POLBAN

Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

Pertemuan 2 Java Fundamental 2
3.1 – 3.6

D3 Kelas 2A/2B

Dosen Pengampu :

Zulkifli Arsyad, Yadi Adithya, Beri N

Topics

- Input Output
- Big Integer
- Control Flow
- Array

Input

```
import java.util.*;


Scanner in = new Scanner(System.in);
System.out.print("What is your name? ");
String name = in.nextLine();
System.out.print("How old are you? ");
int age = in.nextInt();
```

Big Integer

- BigInteger class is used for mathematical operation which involves very big integer calculations that are outside the limit of all available primitive data types.

```
if(x>=-128 && x<=127)System.out.println("* byte");
if(x>=-32768 && x<=32767)System.out.println("* short");
if(x>=-2147483648 && x<=2147483647)
System.out.println("* int"); if(x>=(-(Math.pow(2,63))) && x<=(Math.pow(2,63)-
1))System.out.println("* long")
```

Big Integer

- BigInteger class is used for mathematical operation which involves very big integer calculations that are outside the limit of all available primitive data types.
- For example factorial of 100 contains 158 digits in it so we can't store it in any primitive data type available.

```java
// Java program to find large factorials using BigInteger
import java.math.BigInteger;
import java.util.Scanner;

public class Example
{
    // Returns Factorial of N
    static BigInteger factorial(int N)
    {
        // Initialize result
        BigInteger f = new BigInteger("1"); // Or BigInteger.ONE

        // Multiply f with 2, 3, ...N
        for (int i = 2; i <= N; i++)
            f = f.multiply(BigInteger.valueOf(i));

        return f;
    }

    // Driver method
    public static void main(String args[]) throws Exception
    {
        int N = 20;
        System.out.println(factorial(N));
    }
}
```

Output:

```
2432902008176640000
```

Array

- In java, you can declare arrays in two ways. Those two ways of declaring arrays in java are,

Data_Type[] Variable_Name;

AND

Data_Type Variable_Name[];

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        int[] arrayOfInts;    //Declaring an array of ints

        double arrayOfDoubles[];   //Declaring an array of doubles

        char[] arrayOfChars;     //Declaring an array of characters

        boolean arrayOfBooleans[];   //Declaring an array of booleans
    }
}
```

Instantiating An Array Object

- You can instantiate or create an array object using new operator. The syntax for instantiating arrays in java is,

Variable_Name = new Data_Type[ArraySize];

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        int[] arrayOfInts;      //Declaring an array of int

        arrayOfInts = new int[10];   //Instiantiating an array of int using new operator
     }
}
```

Initializing Array Elements :

- The syntax for initializing array elements is,

Variable_Name[index] = Value;

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        int[] arrayOfInts;      //Declaring an array of int

        arrayOfInts = new int[10];   //Instantiating an array of int using new operator

        arrayOfInts[2] = 12;         //Initializing 3rd element

        arrayOfInts[5] = 56;         //Initializing 6th element
    }
}
```

Combining declaration, instantiate and initialization into one statement :

- You can combine declaration, instantiate and initialization in one statement like below,
-

Data_Type[] Variable_Name = new Data_Type[]
{Value0, Value1, Value2, Value3 ….. };

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        //Declaring, instantiating and Initializing an array in one statement

        double[] arrayOfDoubles = new double[] {12.56, 45.87, 14.85};

        //This is also ok.

        int[] arrayOfInts = {12, 21, 0, 5, 7};
    }
}
```

```
public class ArraysInJava
{
    public static void main(String[] args)
    {
        //Declaring and Initializing an array in one statement

        int[] arrayOfInts = {12, 21, 0, 5, 7};

        System.out.println(arrayOfInts[0]);   //accessing 1st element

        System.out.println(arrayOfInts[3]);   //accessing 4th element

        System.out.println(arrayOfInts[2]);   //accessing 3rd element
    }
}
```

Multidimensional

```
public class MultiDimensionalArraysInJava
{
    public static void main(String[] args)
    {
        //One Dimensional Arrays
        int[] fisrtArray = {1, 2, 3};

        int[] secondArray = {4, 5, 6};

        int[] thirdArray = {7, 8, 9};

        int[] fourthArray = {10, 11, 12};

        int[] fifthArray = {13, 14, 15};

        int[] sixthArray = {16, 17, 18};

        int[] seventhArray = {19, 20, 21};

        int[] eighthArray = {22, 23, 24};

        int[] ninthArray = {25, 26, 27};

        //Two Dimensional Arrays
        int[][] twoDimensionalArray1 = {fisrtArray, secondArray, thirdArray};

        int[][] twoDimensionalArray2 = {fourthArray, fifthArray, sixthArray};

        int[][] twoDimensionalArray3 = {seventhArray, eighthArray, ninthArray};

        //Three Dimensional Array
        int[][][] threeDimensionalArray = {twoDimensionalArray1, twoDimensionalArray2, twoDimensionalArray3};
```