

Data Structure and Algorithm Practicum

Array of Objects



Name

Dicha Zelianivan Arkana

NIM

2241720002

Class

li

Department

Information Technology

Study Program

D4 Informatics Engineering

Contents

1.2	Create, insert, and display Array of Object	2
1.2.1	Steps	2
1.2.2	Result	3
1.2.3	Questions	3
1.3	Get input in array of Objects using Loops	3
1.3.1	Questions	3
1.4	Mathematical operation in array of object's attribute	4
1.4.1	Questions	4
1.5	Practice	7

1.2 Create, insert, and display Array of Object

1.2.1 Steps

1. Create a new project with name ArrayOfObjects. Create the package with name 'week3'
2. Create a **Rectangle** class:

```
public class Rectangle {  
    public int length;  
    public int width;  
}
```

3. In main method in ArrayOfObjects class, create an array Rectangle and its length is 3

```
public class ArrayOfObjects {  
    public static void main(String[] args) {  
        Rectangle[] rectangleArray = new Rectangle[3];  
    }  
}
```

4. Then insert values for each of the object's attributes

```
rectangleArray[0] = new Rectangle();  
rectangleArray[0].length = 110;  
rectangleArray[0].width = 30;
```

```
rectangleArray[1] = new Rectangle();  
rectangleArray[1].length = 80;  
rectangleArray[1].width = 40;
```

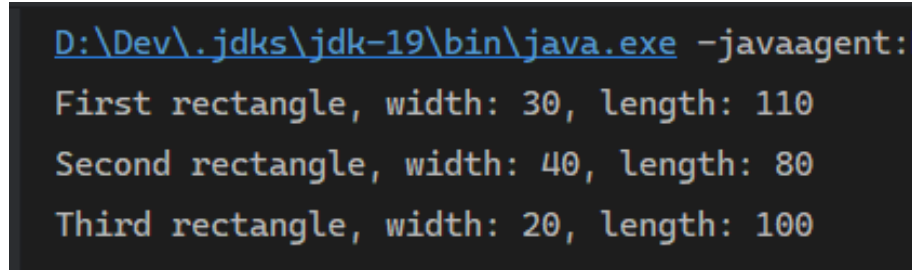
```
rectangleArray[2] = new Rectangle();  
rectangleArray[2].length = 100;  
rectangleArray[2].width = 20;
```

5. Print all the attributes object from ppArray as follows

```
System.out.println("First rectangle, width: " + rectangleArray[0].width + ", length: "  
System.out.println("Second rectangle, width: " + rectangleArray[1].width + ", length: "  
System.out.println("Third rectangle, width: " + rectangleArray[2].width + ", length: "
```

1.2.2 Result

Compile the code and see the result if it matches with the following image.



```
D:\Dev\jdk-19\bin\java.exe -javaagent:
First rectangle, width: 30, length: 110
Second rectangle, width: 40, length: 80
Third rectangle, width: 20, length: 100
```

Figure 1: The result

1.2.3 Questions

1. An object used for the array is not required to have both attribute *and* method. It needs to have either one of them, or both, but it depends on the use case. If we do need the attribute then we add it, if we don't need it then we don't have to add it.

In this case specifically, we need the attributes for the rectangle because an object without attribute or method would be useless.

2. Every class in java, if we don't define the constructor, we can use the parameterless constructor adding it explicitly.
3. We initialise an array of `Rectangle` with the length of 3
4. We set the first element of the array `rectangleArray` with an instance of `Rectangle`, and we set the `length` and `width` attribute with a number
5. It is not required, but it's better if we differentiate them to better modularise our code and apply the Single Responsibility Principle which means that a class should only concern about one thing and one thing only.

1.3 Get input in array of Objects using Loops

1.3.1 Questions

1. The array of objects can be implemented on 2D array. We would write it like how we would write it normally with primitive datatype.

-
2. Example of 2D array using objects:

```
Point[] [] coordinates = new Point[10][2];
```

3. Because we haven't instantiated the array but we try to change its attribute value.
4. `import java.util.Scanner;`

```
public class ArrayOfObjects {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Insert the length of the array: ");
        int length = sc.nextInt();

        Rectangle[] rectangleArray = new Rectangle[length];

        for (int i = 0; i < length; i++) {
            rectangleArray[i] = new Rectangle();
            System.out.println("Rectangle " + i);

            System.out.print("Input length: ");
            rectangleArray[i].length = sc.nextInt();

            System.out.print("Input width: ");
            rectangleArray[i].width = sc.nextInt();
        }

        for (int i = 0; i < length; i++) {
            System.out.println("Rectangle " + i);
            System.out.println("width: " + rectangleArray[0].width + ", length: " + re
        }
    }
}
```

5. Yes, if we do that then the previous value will get overridden

1.4 Mathematical operation in array of object's attribute

1.4.1 Questions

1. Yes, we can have multiple constructors in a single class as long as they have different signatures. It's called "overloading"

2. Create a Triangle class

```
public class Triangle {  
    public int base;  
    public int height;  
}
```

3.

```
public class Triangle {  
    public int base;  
    public int height;
```

```
    public Triangle(int a, int t) {  
        this.base = a;  
        this.height = t;  
    }
```

```
    public double calculateArea() {  
        return (double)(base/2)*height;  
    }
```

```
    public double calculatePerimeter() {  
        double side = Math.sqrt(Math.pow(base, 2) + Math.pow(height, 2));  
        return side + base + height;  
    }  
}
```

4.

```
public class TriangleMain {  
    public static void main(String[] args) {  
        Triangle[] triangles = new Triangle[4];  
  
        triangles[0] = new Triangle(10, 4);  
        triangles[1] = new Triangle(20, 10);  
        triangles[2] = new Triangle(15, 6);  
        triangles[3] = new Triangle(25, 10);  
    }  
}
```

```
5. public class TriangleMain {
    public static void main(String[] args) {
        Triangle[] triangles = new Triangle[4];

        triangles[0] = new Triangle(10, 4);
        triangles[1] = new Triangle(20, 10);
        triangles[2] = new Triangle(15, 6);
        triangles[3] = new Triangle(25, 10);

        for (int i = 0; i < triangles.length; i++) {
            Triangle triangle = triangles[i];
            System.out.printf(
                "%d. Area: %.2f | Perimeter: %.2f\n",
                i + 1,
                triangle.calculateArea(),
                triangle.calculatePerimeter());
        }
    }
}
```

```
D:\Dev\jdk\jdk-19\bin\java.exe -javaagent:D:\Dev\JetBrains\Toolb
1. Area: 20.00 | Perimeter: 24.77
2. Area: 100.00 | Perimeter: 52.36
3. Area: 42.00 | Perimeter: 37.16
4. Area: 120.00 | Perimeter: 61.93
```

1.5 Practice

1. Cube.java

```
public class Cube {  
    public int vertice;  
  
    public double calculateVolume() {  
        return Math.pow(vertice, 3);  
    }  
  
    public double calculateSurfaceArea() {  
        return vertice * vertice * 12;  
    }  
}
```

CubeMain.java

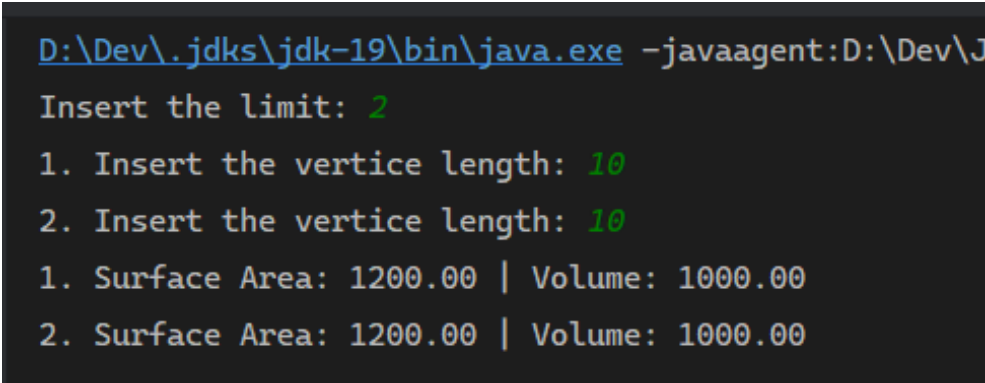
```
import java.util.Scanner;

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

        System.out.print("Insert the limit: ");
        int limit = input.nextInt();
        Cube[] cubes = new Cube[limit];

        for (int i = 0; i < cubes.length; i++) {
            cubes[i] = new Cube();
            System.out.printf("%d. Insert the vertice length: ", i + 1);
            cubes[i].vertice = input.nextInt();
        }

        for (int i = 0; i < cubes.length; i++) {
            System.out.printf(
                "%d. Surface Area: %.2f | Volume: %.2f\n",
                i + 1,
                cubes[i].calculateSurfaceArea(),
                cubes[i].calculateVolume()
            );
        }
    }
}
```



```
D:\Dev\jdk-19\bin\java.exe -javaagent:D:\Dev\J
Insert the limit: 2
1. Insert the vertice length: 10
2. Insert the vertice length: 10
1. Surface Area: 1200.00 | Volume: 1000.00
2. Surface Area: 1200.00 | Volume: 1000.00
```

```
2. import java.util.Scanner;

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

        System.out.print("How many lands? ");
        int limit = input.nextInt();

        Land[] lands = new Land[limit];
        for (int i = 0; i < lands.length; i++) {
            lands[i] = new Land();
            System.out.printf("Land %d\n", i + 1);
            System.out.print("Length: ");
            lands[i].length = input.nextInt();
            System.out.print("Width: ");
            lands[i].width = input.nextInt();
        }

        for (int i = 0; i < lands.length; i++) {
            System.out.printf(
                "Land Area %d: %.2f\n",
                i + 1, lands[i].calculateArea()
            );
        }
    }
}
```

```
3. import java.util.Scanner;

public class LandMain {
    static void printWidestArea(Land[] lands) {
        int widestIndex = 0;
        for (int i = 1; i < lands.length; i++) {
            if (lands[i].calculateArea() > lands[widestIndex].calculateArea()) {
                widestIndex = i;
            }
        }
        System.out.printf("The widest land is Land %d", widestIndex);
    }

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

        System.out.print("How many lands? ");
    }
}
```

```

        int limit = input.nextInt();

        Land[] lands = new Land[limit];
        for (int i = 0; i < lands.length; i++) {
            lands[i] = new Land();
            System.out.printf("Land %d\n", i + 1);
            System.out.print("Length: ");
            lands[i].length = input.nextInt();
            System.out.print("Width: ");
            lands[i].width = input.nextInt();
        }

        for (int i = 0; i < lands.length; i++) {
            System.out.printf("Land Area %d: %.2f\n", i + 1, lands[i].calculateArea())
        }

        System.out.println();
        printWidestArea(lands);
    }
}

```

```

4. public class Student {
    public String name;
    public String nim;
    public char gender;
    public double ipk;
}

import java.util.Scanner;

public class StudentMain {
    static void printAverageScore(Student[] students) {
        double total = 0;
        for (Student student : students) {
            total += student.ipk;
        }
        System.out.printf("Average IPK of all students : %.2f", total / students.length)
    }

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

        System.out.println("Insert the limit: ");
        int limit = input.nextInt();
    }
}

```

```

Student[] students = new Student[limit];

for (int i = 0; i < students.length; i++) {
    System.out.print("Insert name: ");
    students[i].name = input.next();
    System.out.print("Insert nim: ");
    students[i].nim = input.next();
    System.out.print("Insert gender: ");
    students[i].gender = input.next().charAt(0);
    System.out.print("Insert ipk: ");
    students[i].ipk = input.nextDouble();
}

for (int i = 0; i < students.length; i++) {
    System.out.printf(
        "Student %d\nname : %s\nnim : %s\ngender : %c\nIPK score : %.2f\n",
        i + 1,
        students[i].name,
        students[i].nim,
        students[i].gender,
        students[i].ipk
    );
}
}
}

```

5. `import java.util.Scanner;`

```

public class StudentMain {
    static void printAverageScore(Student[] students) {
        double total = 0;
        for (Student student : students) {
            total += student.ipk;
        }
        System.out.printf("Average IPK of all students : %.2f", total / students.length);
    }

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

        System.out.println("Insert the limit: ");
        int limit = input.nextInt();
        Student[] students = new Student[limit];
    }
}

```

```

    for (int i = 0; i < students.length; i++) {
        System.out.print("Insert name: ");
        students[i].name = input.next();
        System.out.print("Insert nim: ");
        students[i].nim = input.next();
        System.out.print("Insert gender: ");
        students[i].gender = input.next().charAt(0);
        System.out.print("Insert ipk: ");
        students[i].ipk = input.nextDouble();
    }

    for (int i = 0; i < students.length; i++) {
        System.out.printf(
            "Student %d\nname : %s\nnim : %s\ngender : %c\nIPK score : %.2f\n",
            i + 1,
            students[i].name,
            students[i].nim,
            students[i].gender,
            students[i].ipk
        );
    }
}

```