Program No: 1

Date: 12-January-2024

SADDLE POINTS OF AN INTEGER MATRIX

Problem

Program to print the saddle points of an integer matrix

Class Design

Class Matrix

- Properties
  - matrix, numRows, numColumns
- Methods
  - populateFromUserInput ()
    - insert matrix values from the user
  - Display()
    - Display the elements of matrix
  - identifySaddlePoints()
    - To find the saddle point of a matrix.

Program

```
import java.util.Scanner;


class Matrix {
    private int[][] matrix;
    private int numRows;
    private int numColumns;

    Matrix(int numRows, int numColumns) {
        this.numRows = numRows;
```

```
        this.numColumns = numColumns;
        this.matrix = new int[numRows][numColumns];
    }

    void populateFromUserInput(Scanner inputScanner) {
        System.out.println("Enter the matrix elements row-
wise: ");
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numColumns; j++) {
                matrix[i][j] = inputScanner.nextInt();
            }
        }
    }

    void display() {
        System.out.println("The matrix is: ");
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numColumns; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }

    void identifySaddlePoints() {
        boolean saddleFound = false;
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numColumns; j++) {
                int element = matrix[i][j];
```

```
                    boolean isSaddle = true;

                    // Check if element is largest in its row
                    for (int k = 0; k < numColumns; k++) {
                        if (matrix[i][k] < element) {
                            isSaddle = false;
                            break;
                        }
                    }

                    // Check if element is smallest in its column
                    if (isSaddle) {
                        for (int k = 0; k < numRows; k++) {
                            if (matrix[k][j] > element) {
                                isSaddle = false;
                                break;
                            }
                        }
                    }

                    if (isSaddle) {
                        System.out.println("Saddle point found
at (" + (i + 1) + ", " + (j + 1) + ") with value " + element);
                        saddleFound = true;
                    }
                }
            }

            if (!saddleFound) {
```

```
                System.out.println("No saddle points found.");
            }
        }
    }

    class Main {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the number of rows:");
            int rows = scanner.nextInt();
            System.out.println("Enter the number of columns:");
            int columns = scanner.nextInt();
            Matrix matrix1 = new Matrix(rows, columns);
            matrix1.populateFromUserInput(scanner);
            matrix1.display();
            matrix1.identifySaddlePoints();
        }
    }
```

Output

```
Enter the number of rows:
3
Enter the number of columns:
3
Enter the matrix elements row-wise:
9
8
7
6
5
4
3
```

```
2
1
The matrix is:
9 8 7
6 5 4
3 2 1
Saddle point found at (1, 3) with value 7
```

Program No: 2

Date: 12-January-2024

NUMBER OF OCCURRENCES OF EACH CHARACTER OF A STRING

Problem

Program to print the number of occurrences of each character of a string.

Class Design

Class Characters

- Properties
  - MAX_CHARS
- Methods
  - countChars()
    - count the Total number of characters in a string.

Program

```
import java.util.Scanner;

class CharCounter {
    static final int MAX_CHARS = 256;

    static void countChars(String str) {
        int[] count = new int[MAX_CHARS];
        int len = str.length();

        for (int i = 0; i < len; i++) {
            count[str.charAt(i)]++;
        }

        char[] ch = new char[len];
        for (int i = 0; i < len; i++) {
            ch[i] = str.charAt(i);
```

```
            int find = 0;
            for (int j = 0; j <= i; j++) {
                if (str.charAt(i) == ch[j]) {
                    find++;
                }
            }
            if (find == 1) {
                System.out.println("total   number   of   "   +
str.charAt(i) + ": " + count[str.charAt(i)]);
            }
        }
    }
}

class Main {
    public static void main(String[] args) {
        CharCounter c = new CharCounter();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string:");
        String str = sc.nextLine();
        c.countChars(str);
    }
}
```

Output

```
Enter the string:hello_world
Total Number of h: 1
Total Number of e: 1
Total Number of l: 3
Total Number of o: 2
Total Number of _: 1
Total Number of w: 1
Total Number of r: 1
Total Number of d: 1
```

Program No: 3

Date: 12-January-2024

ROOTS OF A QUADRATIC EQUATION

Problem

Program to find the roots of a quadratic equation. The coefficients and the constant are given.

Class Design

Class Quadratic

- Properties
  - Value1,value2,value3
- Methods
  - Main()
    - Finding the Quadratic roots in Main Methode itself.

Program

```
import java.util.Scanner;

class QuadraticEquation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the 3 Values of Quadratic :");
        double value1 = scanner.nextDouble();
        double value2 = scanner.nextDouble();
        double value3 = scanner.nextDouble();

        double discriminant = value2 * value2 - 4 * value1 *
value3;

        if (discriminant > 0) {
```

```
            double root1 = (-value2 + Math.sqrt(discriminant)) /
(2 * value1);
            double root2 = (-value2 - Math.sqrt(discriminant)) /
(2 * value1);
            System.out.println("The roots are: " + root1 + " and
" + root2);
        } else if (discriminant == 0) {
            double root = -value2 / (2 * value1);
            System.out.println("The root is: " + root);
        } else {
            System.out.println("The equation has no real roots.");
        }
    }
}
```

Output 1

```
Enter the 3 Values of Quadratic :
2
-5
-3
The roots are: 3.0 and -0.5
```

Output 2

```
Enter the 3 Values of Quadratic :
3
4
5
The equation has no real roots.
```

Program No: 4

Date: 12-January-2024

ADD TWO REAL NUMBERS

Problem

Program to add two real numbers. Read the input as command line arguments

Class Design

Class Addition

- Properties
  - Value1,value2
- Methods
  - sum()
    - Add the values and print the result

Program

```
import java.util.Scanner;

class Addition {
    int value1, value2;

    Addition(int x, int y) {
        value1 = x;
        value2 = y;
    }

    void sum() {
        int result = value1 + value2;
        System.out.println("Sum = " + result);
    }
}

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

```
        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);

        Addition result1 = new Addition(num1, num2);
        result1.sum();
    }
}
```

Output

```
java Insert 3 5

Sum = 8
```

Program No: 5

Date: 19-January-2024

AREA OF RECTANGLE AND CIRCLE

Problem

Menu driven program to calculate the area of a rectangle from its length and width and that of a circle from its radius

Class Design

Class Rectangle

- Properties
  - length, width
- Methods
  - setDimensions()
    - Insert the values of rectangle.
  - calculateArea()
    - To calculate the area of a rectangle.

Class Circle

- Properties
  - radius
- Methods
  - setRadius
    - To insert the radius from the user.
  - calculateArea()
    - To calculate the area of a rectangle.

Program

```
import java.util.Scanner;

class Rectangle {
    double length, width;
```

```
    public void setDimensions() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the length and the width:");
        length = scanner.nextDouble();
        width = scanner.nextDouble();
    }

    public double calculateArea() {
        return length * width;
    }
}

class Circle {
    double radius;

    void setRadius() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the radius:");
        radius = scanner.nextDouble();
    }

    double calculateArea() {
        return Math.PI * radius * radius;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Rectangle rectangle = new Rectangle();
        Circle circle = new Circle();
        int choice;

        while (true) {
            System.out.println("Enter your choice:\n1. Calculate
area of rectangle\n2. Calculate area of circle\n3. Exit");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    rectangle.setDimensions();
                    System.out.println("Area of the rectangle is
" + rectangle.calculateArea());
                    break;
                case 2:
                    circle.setRadius();
                    System.out.println("The area of the circle is
" + circle.calculateArea());
```

```
                    break;
                case 3:
                    System.exit(0);
                default:
                    System.out.println("Invalid Input!");
            }
        }
    }
}
```

Output 1

```
Enter your choice:
1. Calculate area of rectangle
2. Calculate area of circle
3. Exit
1    //entered choice
Enter the length and the width:
2
3
Area of the rectangle is 6.0
Enter your choice:
1. Calculate area of rectangle
2. Calculate area of circle
3. Exit
2
Enter the radius:
3
The area of the circle is 28.274333882308138
```

Output 2

```
Enter your choice:
1. Calculate area of rectangle
2. Calculate area of circle
3. Exit
4
Invalid Input !!!!
```

Program No: 6

Date: 19-January-2024

AREA OF A CIRCLE GIVEN THE COORDINATES

Problem

Program to calculate the area of a circle given the coordinates of its center and that of a point on its boundary.

Class Design

Class Circle

- Properties
  - x1,x2,y1,y2
- Methods
  - Insertvalue()
    - To insert the values from user.
  - area()
    - To find the Distance(area) of a circle from the coordinate values of center and boundary

Program

```
import java.util.Scanner;

class Circle {
    double x1, y1, x2, y2;

    void insertvalue() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the coordinates of the two points:");
        x1 = scanner.nextDouble();
        y1 = scanner.nextDouble();
        x2 = scanner.nextDouble();
```

```
        y2 = scanner.nextDouble();
    }

    double area() {
       double r= Math.sqrt((x1-y1)*(x1-y1)+(x2-y2)*(x2-y2));
       return Math.PI*r*r;
    }
}

public class Main {
    public static void main(String[] args) {
        Circle circle = new Circle();
        circle.insertvalue ();
        System.out.print("area of circle = ");
        System.out.println(circle.area());
    }
}
```

Output

```
Enter the coordinates of the two points:
2
4
5
6
Area of circle = 15.707963267948967
```

Program No: 7

Date: 12-January-2024

STUDENT DATABASE

Problem

Program to store student data which include name, register number and marks obtained in 4 subjects and to print the results. The result should contain name, register number, marks, passed/failed per subject, passed/failed in the whole examination and total marks if passed in all the subjects. The maximum total per subject is 50 and 25 is required for a pass.

Class Design

Class Student

- Properties
  - name,registerNumber,marks[4],isSubjectPassed,result.
- Methods
  - Marks()
    - To insert the mark
  - Result()
    - To check whether the student is passed or not.
  - Display()
    - Print the result

Program

```
import java.util.Scanner;

class Student {
    String name;
    int registerNumber;
    int[] marks = new int[4];
    boolean[] isSubjectPassed = new boolean[4];
    boolean result;
    int totalMarks;
```

```
    Student(String name, int registerNumber) {
        this.name = name;
        this.registerNumber = registerNumber;
    }

    void Marks() {
        Scanner scanner = new Scanner(System.in);
        for (int i = 0; i < 4; i++) {
            System.out.println("Enter marks for subject " + (i +
1) + ": ");
            marks[i] = scanner.nextInt();
            isSubjectPassed[i] = marks[i] >= 25;
        }
    }

    void Result() {
        int total = 0;
        result = true;
        for (int i = 0; i < 4; i++) {
            total += marks[i];
            if (!isSubjectPassed[i]) {
                result = false;
            }
        }
        totalMarks = total;
    }

    void display() {
        System.out.println("Student Name: " + name);
        System.out.println("Register Number: " + registerNumber);
        System.out.println("Marks per subject:");
        for (int i = 0; i < 4; i++) {
            System.out.println("Subject " + (i + 1) + ": " +
marks[i] + " (" +
                    (isSubjectPassed[i] ? "Passed" : "Failed") +
")");
        }
        System.out.println("Result: " + (result ? "Passed" :
"Failed"));
        if (result) {
            System.out.println("Total Marks: " + totalMarks);
```

```
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the student details:");
        System.out.print("Name: ");
        String name = scanner.nextLine();
        System.out.print("Register Number: ");
        int registerNumber = scanner.nextInt();
        Student student = new Student(name, registerNumber);
        student.Marks();
        student.Result();
        student.display();
    }
}
```

Output

```
Enter the student details:
Name: Sneha
Register Number: 9745916
Enter marks for subject 1:
85
Enter marks for subject 2:
70
Enter marks for subject 3:
45
Enter marks for subject 4:
60
Student Name: Sneha
Register Number: 9745916
Marks per subject:
Subject 1: 85 (Passed)
Subject 2: 70 (Passed)
Subject 3: 45 (Passed)
Subject 4: 60 (Passed)
Result: Passed
Total Marks: 260
```