**Tokens**

In Java, tokens are the smallest elements of a program that are meaningful to the compiler. They are the building blocks of a Java program. Here are the main types of tokens in Java:

1. **Keywords**: Reserved words in Java that have a specific meaning to the compiler.

Examples include

class, public, static, void, int, if, else, while, for, switch, case, try, catch, finally, return, break, continue, new, this, super, import, package, extends, implements, interface, abstract, synchronized, volatile, transient, final, private, protected, throws, throw, default, instanceof, enum, assert, and strictfp.

1. **Identifiers**: Names given to various program elements such as variables, methods, classes, and interfaces. Identifiers must begin with a letter (A-Z or a-z), a dollar sign ($), or an underscore (\_), followed by letters, digits (0-9), dollar signs, or underscores. They are case-sensitive.

Examples:

int number;

String name;

MyClass myObject;

1. **Literals**: Constants or fixed values that are assigned to variables. There are several types of literals in Java:

* Integer literals (e.g., 10, 0b1010, 012, 0xA)
* Floating-point literals (e.g., 10.5, 1.23e2)
* Character literals (e.g., 'a', '\n')
* String literals (e.g., "Hello", "World")
* Boolean literals (true, false)
* Null literal (null)

Examples:

int age = 25;

double salary = 50000.75;

char grade = 'A';

String message = "Hello, world!";

boolean isStudent = true;

1. **Operators**:

Symbols that perform operations on variables and values. Operators are categorized into:

* Arithmetic operators: +, -, \*, /, %
* Relational operators: ==, !=, >, <, >=, <=
* Logical operators: &&, ||, !
* Bitwise operators: &, |, ^, ~, <<, >>, >>>
* Assignment operators: =, +=, -=, \*=, /=, %=, &=, |=, ^=, <<=, >>=, >>>=
* Unary operators: +, -, ++, --, !
* Ternary operator: ? :

int a = 5 + 3;

boolean result = (a > 3) && (a < 10);

1. **Separators**: Symbols used to separate tokens in the source code. These include:

* Parentheses () used for method calls and grouping expressions
* Braces {} used to define blocks of code
* Brackets [] used to define arrays
* Semicolon ; used to terminate statements
* Comma , used to separate variables in declarations
* Dot . used to access members of a class or object

int[] numbers = {1, 2, 3};

System.out.println("Hello, world!");

1. **Comments**: Notes or explanations within the source code that are ignored by the compiler. There are three types of comments:

* Single-line comments: // comment
* Multi-line comments: /\* comment \*/
* Documentation comments: /\*\* comment \*/