Xebia

Language Fundamentals

Data Types

Primitive Data Types:

- byte
- short
- int
- long
- float
- double
- char
- boolean

Reference Data Types:

- Objects
- Arrays
- String
- Classes
- Interfaces
- Enums

Special Data Type:

- void

Variables, Keywords, Literals

Variables:

- Store data values.
- Require declaration before use.
- Case-sensitive names.
- Have specific data types.
- Assigned values using =.

Keywords:

- Reserved words in Java.
- Predefined meanings.
- Cannot be used as identifiers.
- Examples: class, public, static, void, if, for, etc.

Literals:

- Constants representing fixed values.
    - Types include:
    - Integer literals (e.g., 42)
    - Floating-point literals (e.g., 3.14)
    - Character literals (e.g., 'A')
    - String literals (e.g., "Hello")
    - Boolean literals (true and false)

Comments

- Comments are used for explanations and notes in code.
  - Three types of comments:
  - Single-line comments with //.
  - Multi-line comments with /* ... */.
  - Documentation comments with /** ... */.
- Comments are ignored by the compiler.
- Good comments enhance code readability and understanding.
- Avoid excessive or redundant comments; focus on clarity and relevance.

Assignment, Initialization

Assignment:

- Process of storing a value in a variable.
- Done using = operator.
- Variables must be declared before assignment.

Initialization:

- Giving a variable an initial value at the time of declaration.
- Best practice to initialize variables to avoid default values.
- Example: int count = 0;
- They can be initialized during declaration or in constructors/static initializer blocks.

Assignment, Initialization

Default Initialization:

- Variables automatically initialized to default values if not explicitly initialized.
- Default values depend on data type.

Final Variables:

- final variables are constants and cannot be reassigned.
- Declared using final keyword.
- Example: final int MAX_VALUE = 100;

Instance and Class Variables:

- Instance variables are associated with instances (objects) of a class.
- Class variables are shared among all instances of a class.

Control Structures

IF-ELSE Statements:

- Used for conditional branching.
- else if clauses can be used for multiple condition checks.
- Useful for handling different cases based on conditions.

SWITCH-CASE Statements:

- Used for performing actions based on the value of an expression.
- break statements are used to exit the switch block after a case is executed.
- default case is optional and executes when no other cases match.
- Efficient for handling multiple values or options.

Loops - For, While, Do While, ForEach

For Loop:

- Iterates a specific number of times.
- Syntax: for (initialization; condition; update) { ... }

While Loop:

- Repeatedly executes code while a condition is true.
- Syntax: while (condition) { ... }

Do-While Loop:

- Executes code at least once, then continues while a condition is true.
- Syntax: do { ... } while (condition);

For-Each (Enhanced for) Loop:

- Iterates over elements in an array or collection.
- Syntax: for (elementType element : arrayOrCollection) { ... }

xebia.com