

return Math.sqrt(xDiff \* xDiff + yDiff \* yDiff);

}

public double distance() {

return Math.sqrt(x \* x + y \* y);

}

@Override

public String toString() {

return "(" + x + ", " + y + ")";

}

}

public class Pj4 {

public static void main(String[] args) {

System.out.println("MyPoint");

MyPoint p1 = new MyPoint();

MyPoint p2 = new MyPoint(3, 4);

p1.setXY(1, 2);

int[] coordinates = p1.getXY();

System.out.println(coordinates[0] + " " + coordinates[1]);

System.out.println(p1.distance(5, 6));

System.out.println(p1.distance(p2));

System.out.println(p1.distance());

System.out.println(p1);

System.out.println(p2);

}

}

5th program

package pkg2;

class Shape{

public void draw() {

System.out.println("Drawing a shape");

}

public void erase() {

System.out.println("Erasing a shape");

}

}

class Circle extends Shape{

@Override

public void draw() {

System.out.println("Drawing a circle");

}

@Override

public void erase() {

System.out.println("Erasing a circle");

}

}

class Triangle extends Shape{

@Override

public void draw() {

System.out.println("Drawing a triangle");

}

@Override

public void erase() {

System.out.println("Erasing a triangle");

}

}

class Square extends Shape{

@Override

public void draw() {

System.out.println("Drawing a square");

}

@Override

public void erase() {

System.out.println("Erasing a square");

}

}

public class Pj5 {

public static void main(String[] args) {

System.out.println("Test Shape");

Shape[] shapes=new Shape[3];

shapes[0]=new Circle();

shapes[1]=new Triangle();

shapes[2]=new Square();

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

shapes[i].draw();

shapes[i].erase();

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

}

}

6th program

package pkg2;

abstract class Shape1{

abstract double calculateArea();

abstract double calculatePerimeter();

}

class Circle1 extends Shape1{

private double radius;

public Circle1(double radius) {

this.radius=radius;

}

@Override

double calculateArea() {

return Math.PI\*radius\*radius;

}

@Override

double calculatePerimeter() {

return 2\*Math.PI\*radius;

}

}

public class Pj6 {

public static void main(String[] args) {

System.out.println("Abstract Shapes");

Circle1 c1=new Circle1(5.0);

System.out.println(c1.calculateArea());

System.out.println(c1.calculatePerimeter());

}

}

7th program

package pkg2;

interface Resizable{

void resizeWidth(int width);

void resizeHeight(int height);

}

class Rectangle implements Resizable{

private int width;

private int height;

public Rectangle(int width,int height) {

this.width=width;

this.height=height;

}

public void resizeWidth(int width) {

this.width=width;

}

public void resizeHeight(int height) {

this.height=height;

}

public void display() {

System.out.println(width + "," + height);

}

}

public class Pj7 {

public static void main(String[] args) {

System.out.println("Resizable");

Rectangle r1=new Rectangle(5,10);

r1.display();

r1.resizeWidth(8);

r1.resizeHeight(12);

r1.display();

}

}

8th program

package pkg2;

class Outer{

void display() {

System.out.println("Outer class display method");

}

class Inner{

void display() {

System.out.println("Inner class display method");

}

}

}

public class Pj8 {

public static void main(String[] args) {

System.out.println("Outer Inner");

Outer o1 = new Outer();

o1.display();

Outer.Inner i1 = o1.new Inner();

i1.display();

}

}

9th program

package pkg2;

class DivByZeroException extends Exception{

public DivByZeroException(String message) {

super(message);

}

}

public class Pj9 {

public static void main(String[] args) {

System.out.println("Exception");

int nr=10;

int dr=0;

int result;

try {

if(dr==0) {

throw new DivByZeroException("division by zero not allowed");

}

}

catch(DivByZeroException e1) {

System.out.println(e1.getMessage());

}

catch(ArithmeticException e2) {

System.out.println(e2.getMessage());

}

finally {

System.out.println("Finally block executed");

}

}

}

10th program

package pkg2;

class MyPoint {

private int x;

private int y;

public MyPoint() {

this.x = 0;

this.y = 0;

}

public MyPoint(int x1, int y1) {

this.x = x1;

this.y = y1;

}

public void setXY(int x1, int y1) {

this.x = x1;

this.y = y1;

}

public int[] getXY() {

int[] coordinates = { x, y };

return coordinates;

}

public double distance(int x1, int y1) {

int xDiff = this.x - x1;

int yDiff = this.y - y1;

return Math.sqrt(xDiff \* xDiff + yDiff \* yDiff);

}

public double distance(MyPoint another) {

int xDiff = this.x - another.x;

int yDiff = this.y - another.y;

return Math.sqrt(xDiff \* xDiff + yDiff \* yDiff);

}

public double distance() {

return Math.sqrt(x \* x + y \* y);

}

@Override

public String toString() {

return "(" + x + ", " + y + ")";

}

}

public class Pj4 {

public static void main(String[] args) {

System.out.println("MyPoint");

MyPoint p1 = new MyPoint();

MyPoint p2 = new MyPoint(3, 4);

p1.setXY(1, 2);

int[] coordinates = p1.getXY();

System.out.println(coordinates[0] + " " + coordinates[1]);

System.out.println(p1.distance(5, 6));

System.out.println(p1.distance(p2));

System.out.println(p1.distance());

System.out.println(p1);

System.out.println(p2);

}

}

11th program

package pkg2;

class MyRunnable implements Runnable{

@Override

public void run() {

try {

Thread.sleep(5);

System.out.println("Thread " +Thread.currentThread().getId() + "running");

}

catch(InterruptedException e) {

e.printStackTrace();

}

}

}

public class Pj11 {

public static void main(String[] args) {

System.out.println("Runnable");

MyRunnable mr1=new MyRunnable();

Thread t1=new Thread(mr1);

Thread t2=new Thread(mr1);

Thread t3=new Thread(mr1);

t1.start();

t2.start();

t3.start();

}

}

12th program

package pkg2;

class MyThread extends Thread {

public MyThread(String name) {

super(name);

// Remove the start() here.

}

public void run() {

for (int i = 1; i <= 5; i++) {

System.out.println("Thread " + i);

try {

Thread.sleep(1);

} catch (InterruptedException e) {

System.out.println("Thread interrupted");

}

}

}

}

public class Pj12 {

public static void main(String[] args) {

System.out.println("Thread");

MyThread t1 = new MyThread("ChildThread");

t1.start();

}

}