Java Programming Language

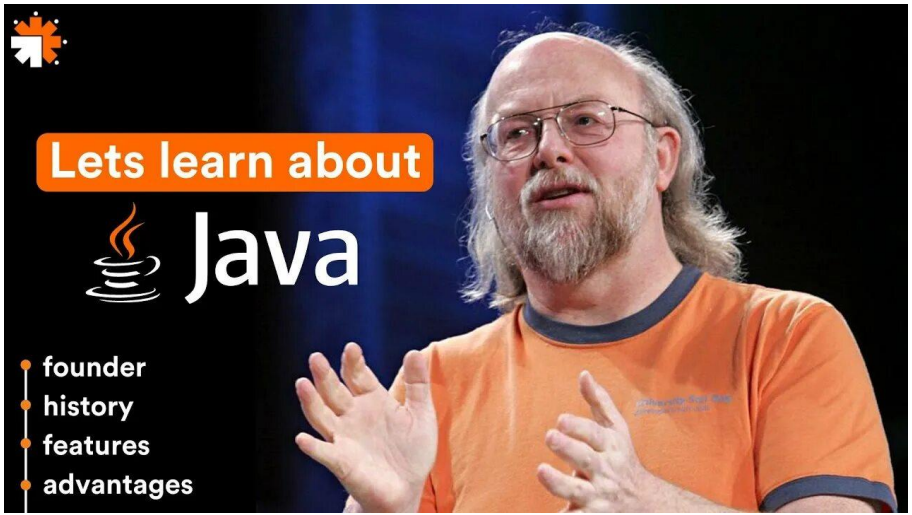
teacher: Майер С.Ф.
sfmayer@sfedu.ru

Why Java?

- General-purpose, object-oriented language.
- Runs on: mobile, desktop, servers.
- Key traits:
    - Simple syntax (C++ derived).
    - Cross-platform (JVM).
    - Robust & secure.

History of Java

- Developed by **James Gosling** at Sun Microsystems (1995).
- Originally named «**Oak**».
- Renamed to **Java** for public release.



Key Characteristics

- Simple syntax (C++-like)
- Object-oriented
- Distributed computing
- Bytecode interpretation
- Strong type checking
- Built-in security
- Cross-platform
- High performance
- Multithreading
- Dynamic adaptability

First Java Program

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

- `public class` — class declaration
- `main` — is the entry point
- `System.out.println` — output to the console

How to Run Java files (in VCode IDE)

1. Save the file as fileName.java.
2. Compile:
   bash (terminal):
   javac fileName.java
3. Run:
   bash (terminal):
   java fileName

IDE: Visual Studio Code (with plugin)

Shortcut	Action
Ctrl + K + C	Comment block
Ctrl + K + U	Uncomment block
F5	Debug run
Ctrl + F5	Run without debug
Alt + Shift + F	Format code

Java Syntax – Statements

A **statement** is a command (e.g., variable declaration, method call).

Rules:

- Formed from tokens (keywords, identifiers, operators).
- Ends with semicolon (;).

```
int myVariable = 2;
```

```
System.out.println("Hello World!");
```

Tokens: `int`, `myVariable`, `=`, `2`, `;`.

Constants in Java

Concept: named constant = permanent value (final keyword).

`final → type → name → = → value`

Syntax:

```
final DataType CONSTANT_NAME = value;
```

Example:

```
final double PI = 3.14159;
double area = PI * radius * radius;
```

Example

```
public class Main {
    public static void main(String[] args) {

        final double PI = 3.14159;

        double radius = 5;
        double area = PI * radius * radius;

        System.out.println("Circle area is: " + area);
    }
}
```

Identifiers – Naming Rules

Rules:

- Case-sensitive (myVar ≠ myvar).
- Can contain: letters, digits, _
- Must start with letter or _ (not digit).
- Cannot be a keyword (double, class).

Examples:

- Valid: myVar, _myVar.
- Invalid: 2Vars.

Variable Declarations and Initialization

```
int count;                  // integer
double radius;              // double
double interestRate;        // double
int i, j, k;                // multiple int
```

Combined declaration + initialization:

```
int count = 0;
double radius = 5.0;
double interestRate = 0.05;
```

Example task 1

Object: **Swap values between two variables** a **and** b**.**

```java
public class SwapVariables {
    public static void main(String[] args) {
        int a = 5;
        int b = 10;

        // Временная переменная для обмена значений
        int temp = a;
        a = b;
        b = temp;

        System.out.println("After swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

Java Operators – Overview

Type	Operators
Arithmetic	+, -, *, /, %
Increment/Decrement	++, --
Comparison	==, !=, <, >, <=, >=
String Concat	+
Logical	&&, ||, !
Assignment	=, +=, -=, *=, /=, %=

Compound Assignment Operators

Operator	Name	Example	Equivalent
+=	Addition assignment	i += 8	i = i + 8
-=	Subtraction assignment	i -= 8	i = i - 8
*=	Multiplication assignment	i *= 8	i = i * 8
/=	Division assignment	i /= 8	i = i / 8
%=	Modulo assignment	i %= 8	i = i % 8

```
x /= 4 + 5.5 * 1.5; // = x / (4 + 5.5*1.5)
```

Increment/Decrement Operators

- **Postfix:** `i++`, `j--` (use then increment).
- **Prefix:** `++i`, `--j` (increment then use).

```
int i = 5;
int result1 = i++;  // result1 = 5, i=6
int result2 = ++i;  // result2 = 7, i=7
```

Example task 2

Object: Calculate the printing cost of a three-volume book collection where each volume has different page counts (Volume 1: 18 pages, Volume 2: 25 pages, Volume 3: 32 pages), and every page costs 1.5 rubles to print.

Total printing cost: 112.50 rubles

```java
public class BookPrintingCost {
    public static void main(String[] args) {
        // Количество страниц в каждом томе
        int pagesVol1 = 18;
        int pagesVol2 = 25;
        int pagesVol3 = 32;

        // Стоимость печати одной страницы
        double pricePerPage = 1.5;

        // Подсчет стоимости каждого тома отдельно
        double costVol1 = pagesVol1 * pricePerPage;
        double costVol2 = pagesVol2 * pricePerPage;
        double costVol3 = pagesVol3 * pricePerPage;

        // Общая стоимость набора книг
        double totalCost = costVol1 + costVol2 + costVol3;

        System.out.printf("Total printing cost: %.2f rubles%n",
totalCost);
    }
}
```

Input with Scanner Class

Import: `import java.util.Scanner;`

Create object: `Scanner scanner = new Scanner(System.in);`

Read data:

```
int num = scanner.nextInt();
double d = scanner.nextDouble();
String text = scanner.next();
```

Close: `scanner.close();`

Example task4

Object: Convert minutes into hours and remaining minutes. Display both hours and minutes.

150 minutes is equal to 2 hour(s) and 30 minute(s).

```java
import java.util.Scanner;

public class MinutesToHours {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter total minutes: ");
        int totalMinutes = scanner.nextInt();

        int hours = totalMinutes / 60;
        int remainingMinutes = totalMinutes % 60;

        System.out.printf("%d minutes is equal to %d hour(s)
and %d minute(s).\n",
                totalMinutes, hours, remainingMinutes);
    }
}
```

150 minutes is equal to 2 hour(s) and 30 minute(s).