

Nama: Muhammad Iqbal Alexandre Saputra
NIM: 11231056
Kelas: PBO B

Source Code

No	com/tugas2/UpgradedCalculator.java
1.	<code>package com.tugas2;</code>
2.	<code>import java.util.Scanner;</code>
3.	
4.	<code>public class UpgradedCalculator {</code>
5.	<code> public static void main(String[] args) {</code>
6.	<code> Scanner scanner = new Scanner(System.in);</code>
7.	<code> double result = readNumber(scanner);</code>
8.	
9.	<code> while (true) {</code>
10.	<code> showMenu();</code>
11.	<code> int operation = scanner.nextInt();</code>
12.	<code> double secondNumber;</code>
13.	<code> switch (operation) {</code>
14.	<code> case 1:</code>
15.	<code> secondNumber = readNumber(scanner);</code>
16.	<code> result = add(result, secondNumber);</code>
17.	<code> break;</code>
18.	
19.	<code> case 2:</code>
20.	<code> secondNumber = readNumber(scanner);</code>
21.	<code> result = subtract(result, secondNumber);</code>
22.	<code> break;</code>
23.	<code> case 3:</code>
24.	<code> secondNumber = readNumber(scanner);</code>
25.	<code> result = multiply(result, secondNumber);</code>
26.	<code> break;</code>
27.	<code> case 4:</code>
28.	<code> secondNumber = readNumber(scanner);</code>
29.	<code> result = divide(result, secondNumber);</code>
30.	<code> break;</code>
31.	<code> default:</code>
32.	<code> return;</code>
33.	<code> }</code>
34.	<code> System.out.println("Current Total: " + result);</code>
35.	<code> }</code>
36.	<code> }</code>
37.	
38.	<code> private static void showMenu() {</code>
39.	<code> System.out.println("Choose the operation:");</code>
40.	<code> System.out.println("1. Add");</code>
41.	<code> System.out.println("2. Subtract");</code>
42.	<code> System.out.println("3. Multiply");</code>
43.	<code> System.out.println("4. Divide");</code>
44.	<code> System.out.print("Your choice (input anything else to stop the program): ");</code>
45.	<code> }</code>
46.	
47.	<code> private static double readNumber(Scanner scanner) {</code>
48.	<code> System.out.print("Enter a number: ");</code>
49.	<code> return scanner.nextDouble();</code>
50.	<code> }</code>
51.	
52.	

```
53. private static double add(double firstNumber, double secondNumber) {
54.     return firstNumber + secondNumber;
55. }
56.
57. private static double subtract(double firstNumber, double secondNumber) {
58.     return firstNumber - secondNumber;
59. }
60.
61. private static double multiply(double firstNumber, double secondNumber) {
62.     return firstNumber * secondNumber;
63. }
64.
65. private static double divide(double firstNumber, double secondNumber) {
66.     if (secondNumber != 0) {
67.         return firstNumber / secondNumber;
68.     } else {
69.         System.out.println(firstNumber + " Cannot be divided by zero");
70.         return firstNumber;
71.     }
72. }
73. }
74. }
```

Source Code

No	com/tugas2/Matrix.java
1.	<code>package com.tugas2;</code>
2.	
3.	<code>import java.util.Scanner;</code>
4.	
5.	<code>public class Matrix {</code>
6.	<code> public static void main(String[] args) {</code>
7.	<code> Scanner scanner = new Scanner(System.in);</code>
8.	<code> System.out.print("Enter amount of row for both matrix: ");</code>
9.	<code> int rowAmount = scanner.nextInt();</code>
10.	<code> System.out.print("Enter amount of column for both matrix: ");</code>
11.	<code> int colAmount = scanner.nextInt();</code>
12.	
13.	<code> int matrix1[][] = new int[rowAmount][colAmount];</code>
14.	<code> int matrix2[][] = new int[rowAmount][colAmount];</code>
15.	<code> int result[][] = new int[rowAmount][colAmount];</code>
16.	
17.	<code> System.out.println("input matrix 1 continuously: ");</code>
18.	<code> matrix1 = inputMatrix(scanner, rowAmount, colAmount);</code>
19.	<code> System.out.println("input matrix 2 continuously: ");</code>
20.	<code> matrix2 = inputMatrix(scanner, rowAmount, colAmount);</code>
21.	
22.	
23.	<code> System.out.println("Matrix 1:");</code>
24.	<code> printMatrix(matrix1);</code>
25.	<code> System.out.println("Matrix 2:");</code>
26.	<code> printMatrix(matrix2);</code>
27.	
28.	<code> result = addMatrix(matrix1, matrix2);</code>
29.	<code> System.out.println("Result:");</code>
30.	<code> printMatrix(result);</code>
31.	
32.	<code> scanner.close();</code>
33.	<code> }</code>
34.	
35.	<code> private static void printMatrix(int[][] matrix) {</code>
36.	<code> for (int i = 0; i < matrix.length; i++) {</code>
37.	<code> for (int j = 0; j < matrix[i].length; j++) {</code>
38.	<code> System.out.print(matrix[i][j] + " ");</code>
39.	<code> }</code>
40.	<code> System.out.println(" ");</code>
41.	<code> }</code>
42.	<code> System.out.println();</code>
43.	<code> }</code>
44.	
45.	<code> private static int[][] addMatrix(int[][] matrix1, int[][] matrix2) {</code>
46.	<code> int result[][] = new int[matrix1.length][matrix1[0].length];</code>
47.	
48.	<code> for (int i = 0; i < matrix1.length; i++) {</code>
49.	<code> for (int j = 0; j < matrix1[i].length; j++) {</code>
50.	<code> result[i][j] = matrix1[i][j] + matrix2[i][j];</code>
51.	<code> }</code>
52.	<code> }</code>
53.	
54.	<code> return result;</code>
55.	
56.	<code> }</code>
57.	

```
58.     private static int[][] inputMatrix(Scanner scanner, int rowAmount, int
59.     colAmount) {
60.         int[][] matrix = new int[rowAmount][colAmount];
61.
62.         for (int i = 0; i < matrix.length; i++) {
63.             for (int j = 0; j < matrix.length; j++) {
64.                 matrix[i][j] = scanner.nextInt();
65.             }
66.         }
67.
68.         return matrix;
69.     }
70. }
71. }
```

Screenshot UpgradedCalculator.java

```
Enter a number: 10
Choose the operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Your choice (input anything else to stop the program): 1
Enter a number: 10
Current Total: 20.0
Choose the operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Your choice (input anything else to stop the program): |
```

```
Current Total: 20.0
Choose the operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Your choice (input anything else to stop the program): 2m
Enter a number: 10
Current Total: 10.0
Choose the operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Your choice (input anything else to stop the program): |
```

```

Enter a number: 5 or choice: 1
Choose the operation: side length: 25
1. Add The volume of the cube is 15625.0 mm
2. Subtract
3. Multiply Choose one of 3 dimensional shape below
4. Divide 1. Cube
Your choice (input anything else to stop the program): 3
Enter a number: 4 Cylinder
Current Total: 20.0 choice: 2
Choose the operation: 2. the length units:
1. Add 1. Millimeter
2. Subtract 2. Centimeter
3. Multiply 3. Meter
4. Divide Your choice: 2
Your choice (input anything else to stop the program): |

```

```

Current Total: 20.0
Your choice (input anything else to stop the program): 4
Choose the operation:
1. Add
2. Subtract Input two integers:
3. Multiply First number:20
4. Divide Second number:2
Your choice (input anything else to stop the program): 4
Enter a number: 5 add
Current Total: 4.0 Subtract
Choose the operation: multiply
1. Add divide
2. Subtract Your choice: 4
3. Multiply The result of dividing 20.0 and 2.0 is 10.0
4. Divide
Screenshot Volume.java
Your choice (input anything else to stop the program): |

```

Screenshot Matrix.java

```
Enter amount of row for both matrix: 3
Enter amount of column for both matrix: 3
input matrix 1 continously:
1
2
3
4
5
6
7
8
9

input matrix 2 continously:
1
2
3
4
5
6
7
8
9

Matrix 1:
1 2 3
4 5 6
7 8 9

Matrix 2:
1 2 3
4 5 6
7 8 9

Result:
2 4 6
8 10 12
14 16 18
```

Penjelasan

UpgradedCalculator.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas UpgradedCalculator berada.
2	Mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
4-5	Membuat class UpgradedCalculator dan metode main.
6	Membuat objek Scanner bernama input untuk mengambil input dari pengguna melalui konsol.
7	Memanggil metode readNumber untuk mendapatkan angka pertama dari pengguna dan menyimpannya dalam variabel result
9-36	Blok while menerima input operasi yang ingin pengguna lakukan dan input angka kedua untuk dihitung, dan menampilkan hasilnya secara berulang-ulang hingga pengguna memilih untuk keluar dengan cara memilih input operasi selain 1 hingga 4 dan memberhentikan while loop.
38-46	Function showMenu berfungsi menampilkan menu pilihan operasi.
47-50	Function readNumber berfungsi menerima input angka dari pengguna.
53-71	Metode metode add, subtract, multiply, divide yang menerima 2 parameter yaitu angka pertama dan angka kedua lalu mengembalikan hasilnya. Metode metode ini dipanggil setelah pengguna menginput angka kedua pada blok while.

Matrix.java

No.	Deskripsi
1	Baris ini menyatakan tempat kelas Matrix berada.
3	Mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
5-6	Membuat class UpgradedCalculator dan metode main.
7	Membuat objek Scanner bernama input untuk mengambil input dari pengguna melalui konsol.
8-11	Menerima input banyak kolom dan baris untuk kedua matriks dari pengguna
13-15	Membuat variabel array matriks1, matriks2, dan result.
18-21	Menerima input dari pengguna untuk mengisi matriks 1 dan 2.
23-26	Mencetak matriks1 dan matriks2.
28-30	Menambahkan matriks1 dan matriks2 dengan function addMatrix dan disimpan ke

	variabel result lalu dicetak.
32	Menutup scanner yang sudah tidak digunakan untuk mencegah <i>resource leak</i>
35-43	Function printMatrix berfungsi mengiterasi setiap elemen elemen matriks (matrix[i][j]) dan dicetak menggunakan System.out.print(), kemudian diikuti dengan spasi. Setelah selesai mencetak seluruh elemen dalam satu baris, dipanggil System.out.println(“ “) untuk mencetak baris baru, sehingga matriks tercetak dalam format yang terstruktur.
45-56	Function addMatrix memiliki dua parameter bertipe array dua dimensi (int[][] matrix1 dan int[][] matrix2), yang merupakan dua matriks yang akan dijumlahkan. Pada awal method, dibuat sebuah array dua dimensi baru dengan nama result, yang memiliki ukuran yang sama dengan matriks pertama (matrix1). Di dalam perulangan, setiap elemen matriks pada posisi yang sama dijumlahkan dan hasilnya disimpan dalam matriks result.
58-69	Function inputMatrix berfungsi menerima input dari pengguna untuk mengisi matriks. Pada awal method, dibuat sebuah array dua dimensi dengan nama matrix, yang memiliki ukuran sesuai dengan parameter. Di dalam perulangan, setiap elemen matriks diisi dengan nilai yang dibaca dari input pengguna menggunakan scanner.nextInt().