Exclusive study materials developed by Anisoft Computer Academy.

# Introduction to the key features of the JAVA language

Tokens

When we submit a source code for compilation, the java compiler parses the text of the program. It removes all the comments and white spaces and extracts individual tokens. A token is the smallest element in a program that is meaningful to the compiler. These tokens define the structure of the language. All the tokens that java comprises of is called the java token set. The java token set is divided into five categories: IDENTIFIERS, KEYWORDS, LITERALS, OPERATORS & SEPERATORS.

Identifiers

Identifiers are the names assigned by the programmer to variables, methods, functions, classes etc to uniquely identify them to the compiler. Java identifier can be of any length (upper limit of 255 characters). It starts with a letter or an undersign character(\_) or a dollar sign ($). Subsequent characters can be alphabets and numbers, space or any other special characters are not allowed. A java keyword cannot be used as an identifier.

A list of valid java identifiers:

HelloWorld  
ACA  
ACA\_Centre  
$Value  
red

A list of invalid java identifiers:

3 Dimensional - Beginning with a number.  
 Page # - # character not allowed  
 short - java keyword  
 run-time - character in between

In addition to the above rules certain naming conventions are also followed while programming:

* Multiple word names start with a lowercase letter and the subsequent words start with uppercase letters. (e.g. : isHeAlive)
* It is widely accepted convention to start the class name with a capital letter. (e.g. : class HelloWorld)

Keywords

Java keywords are the reserved words and have a special meaning for the java compiler. They cannot be used as identifiers in a program. Java has a rich set of keywords some of which are as follows:

Void

static

public

byte

Literals

Literals are the elements that are used in an invariant manner in a program. Literals are also called constants. Literals can be numbers, characters or string. A program element whose value remains unchanged during the course of the program is called a literal or constants. Java does not directly support constants, you can declare variables with initial value and make them constants .

Example

Int num = 415 (Here,num is a constant int which holds value”415”.)

Types of constants:-

1. Integer-constants – Integer constants are constant data elements that have no fractional parts or exponents.
2. Floating constants- They express numbers with decimals and/or exponents.
3. Character constants- a character constant is a single character enclosed between single quotes (‘).
4. String constants- A string constant consists of zero or more characters from the basic character set enclosed within double quotes(“ “).

Operators

Variables and constants store information. Operators are used to operate on them. Operators perform some sort of computation (arithmetic or logical). In simple words an operator in java is a symbol that performs operations on one or more operands to produce a result.

Java operators can be classified as unary, binary or ternary based on the number of arguments an operator takes.

|  |  |  |
| --- | --- | --- |
| OPERATOR TYPE | MEANING | EXAMPLE |
| UNARY | A unary operator takes only one operand. | A++, !A |
| BINARY | A binary operator takes two operands. | A+B , x % y |
| TERNARY | A ternary operator takes three elements. | A + B + C |

Java operators (Basic categorization)

|  |  |
| --- | --- |
| OPERATOR | USAGE |
| Assignment operator | = |
| Compound assignment operator | += -= \*= /= %= <<= >>= >>>= |
| Arithmetic operators | * + \* / % |
| Increment and decrement operators | ++ -- |
| Relational operators | < > >= <= == != |
| Logical operator | && || & | ! ^ |
| Bitwise operator | & | ^ >> >>> |
| Conditional operator | ?: |

SEPARATORS

Punctuators or separators in java are special characters that have some special meaning to the compiler. But they do not by themselves perform any particular operations giving out some results. Separators are used to inform java compiler, how statements are grouped together in the code.

|  |  |
| --- | --- |
| Separator | Purpose |
| () | Encloses arguments in method definitions and calling; adjusts precedence in arithmetic expressions; |
| {} | Defines blocks of code and automatically initializes arrays. |
| [] | Declares array types and references array values |
| ; | Terminates statements |
| , | Separates successive identifiers in variable declarations; chains statements in the test expression of a for loop. |
| . | Selects a field or a method from an object; separates package names from sub-package and class names. |
| : | Used after labels. |

What are DATATYPES, EXPRESSIONS and OPERATORS in java programming language?

# Datatypes

Datatypes define the storage requirement for representing information. Datatypes determine how the compiler interprets the contents of a memory location. Every variable in the source code has a datatype associated with it. A datatype is assigned to a variable at the time of creation of that variable. A variable is created by declaring it in the source code. At the time of declaration of a variable we must specify two things, the variable’s datatype and its name. The syntax of java variable declaration statement is as follows:-

Datatype identifier ;

Datatype identifier 1 , identifier 2;

Example :-

Int val;

Float length, breadth;

Java datatypes can be divided into two categories: basic datatypes & composite datatypes. Basic datatypes are those datatypes which are known to the compiler. Integer, characters, floating point and Boolean are the basic datatypes. Composite datatypes are derived from the basic datatypes, compiler does not know about these datatypes,the programmer has to educate the compiler about these datatypes before the compiler can deal with them. Composite datatypes include strings,arrays,classes and interfaces.

*Integer Datatypes*

There are four integer datatypes: byte, short, int, long.

Java integer datatypes : (Memory space)

|  |  |
| --- | --- |
| Datatypes | Size |
| Byte | 8 bits |
| Short | 16 bits |
| Int | 32 bits |
| Long | 64 bits |

An integer variable can be declared as:

Int I ;

Long k ;

*Floating Point datatypes*

Floating point datatypes are used to present numbers with a fractional part. There are two floating point types: float, double

Java floating point datatypes : (Memory space)

|  |  |
| --- | --- |
| Datatype | Size |
| Float | 32 bits |
| Double | 64 bits |

32 bit floating point numbers are called single precision floating point numbers and 64 bit floating point numbers are called double precision floating point numbers.

A floating point variable can be declared as:

Float weight ;

Double d ;

*Boolean datatypes*

Boolean datatypes are used to store values with two states true or false. Has a storage capacity of 1 bit. A Boolean datatype variable is declared as follows:

Boolean isalive ;

*Character datatypes*

Character datatype is used to store a single Unicode character. Unicode character sets are 16 bits value. Therefore the space required to store a single character is 16 bits.

A character variable can be declared as:

Char ch ;

Char c ;