JOBSHEET 3

Array of Objects

Nama : Yuma Akhunza Kausar Putra

NIM : 2341720259

1.2 Create, insert, and display Array of Object

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

        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;

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

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

QUESTIONS !

1. Based on part 1.2, does the class that are going to be used as an array of object must have attributes and methods? Please explain!
2. Does class Rectangle have constructor? If not, why we instantiate the object as follows?

```java
rectangleArray[1] = new Rectangle();
```

3. What's the meaning of this line of code?

```
Rectangle[] rectangleArray = new Rectangle[3];
```

4. What's the meaning of these lines of code?

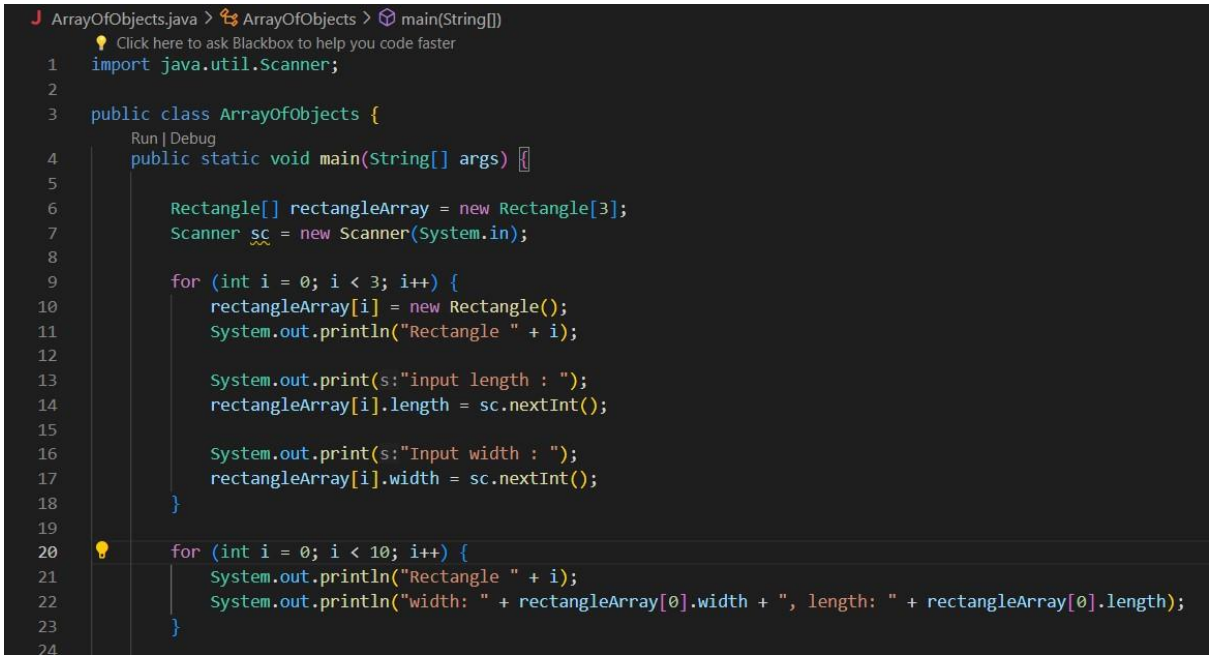
```
rectangleArray[1] = new Rectangle();
rectangleArray[1].length = 80;
rectangleArray[1].width = 40;
```

5. Why ArrayOfObject class and Rectangle class should be separated?

ANSWER !

1. The Rectangle class has attributes (length and width) but no explicit methods. In general, there is no requirement to have methods in a class that will be created as an array of objects. The importance of having a method depends on your program's specific needs and design.
2. does not have an explicit constructor declaration. If a class does not have a defined constructor, Java automatically provides a default constructor without parameters.
3. Those line is the way we declaring Array of object, the format for the declaration is '**ClassName arrayName[ ] = new Classname[length];**
   At those line, we create an array that could be used to store 3 objects from the class Rectangle.
4. – 'rectangleArray[1] = new Rectangle();': In the first line, it creates a new object of the 'Rectangle' class and assigns it to index 1 of the 'rectangleArray'. In other words, the newly created Rectangle object is stored at position 1 (index 1) in the 'rectangleArray'.
   - 'rectangleArray[1].length = 80;': After creating the 'Rectangle' object at index 1, this line accesses the 'length' attribute of that object and sets it to the value 80. In other words, the length of the 'Rectangle' object at index 1 is set to 80.
   - 'rectangleArray[1].width = 40;': In the last line, this code accesses the 'width' attribute of the 'Rectangle' object at index 1 and sets it to the value 40. Consequently, the width of the 'Rectangle' object at index 1 is set to 40.
5. Because in java, only one public class is allowed per file, and the file name must match the name of the public class.

1.3 Input data into Array of objects using loops



```
ArrayOfObjects.java > ArrayOfObjects > main(String[])
Click here to ask Blackbox to help you code faster
import java.util.Scanner;

public class ArrayOfObjects {
    Run | Debug
    public static void main(String[] args) {

        Rectangle[] rectangleArray = new Rectangle[3];
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 3; i++) {
            rectangleArray[i] = new Rectangle();
            System.out.println("Rectangle " + i);

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

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

        for (int i = 0; i < 10; i++) {
            System.out.println("Rectangle " + i);
            System.out.println("width: " + rectangleArray[0].width + ", length: " + rectangleArray[0].length);
        }
```

QUESTIONS

1. Does array of object can be implemented on 2D array?
2. If yes, then please give an example. Otherwise, please explain.
3. There is a Square class that has an attribute side with integer as its data type. There will be an error when we run this code, why?

```
Square[] squareArray = new Square[100];
squareArray[5].side = 20;
```

4. Modify the code on part 1.3 so that the length of the array will be defined from user input.
5. Can we duplicate the instantiation process in array of objects? For example, we assign the object in ppArray[i] and ppArray[0], the instantiation process of ppArray[0] will be done twice. What's the effect of this?

ANSWER!

1. Yes, array of objects can be implemented in two-dimensional arrays in Java. This implementation allows you to create an array containing objects of a class and organize them in a two-dimensional structure.

2.

```java
import java.util.Scanner;

public class Array2D21 {
    Run | Debug
    public static void main(String[] args) {

        Rectangle2d[][] rectangleGrid = new Rectangle2d[3][3];

        rectangleGrid[0][0] = new Rectangle2d(length:10, width:20);
        rectangleGrid[0][1] = new Rectangle2d(length:15, width:25);
        rectangleGrid[0][2] = new Rectangle2d(length:20, width:30);
        rectangleGrid[1][0] = new Rectangle2d(length:25, width:35);
        rectangleGrid[1][1] = new Rectangle2d(length:30, width:40);
        rectangleGrid[1][2] = new Rectangle2d(length:35, width:45);
        rectangleGrid[2][0] = new Rectangle2d(length:40, width:50);
        rectangleGrid[2][1] = new Rectangle2d(length:45, width:55);
        rectangleGrid[2][2] = new Rectangle2d(length:50, width:60);

        System.out.println(
                "Rectangle at (0, 0): length=" + rectangleGrid[0][0].length + ", width=" + rectangleGrid[0][0].width);
        System.out.println(
                "Rectangle at (1, 2): length=" + rectangleGrid[1][2].length + ", width=" + rectangleGrid[1][2].width);
        System.out.println(
                "Rectangle at (2, 1): length=" + rectangleGrid[2][1].length + ", width=" + rectangleGrid[2][1].width);
    }
}

```

```
Rectangle at (0, 0): length=10, width=20
Rectangle at (1, 2): length=35, width=45
Rectangle at (2, 1): length=45, width=55
PS D:\Algoritma-Jobsheet>
```

3. The code will be error, because the variable square not instantiation at the code. 4. create a new class for squares, so that the array of squares can be active.

square.java > square > side

Click here to ask Blackbox to help you code faster

```java
public class square {
    int side;
}

```

```java
square[]  sqrArray = new square[100];
sqrArray[5]= new square();
sqrArray[5].side = 20;
```

6. Yes, you can duplicate the instantiation process for an array of objects, similar to what you did for Rectangle, for an array of another class, like square. Here's how you can modify your existing code to include the instantiation process for an array of square objects.

1.4 Mathematical operation in array of object's attribute

Blocks.java > Blocks > countVolume()

Click here to ask Blackbox to help you code faster

```java
public class Blocks {
    public int width, length, height;

    public Blocks(int p, int l, int t) {
        length = p;
        width = l;
        height = t;
    }

    public int countVolume() {
        return length * width * height;
    }
}
```

ArrayBlocks.java > ArrayBlocks

Click here to ask Blackbox to help you code faster

```java
public class ArrayBlocks {

    Run | Debug
    public static void main(String[] args) {
        Blocks[] blArray = new Blocks[3];

        blArray[0] = new Blocks(p:100, l:30, t:12);
        blArray[1] = new Blocks(p:120, l:40, t:15);
        blArray[2] = new Blocks(p:210, l:50, t:25);

        for (int i = 0; i < 3; i++) {
            System.out.println("Volume blocks - " + i + " : " + blArray[i].countVolume());
        }

    }
}
```

QUESTIONS

1. Can we have more than one constructor in one class? Please explain.
2. Create a Triangle class as follows.

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

   Add another constructor in this class that has parameter int a, int t. These represents its base and height.
3. Add method countArea() and countPerimeter() in class Triangle

4. In main function, instantiate array of Triangle objects. Assign the attributes values as follows:
   0th trArray base: 10, height: 4
   1st trArray base: 20, height: 10
   2nd trArray base: 15, height: 6
   3rd trArray base: 25, height: 10
5. Display the result of area and perimeter for each triangle by calling the method countArea() and countPerimeter()

ANSWER!

1. Yes, a class in Java can have more than one constructor, which is known as constructor overloading. Constructor overloading allows a class to have multiple constructors with different parameter lists.
2.

```
1   public class Triangle {
2       public int base, height;
3
4       public Triangle(int a, int t) {
5           base = a;
6           height = t;
        }
7       }
8
9   }
10
```

3.

```
7       }
8       double countArea() {
9           return 0.5 * base * height;
10      }
11      double countPerimeter() {
12          return 2 * (base + height);
13          }
14      }
15
```

4.

```java
public class TriangleMain {
    Run | Debug
    public static void main(String[] args) {
        Triangle[] t = new Triangle[4];

        for (int i = 0; i < t.length; i++) {
            t[i] = new Triangle(a:0, t:0);
        }

        t[0].base = 10;
        t[0].height = 4;

        t[1].base = 20;
        t[1].height = 10;

        t[2].base = 15;
        t[2].height = 6;

        t[3].base = 25;
        t[3].height = 10;

        for (int i = 0; i < t.length; i++) {
            System.out.println("Area of triangle " + i + " is: " + t[i].countArea());
            System.out.println("Perimeter of triangle " + i + " is: " + t[i].countPerimeter());
        }
    }
}
```

5.

```
Area of triangle 0 is: 20.0
Perimeter of triangle 0 is: 28.0
Area of triangle 1 is: 100.0
Perimeter of triangle 1 is: 60.0
Area of triangle 2 is: 45.0
Perimeter of triangle 2 is: 42.0
Area of triangle 3 is: 125.0
Perimeter of triangle 3 is: 70.0
PS D:\Algoritma-Jobsheet>
```

ASSIGMENT

1. https://github.com/akhunzakp/semester-2/blob/main/jobsheet3/Sphere.java
2. https://github.com/akhunzakp/semester-2/blob/main/jobsheet3/SquarePyramid.java
3. https://github.com/akhunzakp/semester-2/blob/main/jobsheet3/CalculateMain.java
4. https://github.com/akhunzakp/semester-2/blob/main/jobsheet3/Murid.java
5. https://github.com/akhunzakp/semester-2/blob/main/jobsheet3/Uni.java