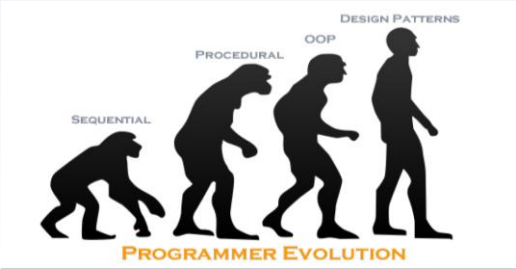
STT-NF

Pemrograman Berorientasi Objek: Variabel, Operasi, dan Tipe Data

Tifani Nabarian, S.Kom., M.T.I.
tifany.nabarian@dosen.nurulfikri.ac.id

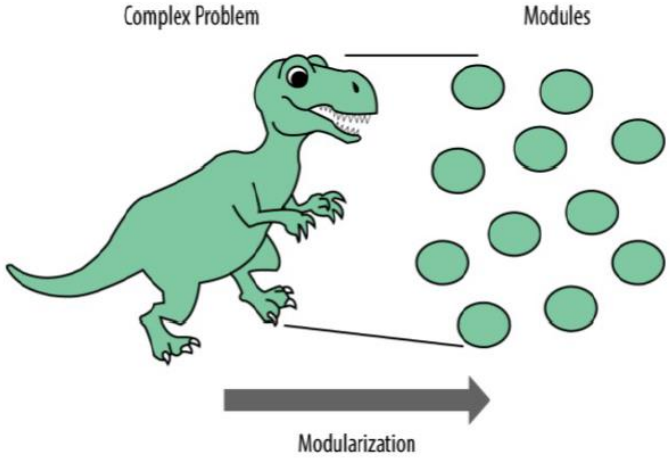
Review Chapter 1

Review Chapter 1




```php
<?php

$firstNumber=20;
$secondNumber=40;
$total= $firstNumber + $secondNumber;
echo $total;

?>
```

```php
<?php

function addEmUp($first,$second)
{
    $total=$first + $second;
    echo $total;
}
addEmUp(20,40);

?>
```

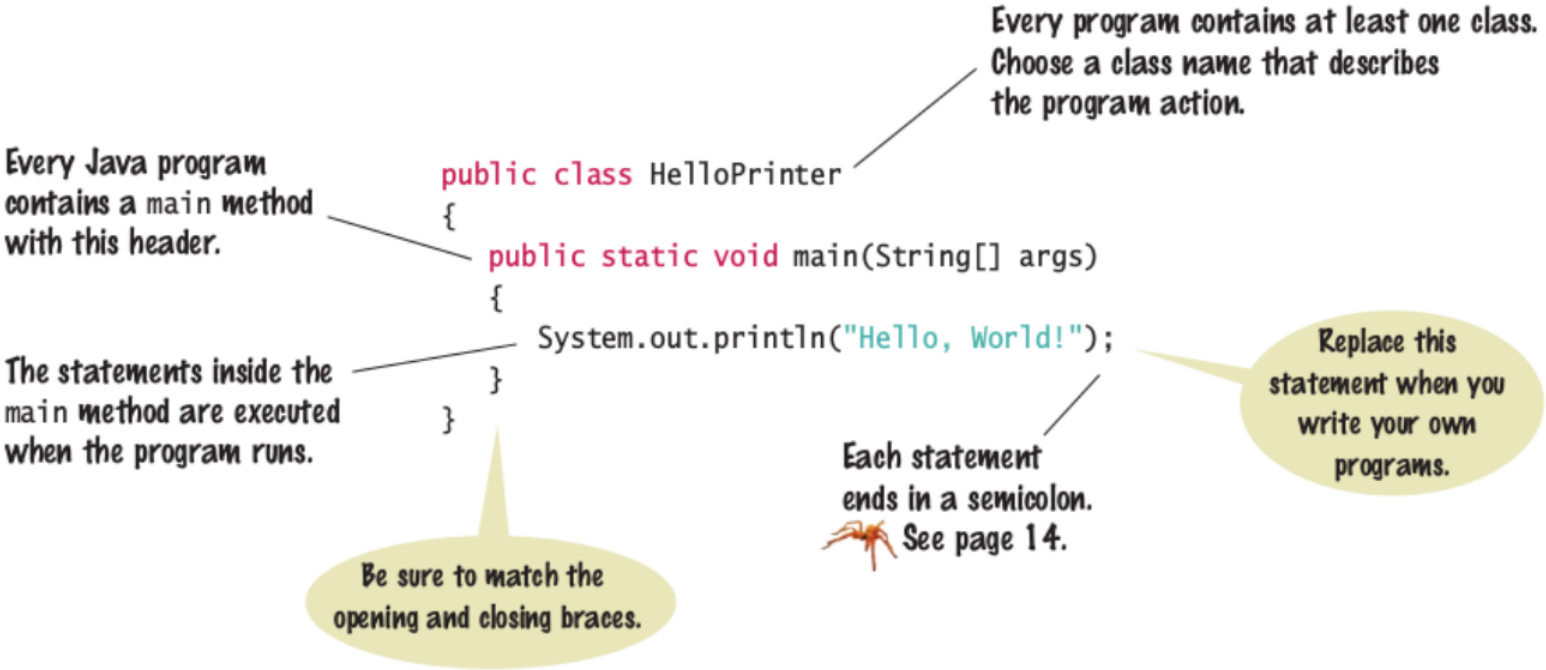


- Object Oriented

Customer, money, account

Variabel, Operasi, dan Tipe Data

Review- Java Structure



What is variable?

- In programming, **a variable** is a value that can change, depending on conditions or on information passed to the program.
- Typically, a program consists of instructions that tell the computer what to do and data that the program uses when it is running.

What is variable?

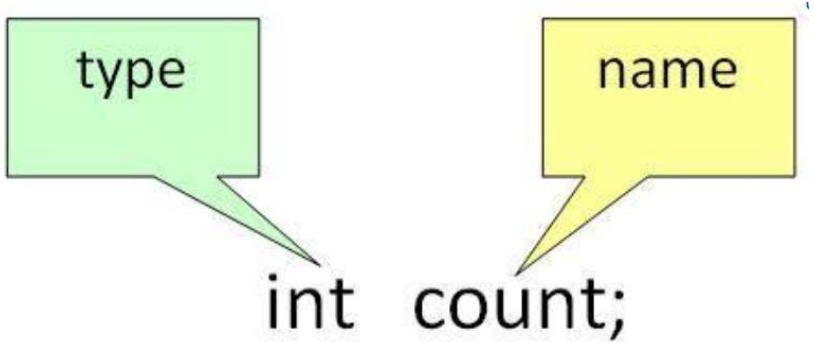
https://www.geeksforgeeks.org/variables-in-java

- Variable in Java is a data container that saves the data values during Java program execution. Every variable is assigned a data type that designates the type and quantity of value it can hold. A variable is a memory location name for the data.
- A variable is a name given to a memory location. It is the basic unit of storage in a program.
- The value stored in a variable can be changed during program execution.

What is variable?

https://www.geeksforgeeks.org/variables-in-java

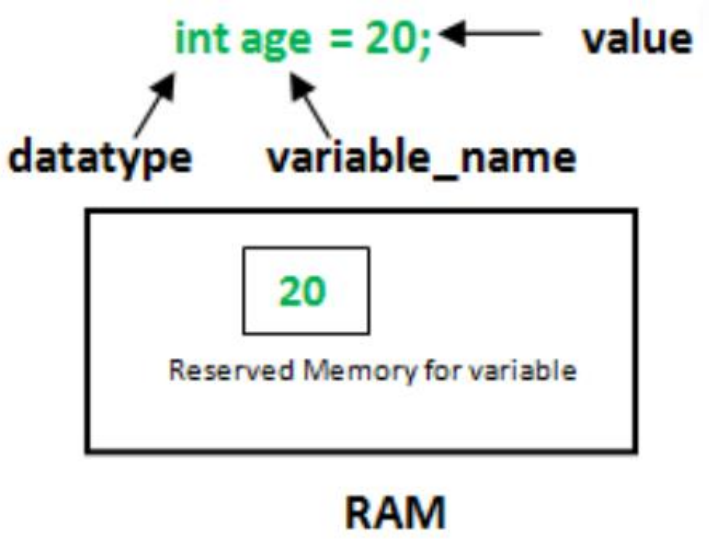
- A variable is only a name given to a memory location. All the operations done on the variable affect that memory location.
- In Java, all variables must be declared before use



How to initialize variables?

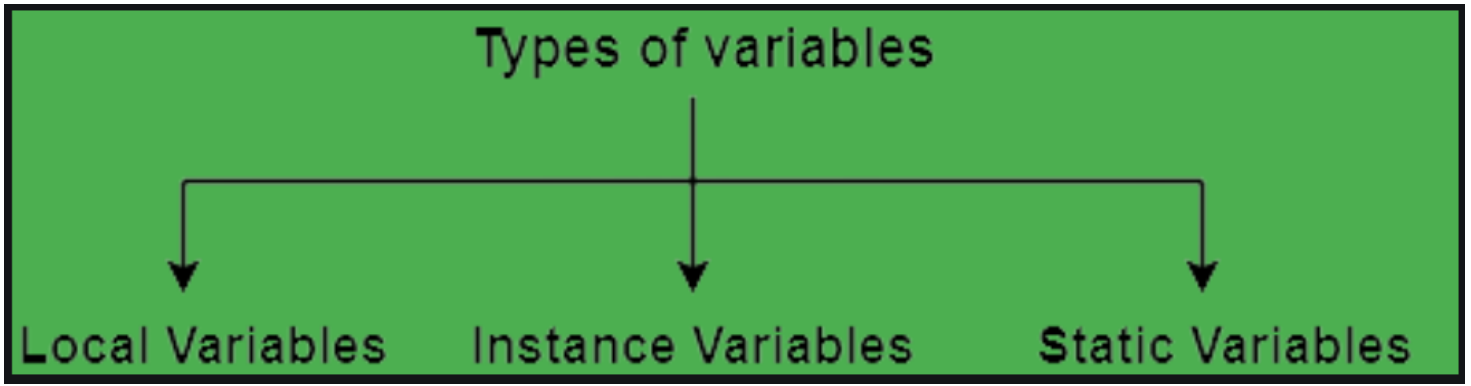
https://www.geeksforgeeks.org/variables-in-java

- It can be perceived with the help of 3 components that are as follows:
  1. datatype: Type of data that can be stored in this variable.
  2. variable_name: Name given to the variable.
  3. value: It is the initial value stored in the variable.



Types of Variable in Java

1. Local Variables
2. Instance Variables
3. Static Variables



Local Variable

A variable defined within a block or method or constructor is called a local variable.

1. These variables are created when the block is entered, or the function is called and destroyed after exiting from the block or when the call returns from the function.
2. The scope of these variables exists only within the block in which the variables are declared, i.e., we can access these variables only within that block.
3. Initialization of the local variable is mandatory before using it in the defined scope.

Local Variable

```
1.  /*package whatever //do not write package name here */
2.  // Contributed by Shubham Jain
3.  import java.io.*;

4.  class GFG {
5.          public static void main(String[ ] args)
6.          {
7.                  int var = 10; // Declared a Local Variable
8.                  // This variable is local to this main method only
9.                  System.out.println("Local Variable: " + var);
10.         }
11. }
```

Instance Variables

Instance variables are non-static variables and are declared in a class outside of any method, constructor, or block.

- As instance variables are declared in a class, these variables are created when an object of the class is created and destroyed when the object is destroyed.
- Unlike local variables, we may use access specifiers for instance variables. If we do not specify any access specifier, then the default access specifier will be used.
- Initialization of an instance variable is not mandatory. Its default value is 0.
- Instance variables can be accessed only by creating objects.

Instance Variables

```
/*package whatever //do not write package name here */
import java.io.*;
class GFG {
        public String geek; // Declared Instance Variable
        public GFG()
        { // Default Constructor
                this.geek = "Shubham Jain"; // initializing Instance Variable
        }
//Main Method
        public static void main(String[ ] args)
        {
                // Object Creation
                GFG name = new GFG();
                // Displaying O/P
                System.out.println("Geek name is: " + name.geek);
        }
}
```

Static Variables

Static variables are also known as class variables.

- These variables are declared similarly as instance variables. The difference is that static variables are declared using the static keyword within a class outside of any method, constructor or block.
- Unlike instance variables, we can only have one copy of a static variable per class, irrespective of how many objects we create.
- Static variables are created at the start of program execution and destroyed automatically when execution ends.

Static Variables

- Initialization of a static variable is not mandatory. Its default value is 0.
- If we access a static variable like an instance variable (through an object), the compiler will show a warning message, which won't halt the program. The compiler will replace the object name with the class name automatically.
- If we access a static variable without the class name, the compiler will automatically append the class name.

Static Variables

```
/*package whatever //do not write package name here */
import java.io.*;
class GFG {
public static String geek = "Shubham Jain";                    //Declared static variable

        public static void main (String[ ] args) {

        //geek variable can be accessed without object creation
        //Displaying O/P
        //GFG.geek --> using the static variable
                System.out.println("Geek Name is : "+GFG.geek);
        }
}
```

Tipe Data Primitif

STT-NF

Table 1 Primitive Types		
Type	**Description**	**Size**
`int`	The integer type, with range –2,147,483,648 (`Integer.MIN_VALUE`) . . . 2,147,483,647 (`Integer.MAX_VALUE`, about 2.14 billion)	4 bytes
`byte`	The type describing a single byte, with range –128 . . . 127	1 byte
`short`	The short integer type, with range –32,768 . . . 32,767	2 bytes
`long`	The long integer type, with range –9,223,372,036,854,775,808 . . . 9,223,372,036,854,775,807	8 bytes
`double`	The double-precision floating-point type, with a range of about $\pm 10^{308}$ and about 15 significant decimal digits	8 bytes
`float`	The single-precision floating-point type, with a range of about $\pm 10^{38}$ and about 7 significant decimal digits	4 bytes
`char`	The character type, representing code units in the Unicode encoding scheme (see Computing & Society 4.2 on page 163)	2 bytes
`boolean`	The type with the two truth values `false` and `true` (see Chapter 5)	1 bit

Literal

- Java Literals are syntactic representations of boolean, character, numeric, or string data.
- Literals provide a means of expressing specific values in your program.

Table 2 Number Literals in Java

Number	Type	Comment
6	int	An integer has no fractional part.
-6	int	Integers can be negative.
0	int	Zero is an integer.
0.5	double	A number with a fractional part has type double.
1.0	double	An integer with a fractional part .0 has type double.
1E6	double	A number in exponential notation: 1×10^6 or 1000000. Numbers in exponential notation always have type double.
2.96E-2	double	Negative exponent: $2.96 \times 10^{-2} = 2.96 / 100 = 0.0296$
100,000		**Error:** Do not use a comma as a decimal separator.
3 1/2		**Error:** Do not use fractions; use decimal notation: 3.5

Magic Number

```
public class Foo {
   public void setPassword(String password){
      // don't do this
      if (password.length() > 7){
         throw new InvalidArgumentException("password");
      }
   }
}
```

Magic Number

```
public class Foo {
    public static final int MAX_PASSWORD_SIZE = 7;

    public void setPassword(String password) {
        if (password.length() > MAX_PASSWORD_SIZE) {
            throw new InvalidArgumentException("password");
        }
    }
}
```

Variable Operations

Table 5 Arithmetic Expressions

Mathematical Expression	Java Expression	Comments
$\frac{x + y}{2}$	`(x + y) / 2`	The parentheses are required; `x + y / 2` computes $x + \frac{y}{2}$.
$\frac{xy}{2}$	`x * y / 2`	Parentheses are not required; operators with the same precedence are evaluated left to right.
$\left(1 + \frac{r}{100}\right)^n$	`Math.pow(1 + r / 100, n)`	Use `Math.pow(x, n)` to compute x^n.
$\sqrt{a^2 + b^2}$	`Math.sqrt(a * a + b * b)`	`a * a` is simpler than `Math.pow(a, 2)`.
$\frac{i + j + k}{3}$	`(i + j + k) / 3.0`	If i, j, and k are integers, using a denominator of 3.0 forces floating-point division.
π	`Math.PI`	`Math.PI` is a constant declared in the `Math` class.

Praktikum 2

Praktikum 2

- Tulislah rumus berikut dalam Bahasa Java!

$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma}$$

- Jangan lupa cantumkan juga identitas Anda (NIM & Nama) saat menjalankan program praktikum 2 ini.
- Screenshot tampilannya dan upload pada kantung tugas di Elena.

Reference

- Tawakal, Hilmi. Materi Perkuliahan PBO 2019-2020. Prodi Teknik Informatika STT-NF.

Thank You!

Subhaanakallohumma wa bihamdika, asy-hadu alla ilaha illa anta, as-tagh-firuka wa atuubu ilaik