Java Fundamentals

Lesson 1: Java programming

Speaker: Nicolae Sîrbu
Alexandru Umaneț

Lesson Objectives

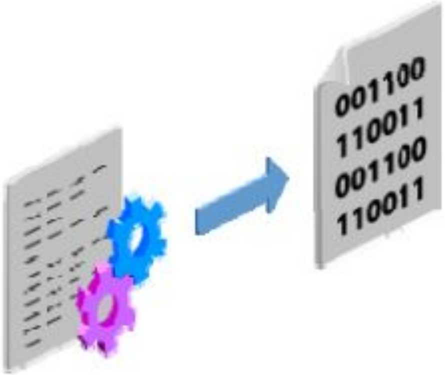
- Introduction to computer programs
- Introduction to Java language
- Installing and configuring your IDE and Java development environment

Introduction to computer programs

Purpose of a Computer Program

A computer program is a set of instructions that run in a computer or some other digital device.

- At the machine level, a program consist of binary instructions (0s and 1s).
  - Machine code
- Most programs are written in *high-level* code (human-readable).
  - Must be translated to machine code



High-level programming languages: C++, Java, Python

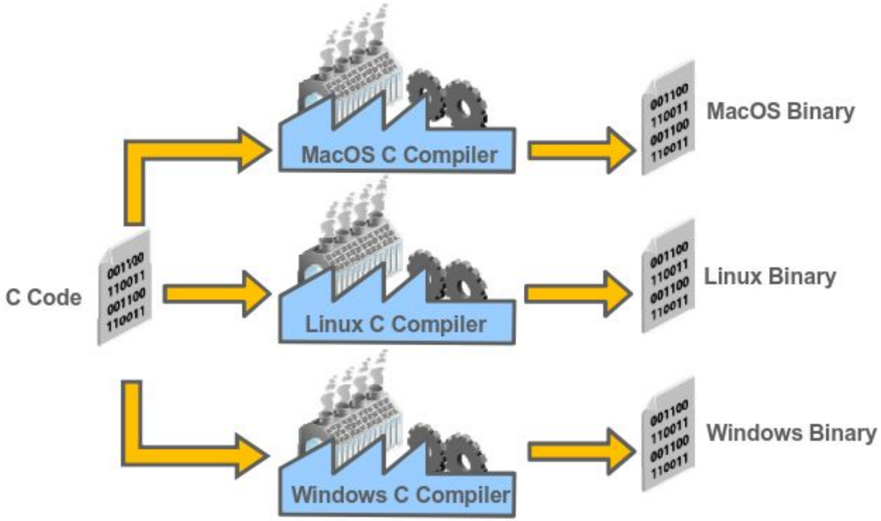
Machine code example

Machine code	Assembly language
0001 00000111	LOAD #7
0100 00001001	ADD #9
0000 00011110	STORE 30

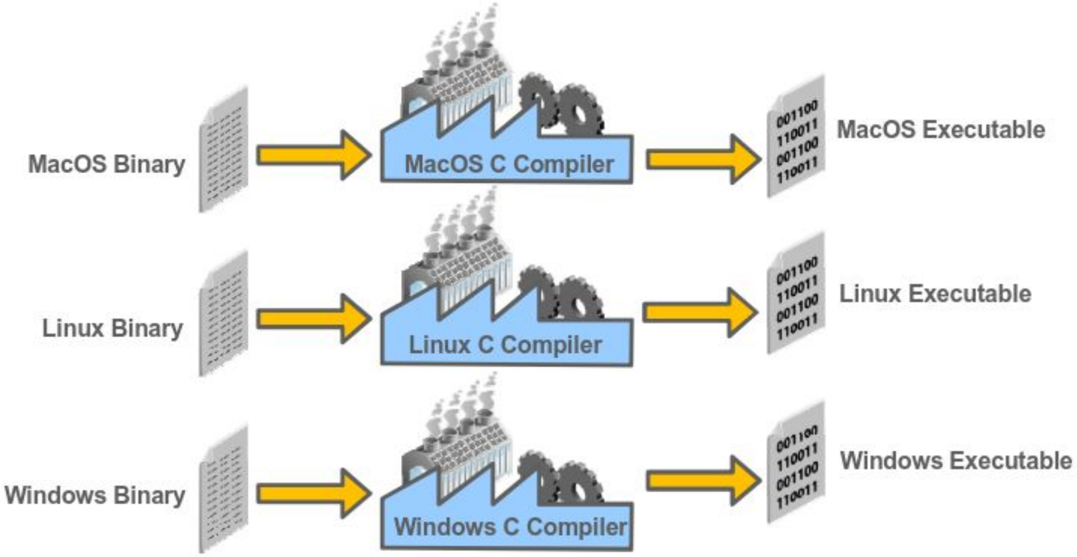
The code above, loads the integer 7 into the Accumulator, adds the integer 9 to the Accumulator, and stores the result, 16, in memory location 30.

Translating High-Level Code to Machine Code

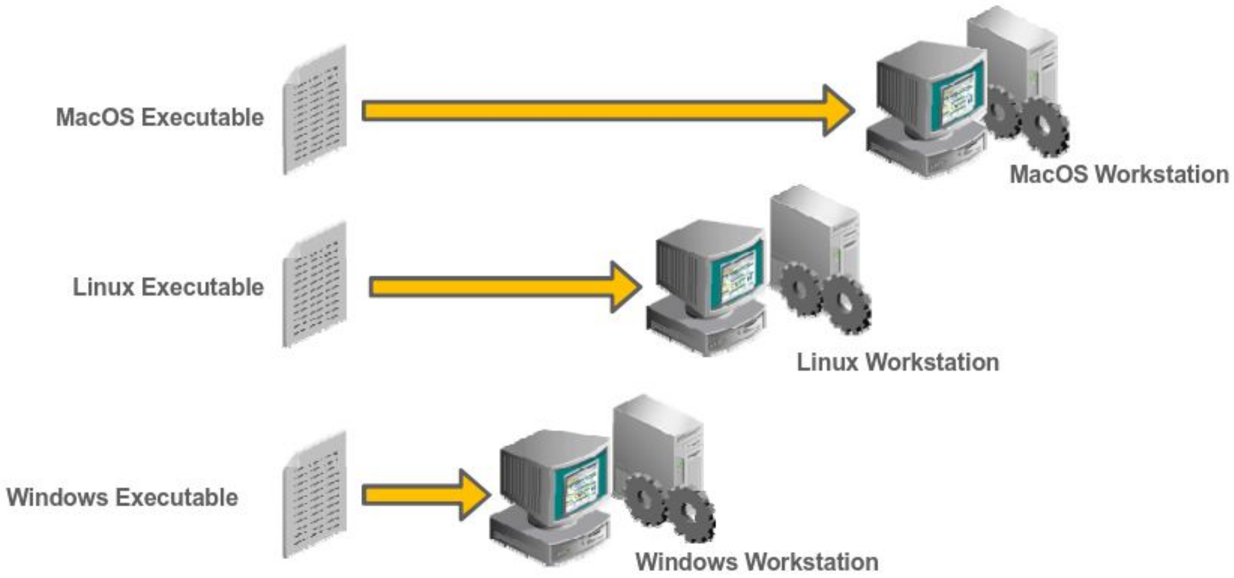
The **compiler translates the source code**, from a high-level programming language to a lower level language, **into machine code**, to create an executable program.



Linked to Platform-Specific Libraries



Platform-Dependent Programs



Introduction to the Java language

Java. Long Story Short

James Gosling

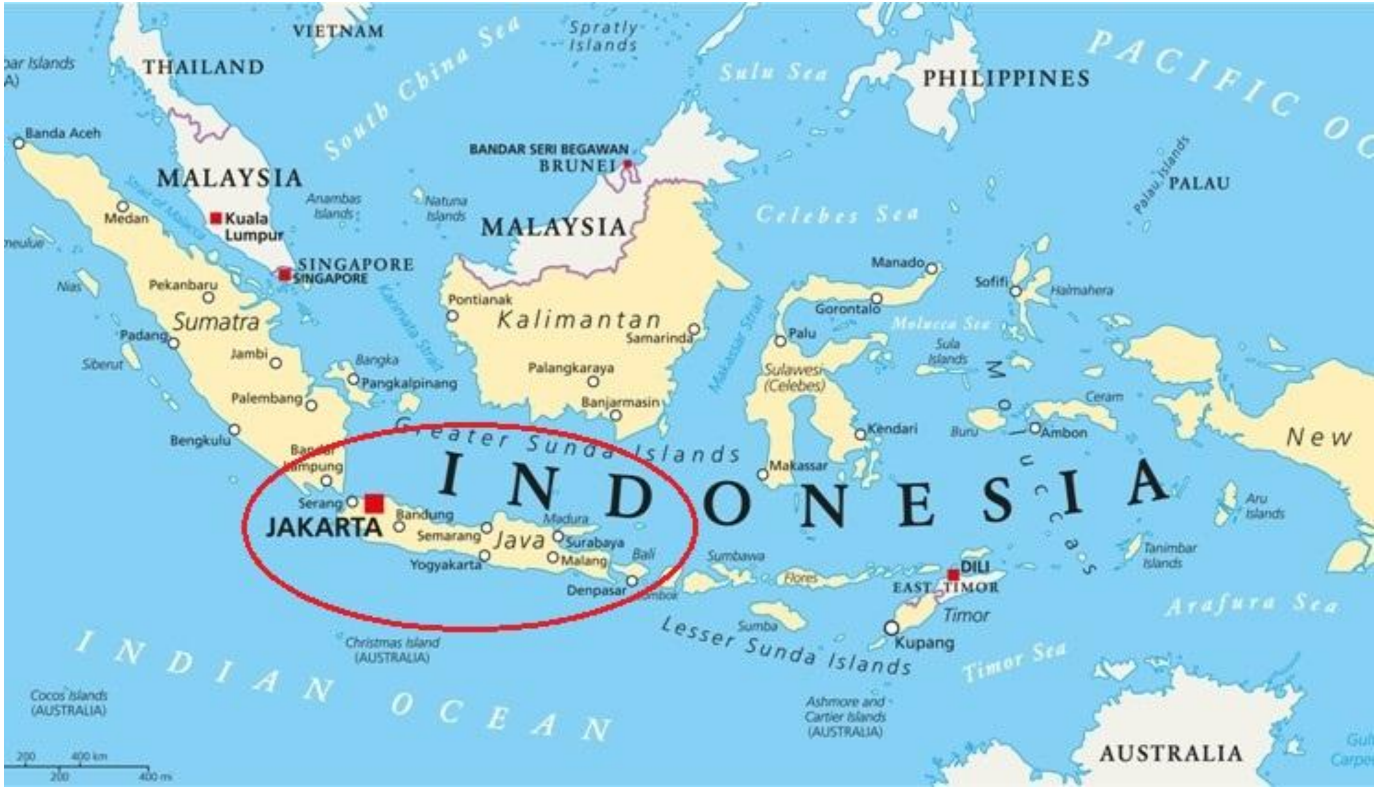
Mike Sheridan

Java. Long Story Short

How did they choose the name?

Greentalk ---> Oak ---> **Java**

Java. Long Story Short



Java. Long Story Short

How did they choose the name?

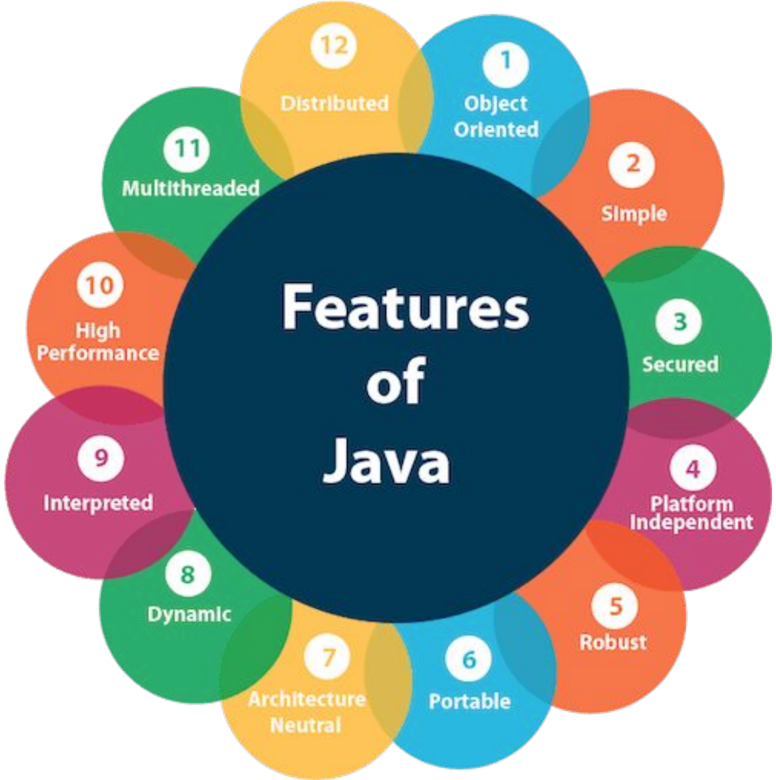
Greentalk ---> Oak ---> **Java**

Java. Long Story Short

1996 - Java 1.0

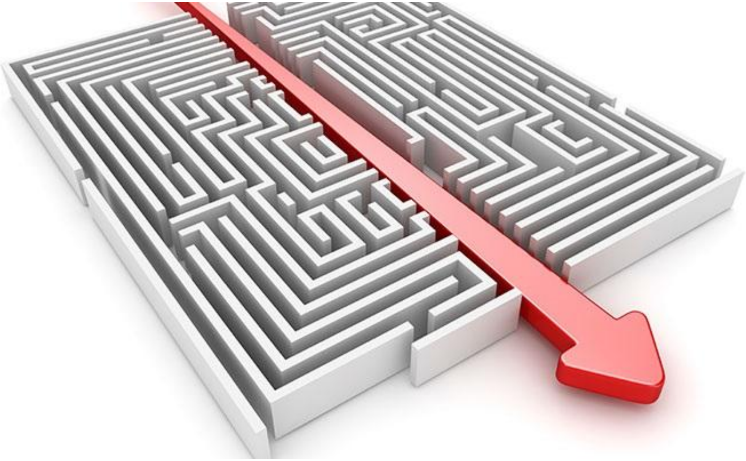
2019 - Java 14

Features of Java



Features of Java. Simple

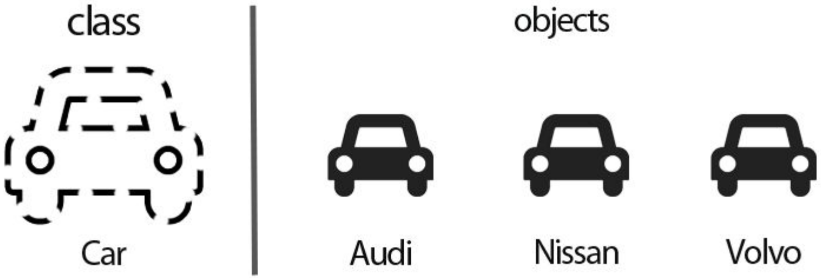
- Easy to learn and understand
- Syntax based on C++
- No pointers, no operator overloading
- No need to deallocate the memory



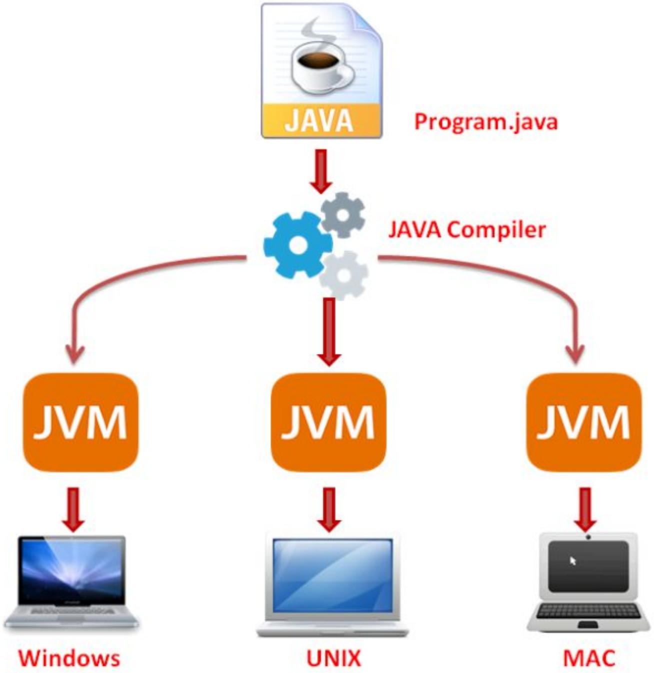
Features of Java. Object Oriented

Basic concepts of OOPs are:

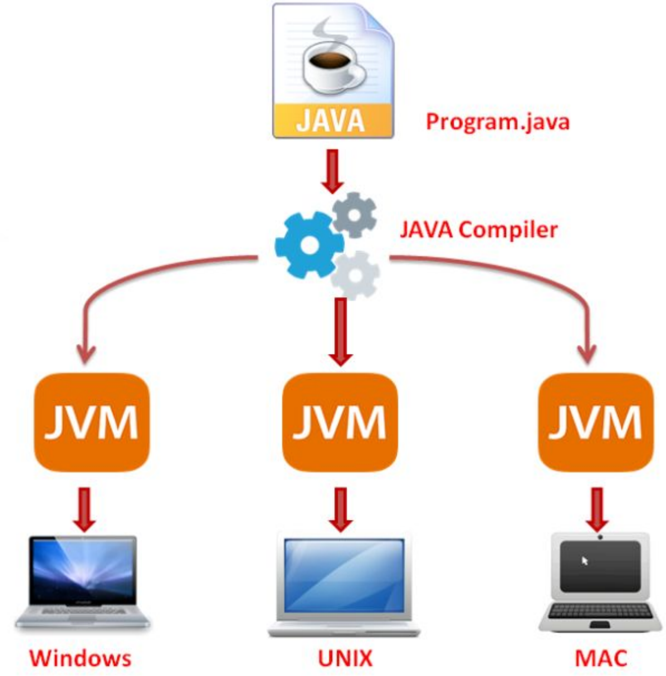
- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation



Features of Java. Platform Independent



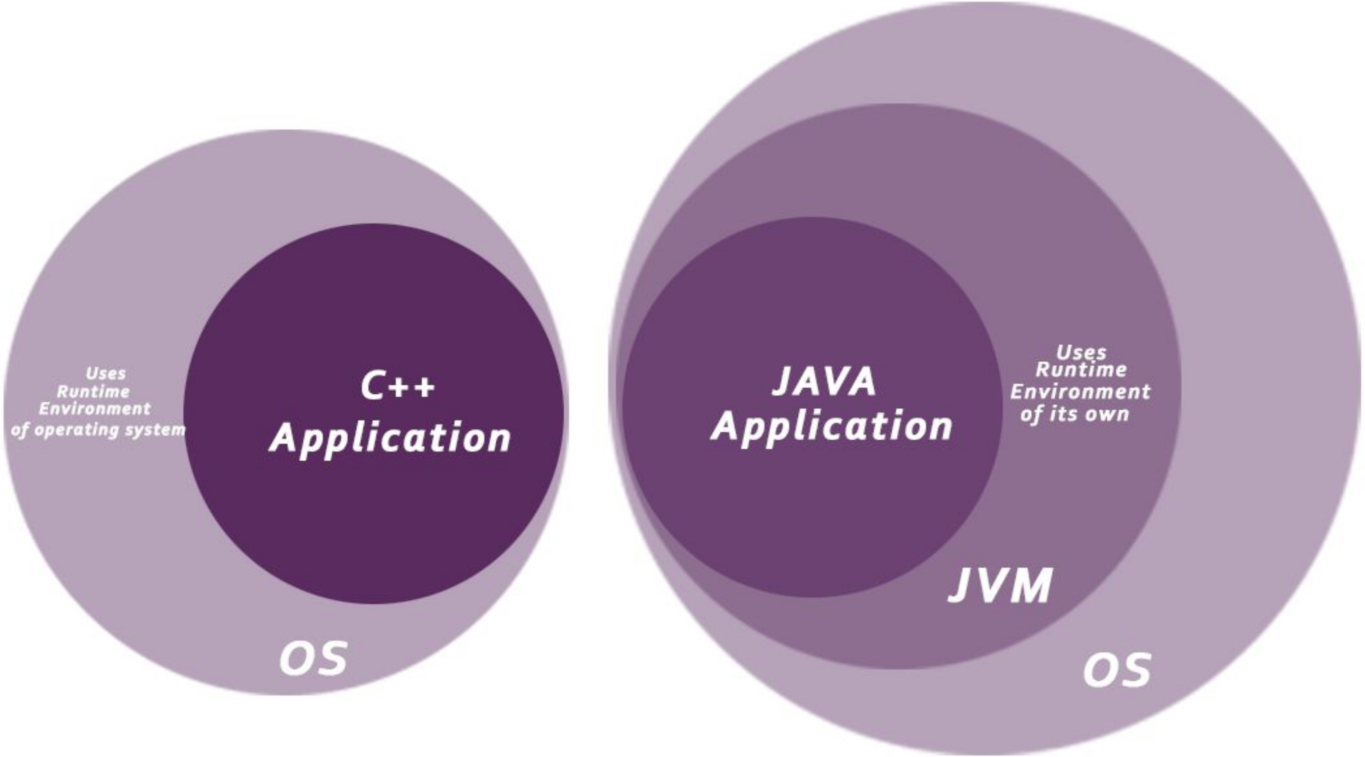
Features of Java. Platform Independent



Write Once,

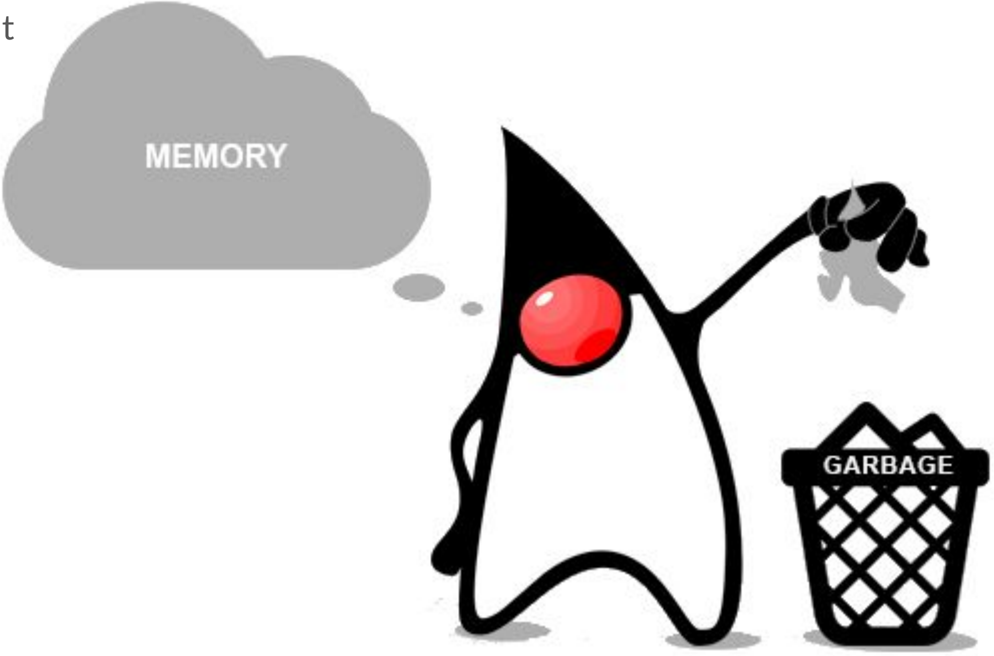
Run Anywhere

Features of Java. Secured



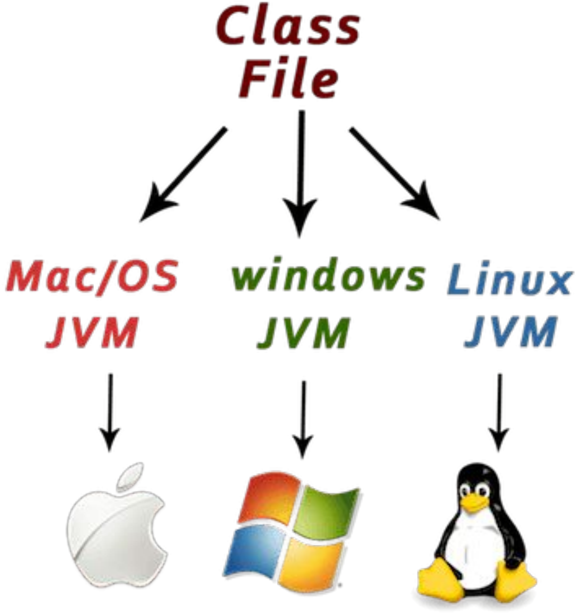
Features of Java. Robust

- Automatic memory management
- No pointers
- Strongly typed languages



Features of Java. Architecture-neutral

Features of Java. Portable

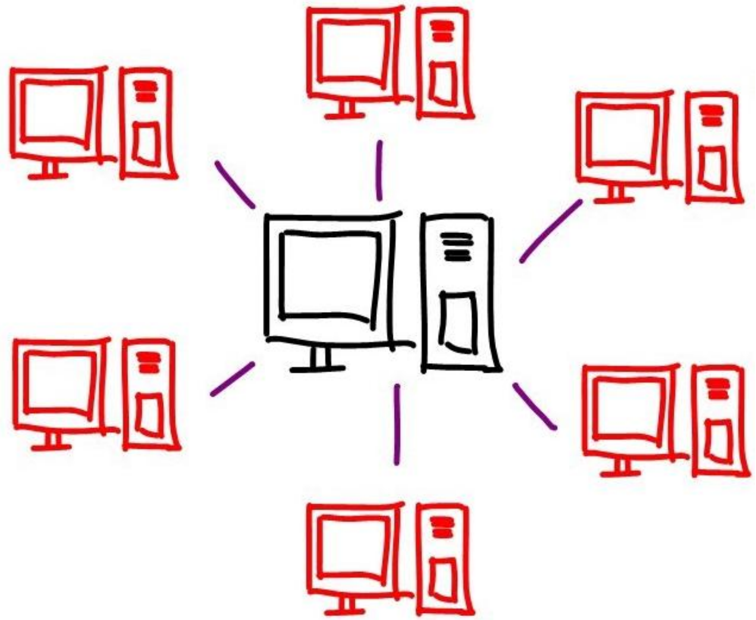


Features of Java. High-performance

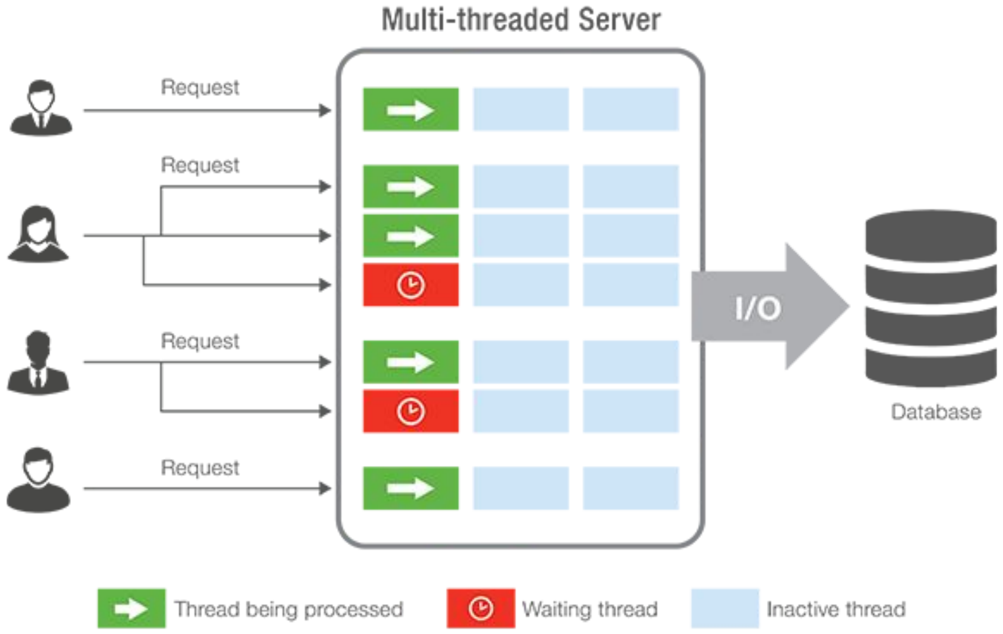
- Java is much faster than any other interpreted language.

Features of Java. Distributed

- Client-Server applications
- Sockets (TCP/IP)
- HTTP Methods
- Networking

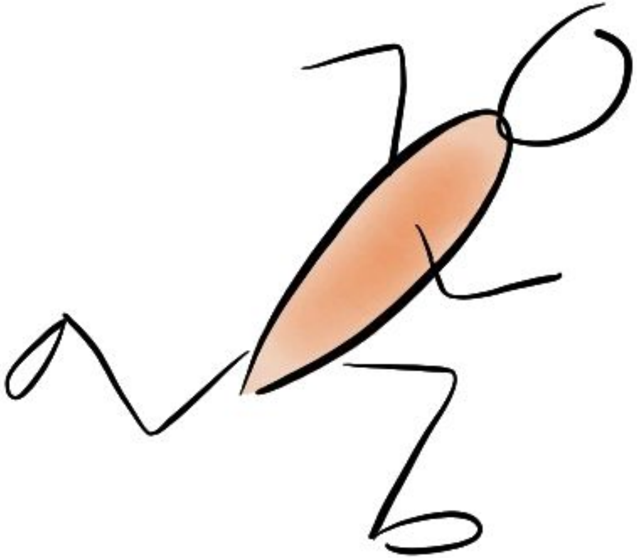


Features of Java. Multi-threaded

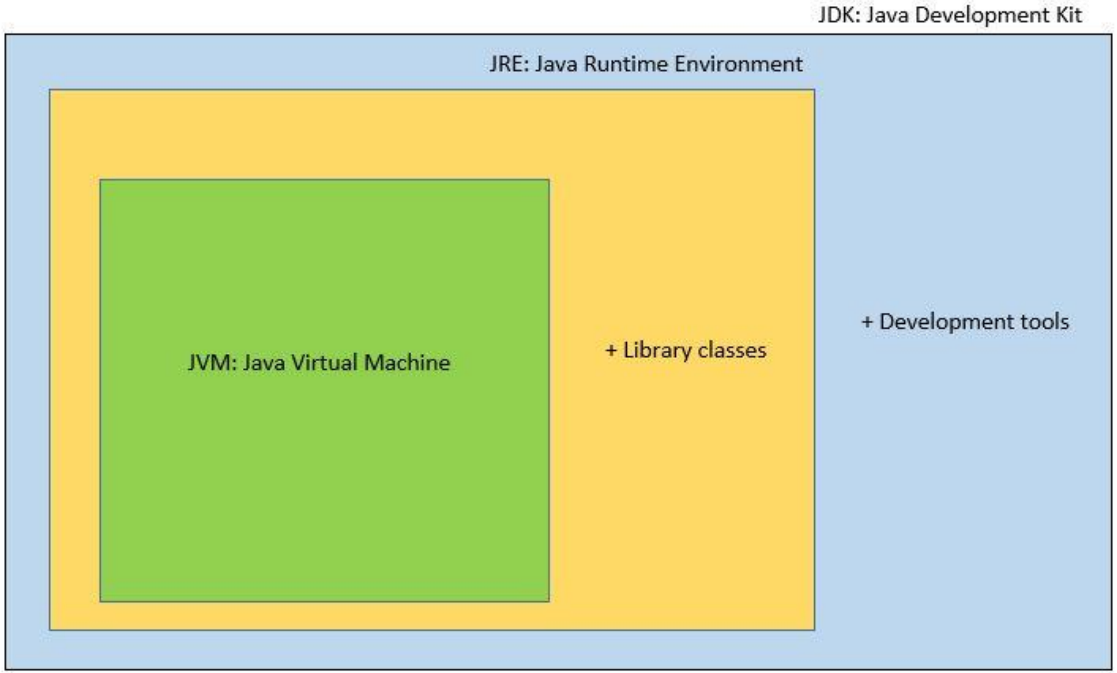


Features of Java. Dynamic

- Polymorphism - allows objects to be interoperable
- Inheritance - extends object capabilities



Java structure


JDK: Java Development Kit
JRE: Java Runtime Environment
JVM: Java Virtual Machine
+ Library classes
+ Development tools

Usage of Java in Real World

2,875,552

Android apps on Play Market

November 2019

Usage of Java in Real World

Web applications:

- Gmail
- LinkedIn
- eBay
- Aliexpress
- Confluence
- Twitter
- Facebook
- YouTube
- NASA WorldWind

Other domains:

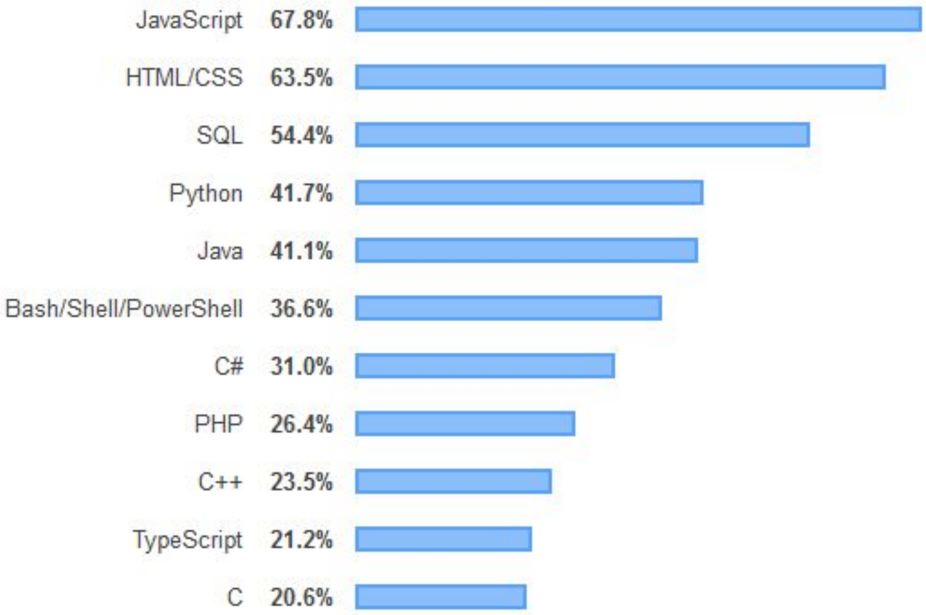
- Big Data
- Cloud Computing
- Robotics
- Banking Systems
- Games (Minecraft)
- Aeronautics
- Embedded System
- Mining
- Warehousing
- ...

Usage of Java in Real World

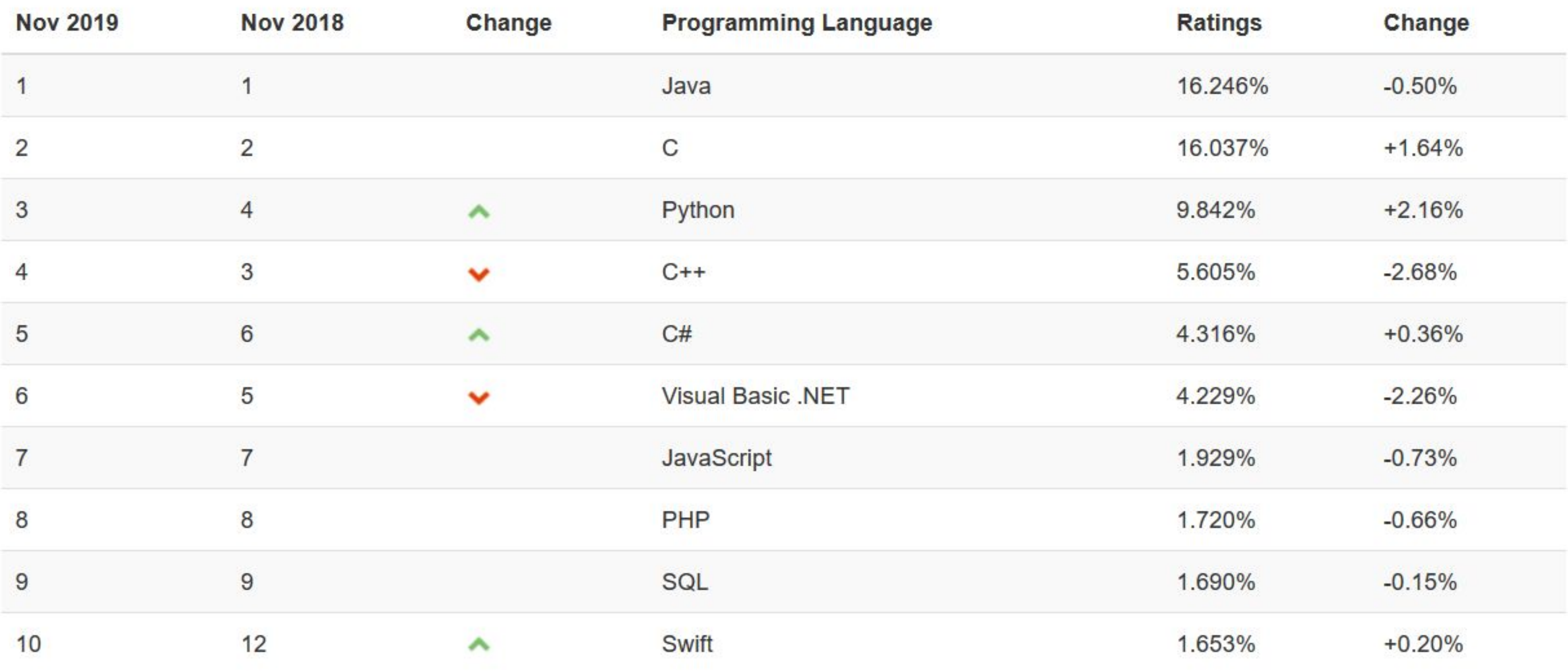
Top companies that use Java in their products:

- Uber
- eBay
- Pinterest
- Spotify
- Google
- Intel
- Symantec
- Evernote
- Nasa
- ...

Java Popularity

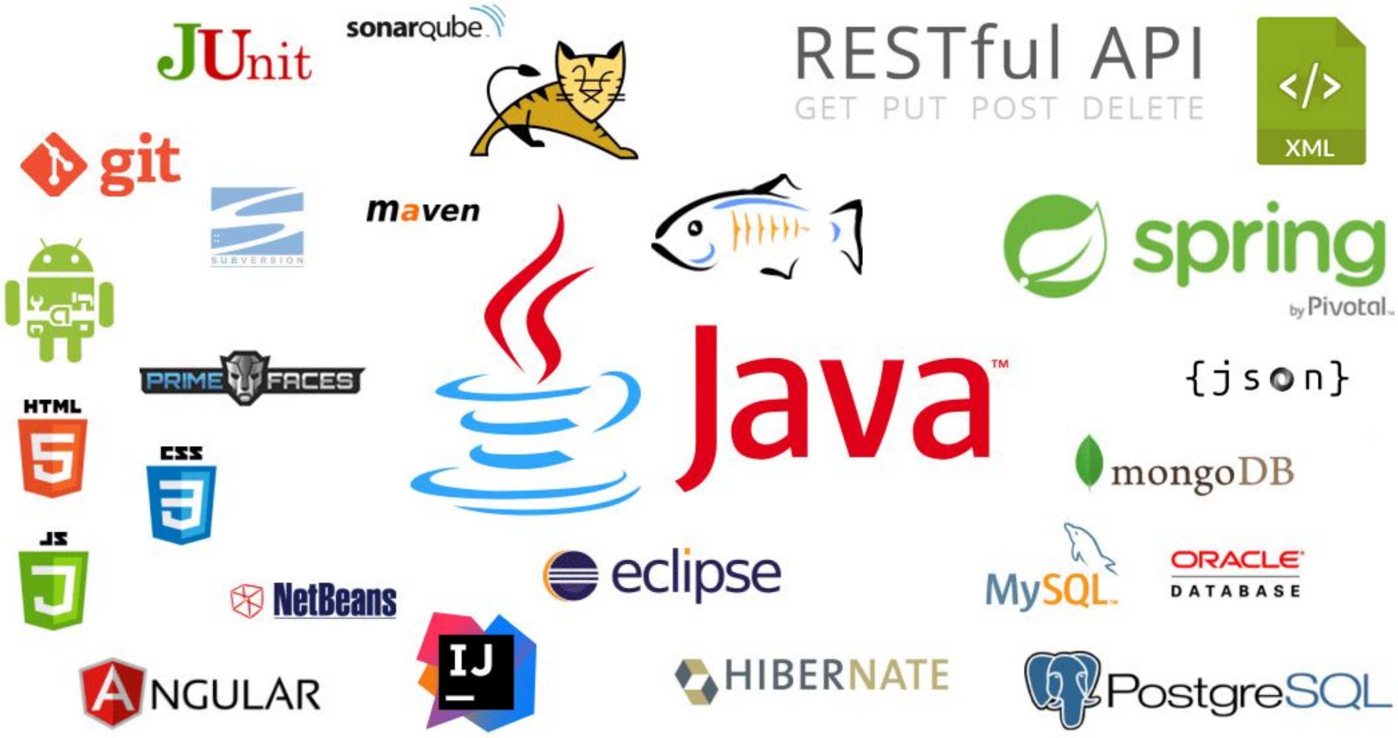


TIOBE Index for November 2019



Nov 2019	Nov 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.246%	-0.50%
2	2		C	16.037%	+1.64%
3	4	▲	Python	9.842%	+2.16%
4	3	▼	C++	5.605%	-2.68%
5	6	▲	C#	4.316%	+0.36%
6	5	▼	Visual Basic .NET	4.229%	-2.26%
7	7		JavaScript	1.929%	-0.73%
8	8		PHP	1.720%	-0.66%
9	9		SQL	1.690%	-0.15%
10	12	▲	Swift	1.653%	+0.20%

Java Ecosystem – It is Not Just a Language



Programming paradigms

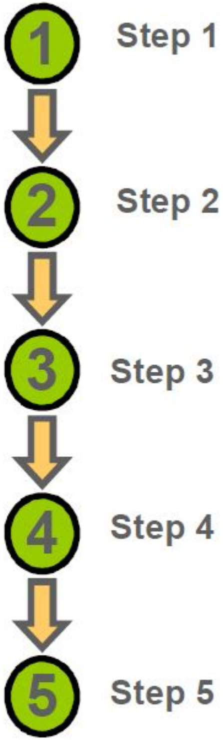
Procedural programing

Procedural programming uses a list of instructions to tell the computer what to do step-by-step.

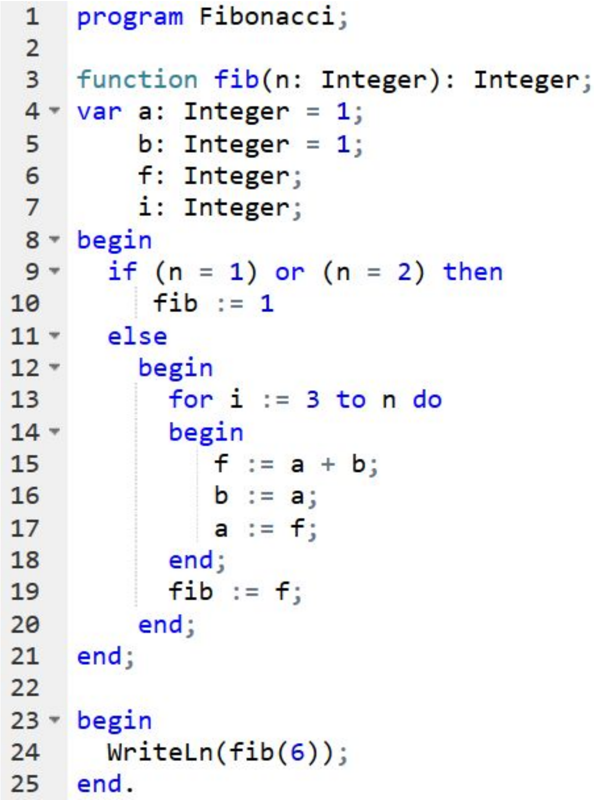
Drawbacks to procedural programming:

- Difficult to translate real-world use cases to sequential pattern
- Difficult to maintain programs
- Difficult to enhance as needed

Ex: Fortran, Cobol, Pascal, C



Procedural programing. Code example

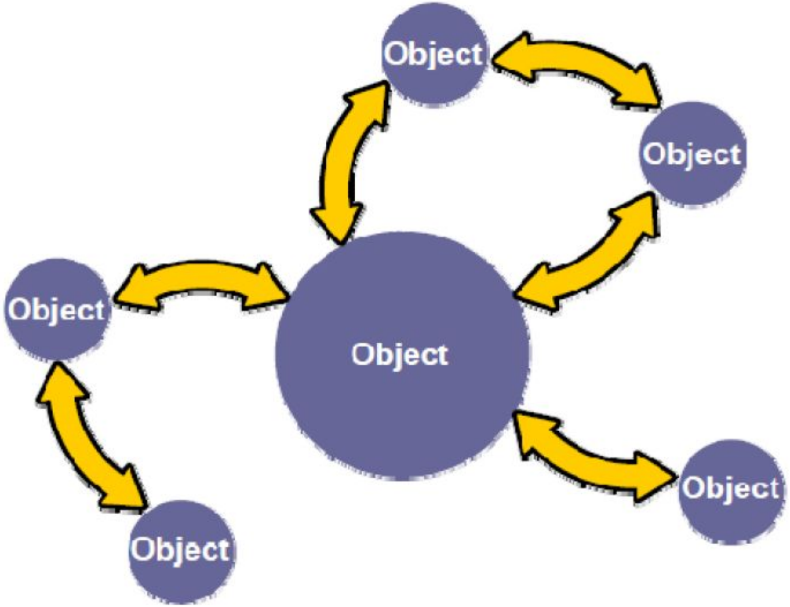


```
program Fibonacci;

function fib(n: Integer): Integer;
var a: Integer = 1;
    b: Integer = 1;
    f: Integer;
    i: Integer;
begin
  if (n = 1) or (n = 2) then
     fib := 1
  else
    begin
      for i := 3 to n do
      begin
         f := a + b;
         b := a;
         a := f;
      end;
      fib := f;
    end;
end;

begin
  WriteLn(fib(6));
end.
```

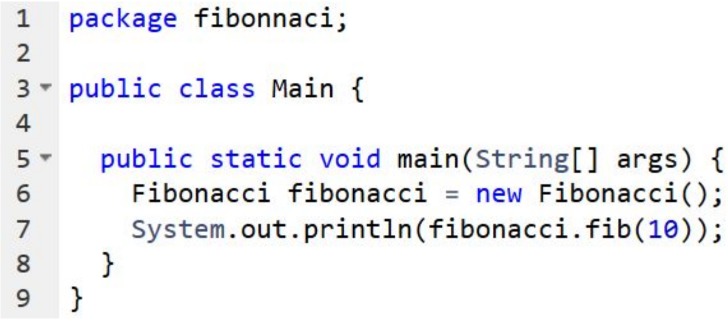
Object-Oriented programming

- Interaction of objects
- No prescribed sequence
- Benefits:
  - Modularity
  - Information hiding
  - Code reuse
  - Maintainability

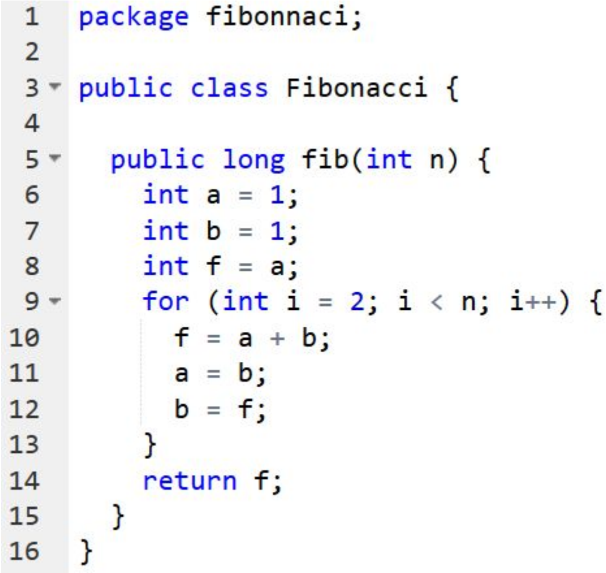
Ex: C++, Python, C#, Java



Object-Oriented programming. Code example



```java
package fibonnaci;

public class Main {

  public static void main(String[] args) {
    Fibonacci fibonacci = new Fibonacci();
    System.out.println(fibonacci.fib(10));
  }
}
```



```java
package fibonnaci;

public class Fibonacci {

  public long fib(int n) {
    int a = 1;
    int b = 1;
    int f = a;
    for (int i = 2; i < n; i++) {
      f = a + b;
      a = b;
      b = f;
    }
    return f;
  }
}
```

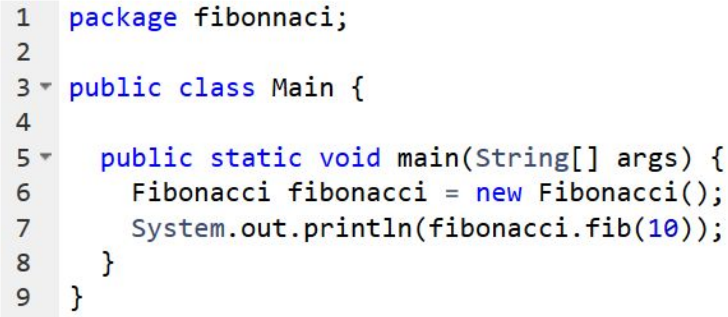
Functional programming

Functional Programming is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

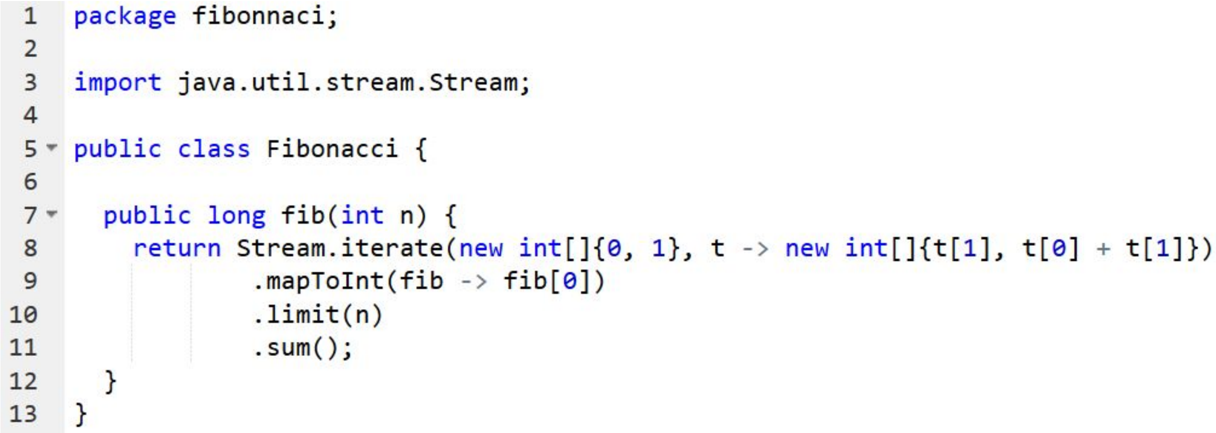
In functional code, the output value of a function depends only on the arguments that are passed to the function.

Calling a function `f` twice with the same value for argument `x` will produce the same result `f(x)` each time. It is the notion of **Pure function**.

Functional programming. Code example



```java
package fibonnaci;

public class Main {

  public static void main(String[] args) {
    Fibonacci fibonacci = new Fibonacci();
    System.out.println(fibonacci.fib(10));
  }
}
```



```java
package fibonnaci;

import java.util.stream.Stream;

public class Fibonacci {

  public long fib(int n) {
    return Stream.iterate(new int[]{0, 1}, t -> new int[]{t[1], t[0] + t[1]})
            .mapToInt(fib -> fib[0])
            .limit(n)
            .sum();
  }
}
```

IDE & JDK configuration

Exercise #2.1 Configuring your IDE and JDK

1. Download and install Java Development Kit (JDK 8) .
2. Check your Java version using Command Prompt: `java -version`
3. Add `JAVA_INSTALL_DIR/bin`to your Windows `Path` variable.
4. Download and install NetBeans.
   (https://netbeans.org/)

Using the IDE

A Java **Integrated Development Environment** (IDE) is a type of software that makes it easier to develop Java applications.

An **IDE** provides:

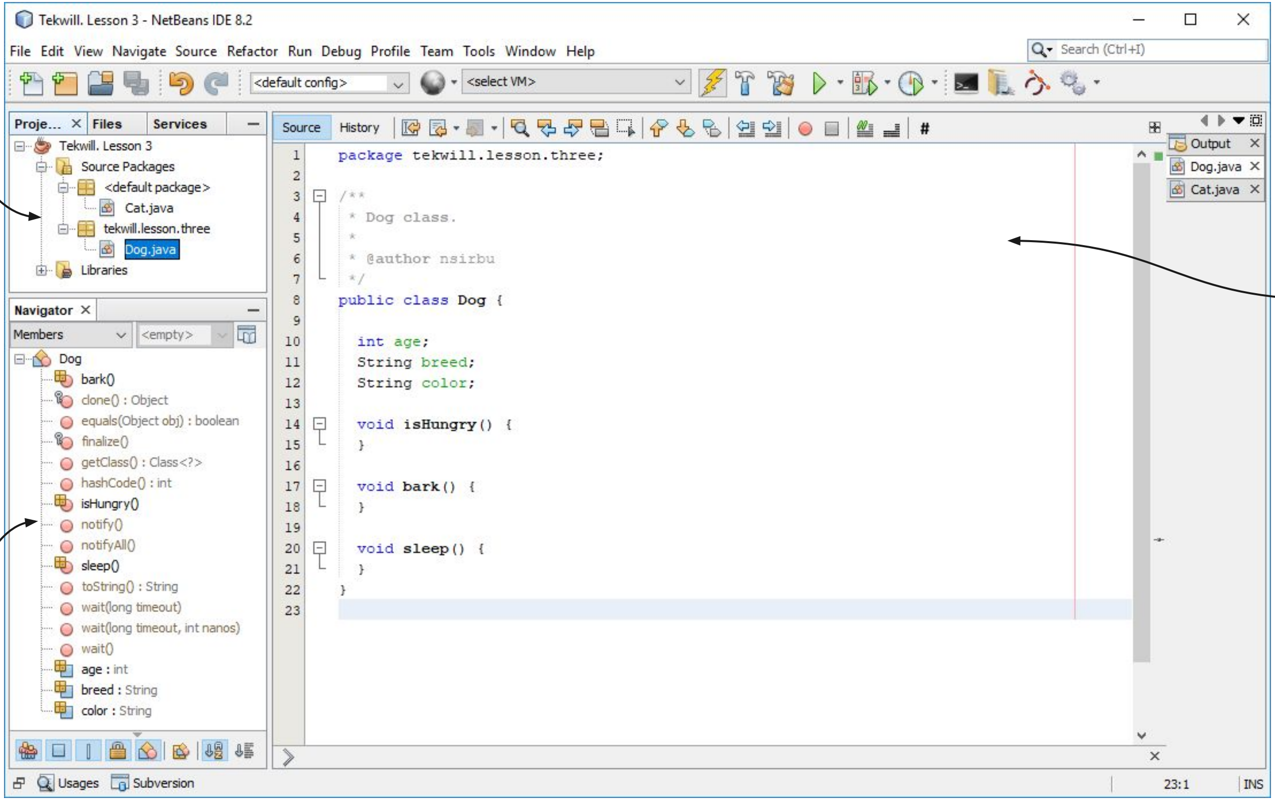
- Syntax checking
- Various automation features
- Runtime environment for testing
- Organizes all Java resources and environment settings into a Project
- Projects contain packages
- Packages contain files, such as **.java**

Using the IDE

Project Navigator

Class Navigator

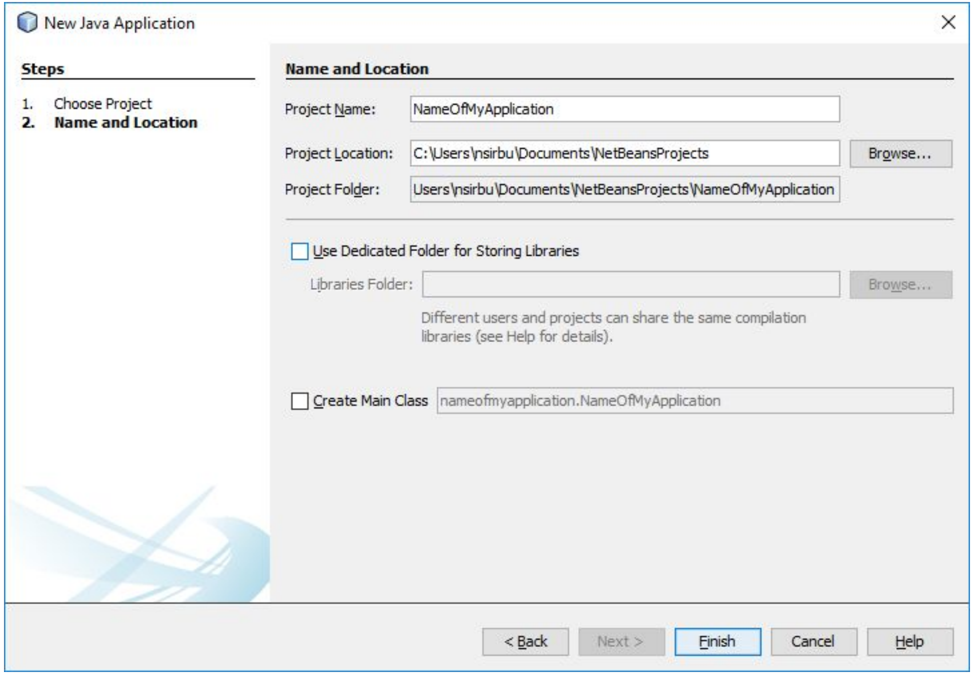
Code editor



```
package tekwill.lesson.three;

/**
 * Dog class.
 *
 * @author nsirbu
 */
public class Dog {

  int age;
  String breed;
  String color;

  void isHungry() {
  }

  void bark() {
  }

  void sleep() {
  }
}
```

Creating a Java Project

1. Select **File** > **New Project**.
2. Select **Java Application**.
3. Name and set the location for the project.
4. Select “*Create Main Class*” if you want it done for you automatically.
5. Click **Finish**.



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: NameOfMyApplication

Project Location: C:\Users\nsirbu\Documents\NetBeansProjects

Project Folder: Users\nsirbu\Documents\NetBeansProjects\NameOfMyApplication

□ Use Dedicated Folder for Storing Libraries

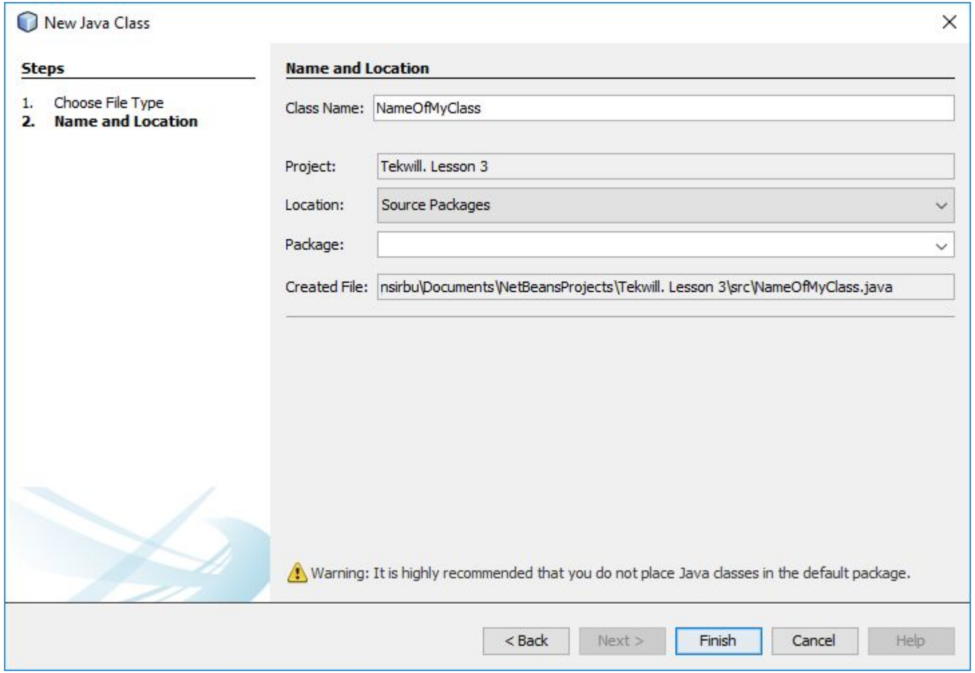
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

□ Create Main Class nameofmyapplication.NameOfMyApplication

Creating a Java Class

1. Select **File** > **New File.**
2. Select your project and choose **Java Class.**
3. Name the class.
4. Assign a package.
5. Click **Finish.**



Class Name: NameOfMyClass

Project: Tekwill. Lesson 3

Location: Source Packages

Package:

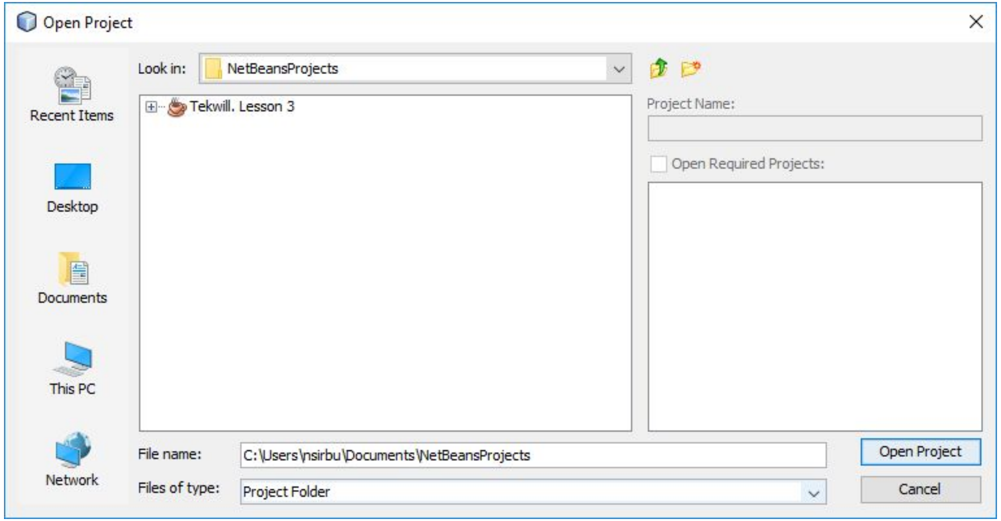
Created File: nsirbu\Documents\NetBeansProjects\Tekwill. Lesson 3\src\NameOfMyClass.java

Warning: It is highly recommended that you do not place Java classes in the default package.

Opening an Existing Java Project

If you ever need to open an existing project, perform the following steps:

1. Select **File** > **Open Project**.
2. Navigate to the directory that contains your projects.
3. Select the project file you want.
4. Click **Open Project**.



Look in: NetBeansProjects

Tekwill. Lesson 3

Project Name:

□ Open Required Projects:

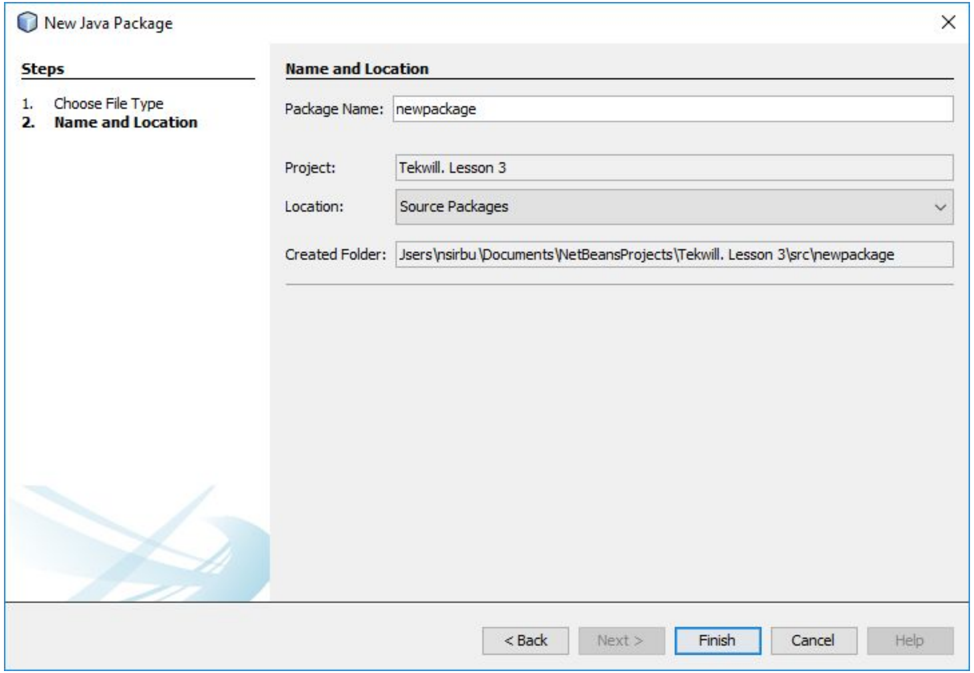
File name: C:\Users\nsirbu\Documents\NetBeansProjects

Files of type: Project Folder

Creating a New Java Package

If you ever need to create a new package, perform the following steps:

1. Right-click your project.
2. Select **New** > **Java Package**.
3. Name the package.
4. Click **Finish**.



New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Package Name: newpackage

Project: Tekwill. Lesson 3

Location: Source Packages

Created Folder: Jsers\nsirbu\Documents\NetBeansProjects\Tekwill. Lesson 3\src\newpackage

Resources

An Introduction to Programming Paradigms

(https://digitalfellows.commons.gc.cuny.edu/2018/03/12/an-introduction-to-programming-paradigms/)

How is Java platform independent?

(https://www.geeksforgeeks.org/java-platform-independent/)

Java Fundamentals

Lesson 1: Java programming

End.

Speaker: Nicolae Sîrbu
Alexandru Umaneț