Tutorial Link https://course.testpad.chitkara.edu.in/tutorials/Java : Variables/5a9ff5303dde0810c26d5305

TUTORIAL

Java : Variables

Topics

Variables in JAVA

Variables and keywords are the base for any programming language. One must follow the rules described for them precisely in all programs. For the syntactic measures Identifiers, constants, literals etc. are specified in every language.

Identifiers: Identifiers are used for class names, method names, and variable names. An identifier may be any descriptive sequence of uppercase and lowercase letters, numbers, or the underscore and $ sign characters. Java is case-sensitive, so MONTH is a different identifier than month.

Some examples of valid identifiers are:

```
currentDate     counter                          next_Month
$dummy          avg_of_3
```

Invalid variable names include:

```
2digits             hello&there     nextYearIs-2016
```

Literals: A constant value in Java is created by using a literal representation of it. For example, here are some literals:

```
300      97.5      ‘G’      “This is a string”
```

First of above examples is an integer, second one is a float literal, third one is a character literal and fourth is a string.

Keywords: Keywords are the words which have a predefined meaning in a language. There are 49 reserved keywords currently defined in the Java language. These are written as below in Table: -

```
abstract      assert      boolean       break     byte
case          catch
char          class       const         continue  default    do
          double
else          extends     final         finally   float      for
          goto
if            implements  import        instanceof  int
interface     long
native        new         package       private   protected
public        return
short         static      strictfp      super     switch
synchronized  this
throw         throws      transient     try       void
volatile      while
```

These keywords cannot be used as names for a variable, class, or method. There are some more words which are reserved by JAVA such as true, false, null etc. which cannot be used as a variable or class name in user programs.

Variables

A variable can be defined by an identifier. A data type must also be there to tell the compiler about the domain of values to be handled by that variable. Variable have their scope to define the visibility and they also have their lifetime. JAVA is a strongly typed language so uppercase letters and lowercase letters are different. To use a variable one must declare it.

Syntax for declaring a variable in JAVA: -

```
Data-type variable;
```

Data-type is one of JAVA atomic types or can be a class or interface. Data-types are discussed in next section and class and packages are discussed in later chapters. Some of the examples of variable declaration in JAVA are: -

```
int date, month, year;
int current_Year = 2015;
double pi = 3.14159;
```

A variable name must begin with a letter and must be a sequence of letters or digits. Note that the terms "letter" and "digit" are much broader in Java than in most languages. A letter is defined as 'A'–'Z', 'a'–'z', '_', or any Unicode character that denotes a letter in a language. For example, German users can use umlauts such as 'ä' in variable names; Greek speakers could use a π. Similarly, digits are '0'–'9' and any Unicode characters that denote a digit in a language. Symbols like '+' or '©' cannot be used inside variable names, nor can spaces. All characters in the name of a variable are significant and case is also significant. The length of a variable name is essentially unlimited.

After you declare a variable, you must explicitly initialize it by means of an assignment statement—you can never use the values of uninitialized variables. For example, the Java compiler flags the following sequence of statements as an error:

Java

```java
class Main
{
  public static void main(String ab[])
  {
    int month;
    System.out.println(month); // ERROR--variable not initialized
  }
}

```

You assign to a previously declared variable by using the variable name on the left, an equal sign (=), and then some Java expression that has an appropriate value on the right.

```
int month;
month = 12;
```

You can both declare and initialize a variable on the same line. For example:

```
int month = 12;
```

So, if we rewrite the above program as below, then it works fine:

```java
class Main
{
  public static void main(String ab[])
  {
    int month=12;
    System.out.println(month);
  }
}

```

Finally, in Java you can put declarations anywhere in your code. For example, the following is a valid code in Java:

```
double rent = 65000.0;
System.out.println(rent);
int month = 12; // ok to declare a variable here
```

Normally all the variables are declared in main() function, but Java allows variables to be declared in any block. A block defines a scope. Thus, each time you start a new block, you are creating a new scope. As you probably know from your previous programming experience, a scope determines what objects are visible to other parts of your program. It also determines the lifetime of those objects.

Naming Conventions

```
Identifier Type          Convention
      Example
Class name               Capitalize each word within identifier
      ConnectionManager
Method names             Capitalize each word except the first
      connectPhone
Variable names           Capitalize each word except the first
      phoneNumber
Constant                 Capitalize each word with underscores
      MAX_CONNECTIONS
```

Constants: In Java, you use the keyword final to denote a constant. For example:

```java
class Main
{
  public static void main(String[] args)
  {
    final double PI = 3.14;
    double length = 8.5;
    double Height = 11;
    System.out.println("Calculation is : " + (length * PI * Height));
  }
}

```

The keyword final indicates that you can assign to the variable once, and then its value is set once and for all. It is customary to name constants in all uppercase.