// Khanmohammadi, Amir Hosein, 991646689

// 2023-01-19

public class Square {

private double side;

private static int numberOfSquares;

public Square() {

// this.side = 1;

// numberOfSquares++;

this(1.0);

}

public Square(double side) {

setSide(side);

numberOfSquares++;

}

public double getSide() {

return side;

}

public void setSide(double side) {

side = (side > 0) ? side : 1.0;

this.side = side;

}

public double getArea() {

return side \* side;

}

public static int getNumberOfSquares() {

return numberOfSquares;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// Khanmohammadi, Amir Hosein, 991646689

// 2023-01-19

import java.util.Random;

class TestSquareArray {

public static void main(String[] args) {

Random rand = new Random();

Square[] squareArray = createSquareArray(rand);

printSquareArray(squareArray);

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

System.out.println("Total area of the squares: " + sum(squareArray));

System.out.println("Total number of squares: " + Square.getNumberOfSquares());

}

public static Square[] createSquareArray(Random rand) {

Square[] squareArray = new Square[5];

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

double side = rand.nextInt(100) + 1;

squareArray[i] = new Square(side);

}

return squareArray;

}

public static void printSquareArray(Square[] squareArray) {

System.out.println("Side\tArea");

for (Square s : squareArray) {

System.out.println(s.getSide() + "\t" + s.getArea());

}

}

public static double sum(Square[] squareArray) {

double total = 0;

for (Square s : squareArray) {

total += s.getArea();

}

return total;

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Output

Side Area

57.0 3249.0

51.0 2601.0

90.0 8100.0

89.0 7921.0

30.0 900.0

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

Total area of the squares: 22771.0

Total number of squares: 5

Text

Description automatically generated

UML

|  |
| --- |
| Square |
| -side: double  -numberOfSquares: int |
| +Square()  +Square(side: double)  +getSide(): double  +setSide(side: double)  +getArea(): double  +getNumberOfSquares(): int |