



# 01 – Java platforma, sąvokos

Jaroslav Grablevski

# Turinys

- Apie dalyką**
- Javos istorija**
- Kodėl java?**
- Sąvokos (JRE, JDK, JVM, bytecode...)**
- Integruotos kūrimo aplinkos**
- Pirma programa**



# Pagrindinės dalyko temos

- **Java programavimo aplinka**
  - **sąvokos**
  - **instaliavimas**
  - **IDE**
- **Java pagrindai**
  - **java kalbos sintaksė**
  - **kintamieji, duomenų tipai, operatoriai**
  - **loginiai operatoriai, if, switch išraiškos**
  - **ciklai (for, while)**
  - **duomenų struktūros, algoritmai**
  - **masyvai, ArrayList**



# Pagrindinės dalyko temos

- **Objektinis programavimas**
- **Programų testavimas ir derinimas**
- **Objektinio programavimo principų taikymas**
- **Darbas su duomenimis**
- **Įvadas į funkcinį programavimą**
- **Kodavimo standartai**



# Istorija



# Java versijos (3)

Version	Release date	End of Free Public Updates <small>[3][8][9][10]</small>	Extended Support Until
JDK Beta	1995	?	?
JDK 1.0	January 1996	?	?
JDK 1.1	February 1997	?	?
J2SE 1.2	December 1998	September 2003	?
J2SE 1.3	May 2000	?	?
J2SE 1.4	February 2002	October 2008	February 2013
Java SE 5	September 2004	November 2009	April 2015
Java SE 6	December 2006	April 2013	December 2018 December 2026 for Azul <small>[11]</small>
Java SE 7	July 2011	July 2019	July 2022
Java SE 8 (LTS)	March 2014	<b>March 2022 for Oracle (commercial)</b> December 2030 for Oracle (non-commercial) December 2030 for Azul May 2026 for IBM Semeru <small>[12]</small> At least May 2026 for Eclipse Adoptium At least May 2026 for Amazon Corretto	December 2030 <small>[13]</small>
Java SE 9	September 2017	March 2018 for OpenJDK	—
Java SE 10	March 2018	September 2018 for OpenJDK	—



# Java versijos (4)

Java SE 11 (LTS)	55	25th September 2018	(OpenJDK currently maintained by Red Hat) <sup>[18]</sup> October 2024 for IBM Semeru <sup>[13]</sup> At least September 2024 for Eclipse Adoptium <sup>[9]</sup> and Microsoft <sup>[19][14]</sup> At least September 2027 for Amazon Corretto <sup>[10]</sup>	January 2032 for Oracle <sup>[8]</sup> September 2026 for Azul <sup>[11]</sup> October 2024 for Red Hat <sup>[12]</sup>
Java SE 12	56	19th March 2019	September 2019	—
Java SE 13	57	17th September 2019	March 2020	—
Java SE 14	58	17th March 2020	September 2020	—
Java SE 15	59	16th September 2020	March 2021	—
Java SE 16	60	16th March 2021	September 2021	—
Java SE 17 (LTS)	61	14th September 2021	(OpenJDK currently maintained by SAP) <sup>[20]</sup> October 2027 for IBM Semeru <sup>[13]</sup> At least September 2027 for Microsoft <sup>[14]</sup> and Eclipse Adoptium <sup>[9]</sup> At least September 2028 for Amazon Corretto <sup>[21]</sup>	September 2029 or later for Oracle <sup>[8]</sup> and Azul <sup>[11]</sup> October 2027 for Red Hat <sup>[12]</sup>
Java SE 18	62	22nd March 2022	September 2022 for OpenJDK and Adoptium	—
Java SE 19	63	20th September 2022	March 2023	—
Java SE 20	64	21st March 2023	September 2023	—
Java SE 21 (LTS)	65	19th September 2023	September 2028 At least September 2030 for Amazon Corretto <sup>[21]</sup>	September 2031 for Oracle <sup>[8]</sup> and Azul
Java SE 22	66	March 2024	September 2024 <sup>[15]</sup>	—

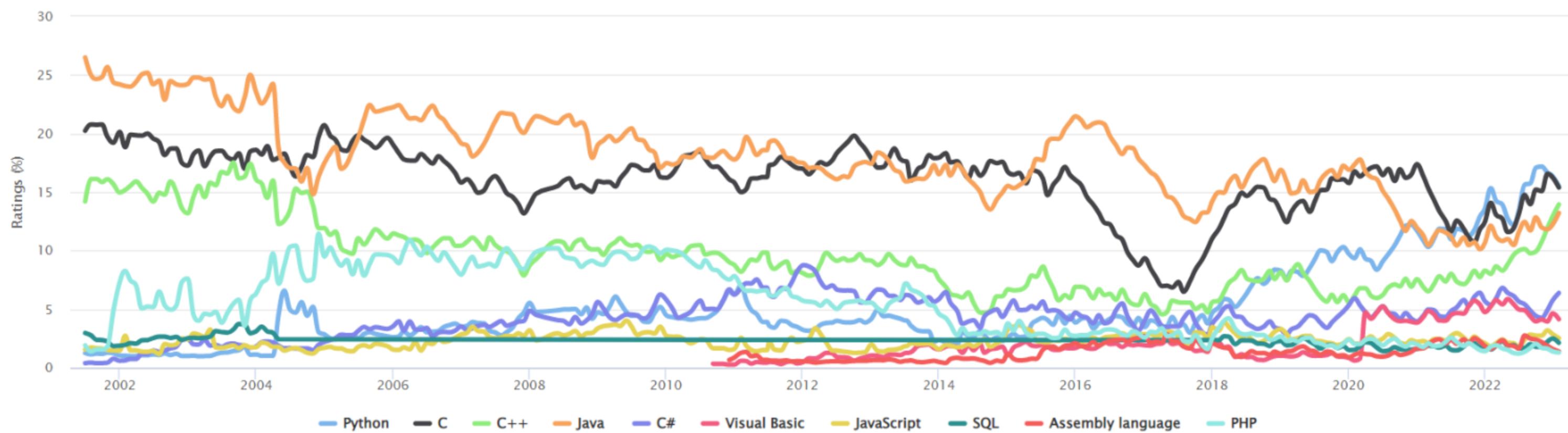


# Koděl JAVA



TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)

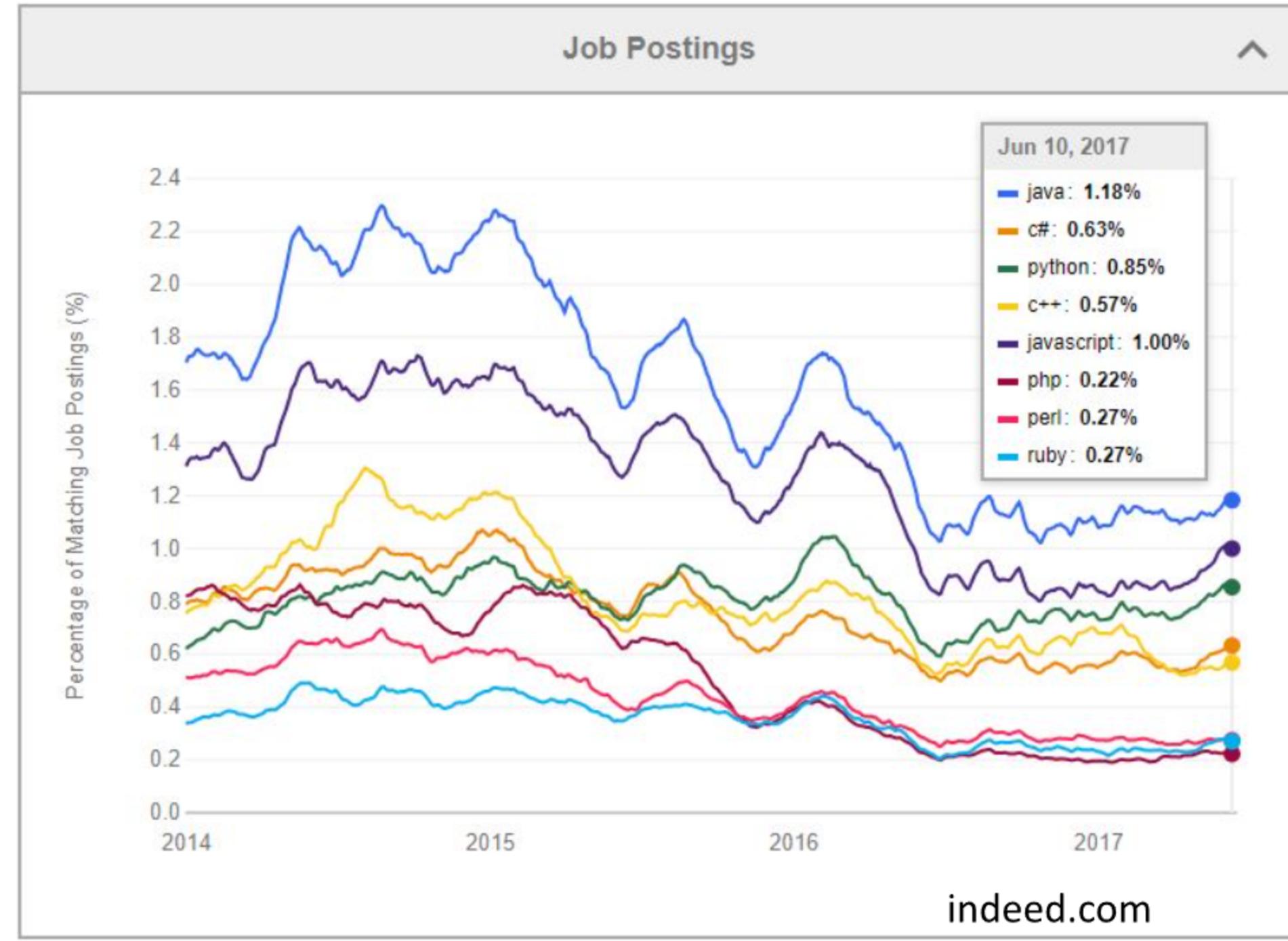


# Koděl JAVA

Oct 2023	Oct 2022	Change	Programming Language	Ratings	Change
1	1		 Python	14.82%	-2.25%
2	2		 C	12.08%	-3.13%
3	4		 C++	10.67%	+0.74%
4	3		 Java	8.92%	-3.92%
5	5		 C#	7.71%	+3.29%
6	7		 JavaScript	2.91%	+0.17%
7	6		 Visual Basic	2.13%	-1.82%
8	9		 PHP	1.90%	-0.14%
9	10		 SQL	1.78%	+0.00%
10	8		 Assembly language	1.64%	-0.75%



# Kodel JAVA



# Kodēt JAVA

MAJOR COMPANIES  
THAT USE



# JAVA testavime



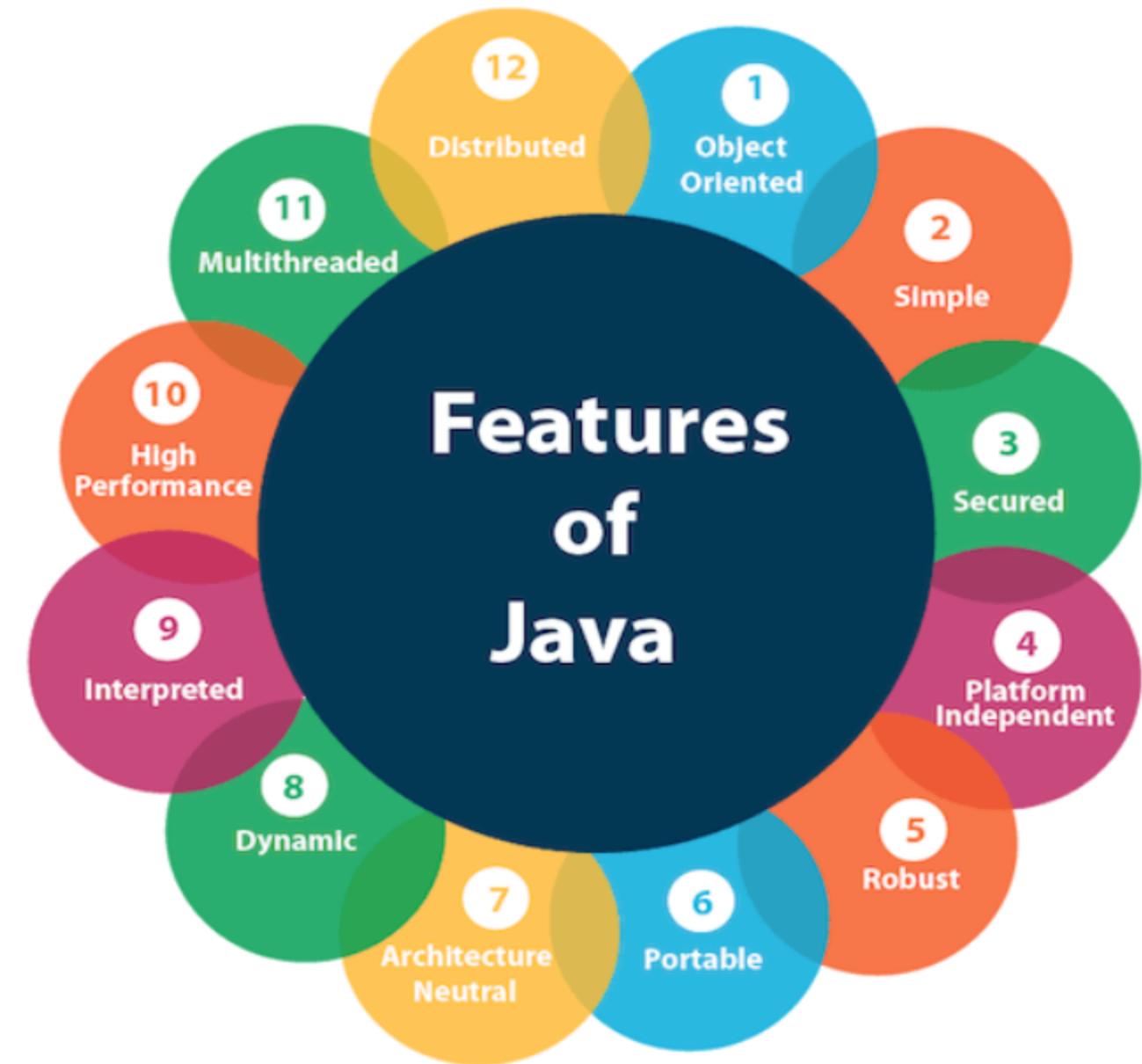
**JUnit**

**REST ASSURED**



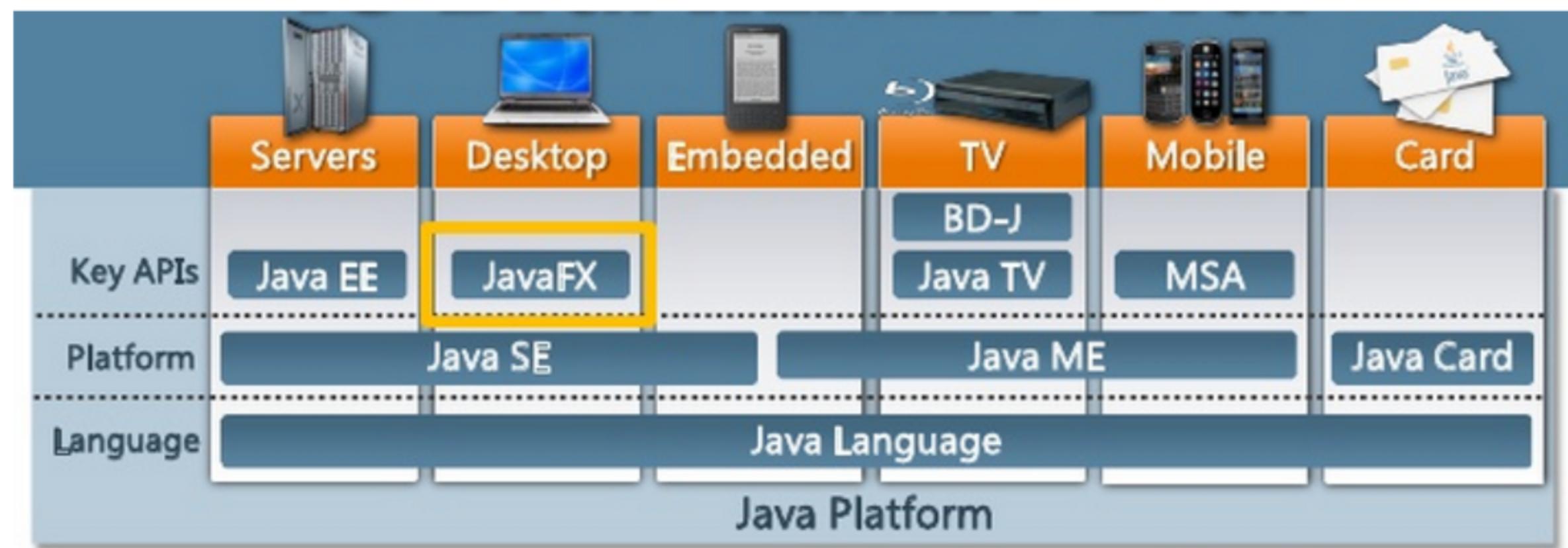
# Kodėl JAVA

- Paprasta
- Nepriklausoma nuo kompiuterių architektūros
- Objektiškai orientuota
- Portatyvi
- Paskirstyta
- Naši
- Daugiagijė
- Patvari
- Dinamiška
- Saugi

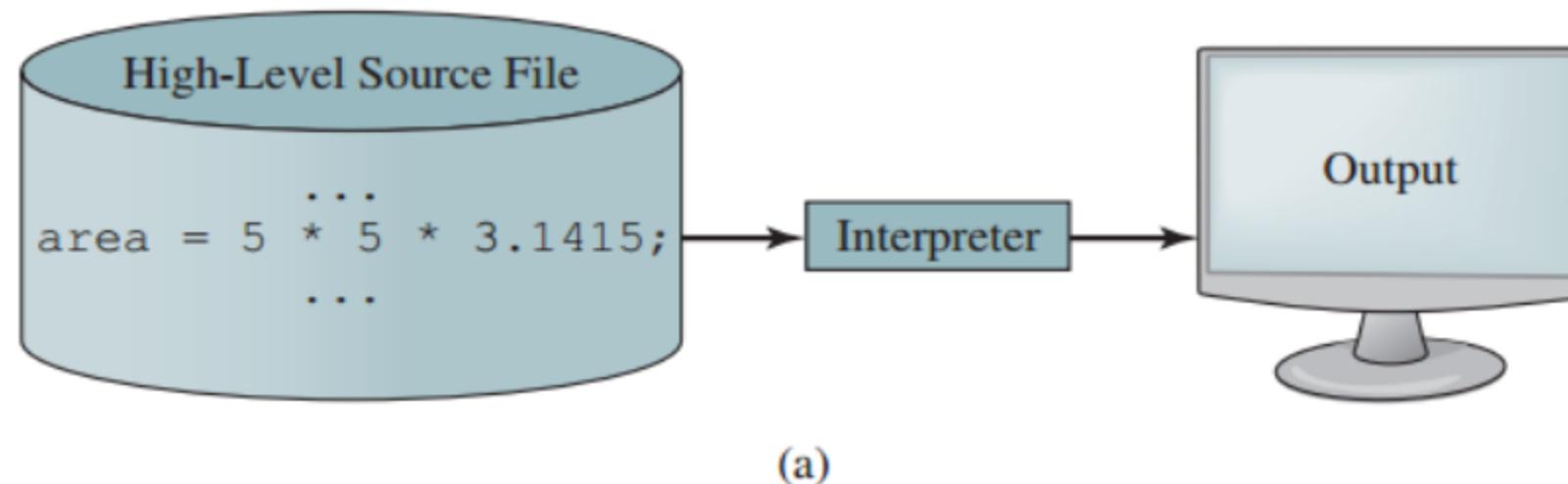


# Java redakcijos

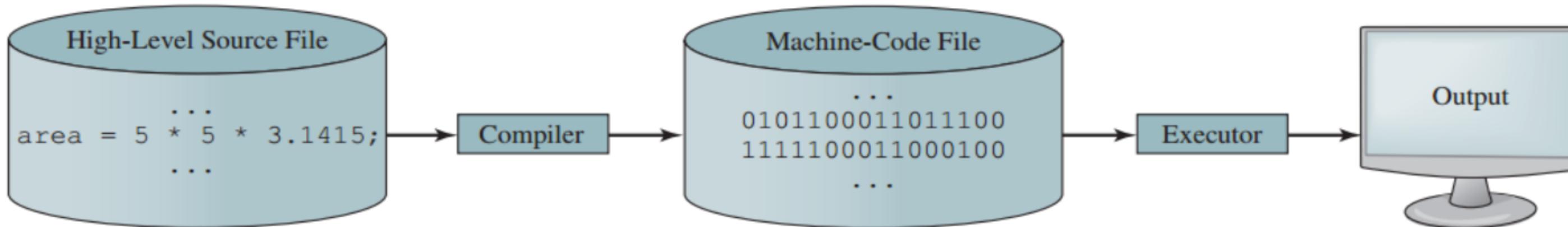
- Java SE - Standart Edition
- Java EE - Enterprise Edition
- Java ME - Micro Edition



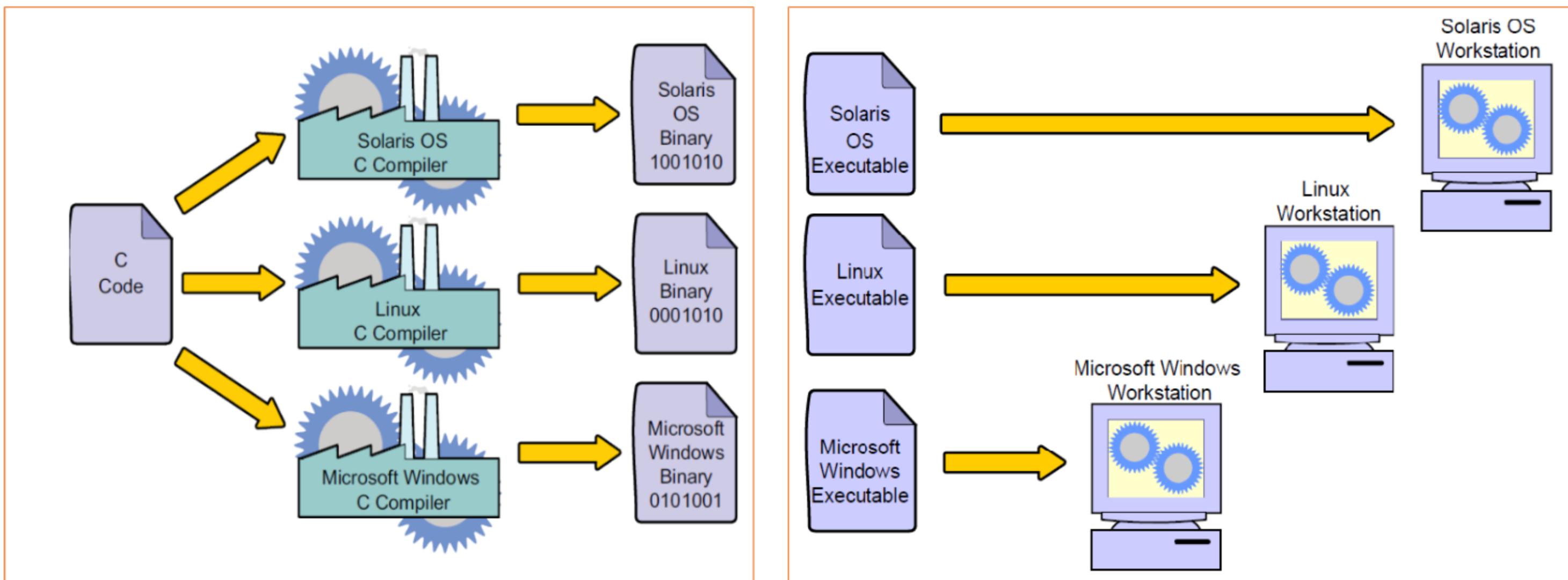
# interpreter vs compiler



(a)

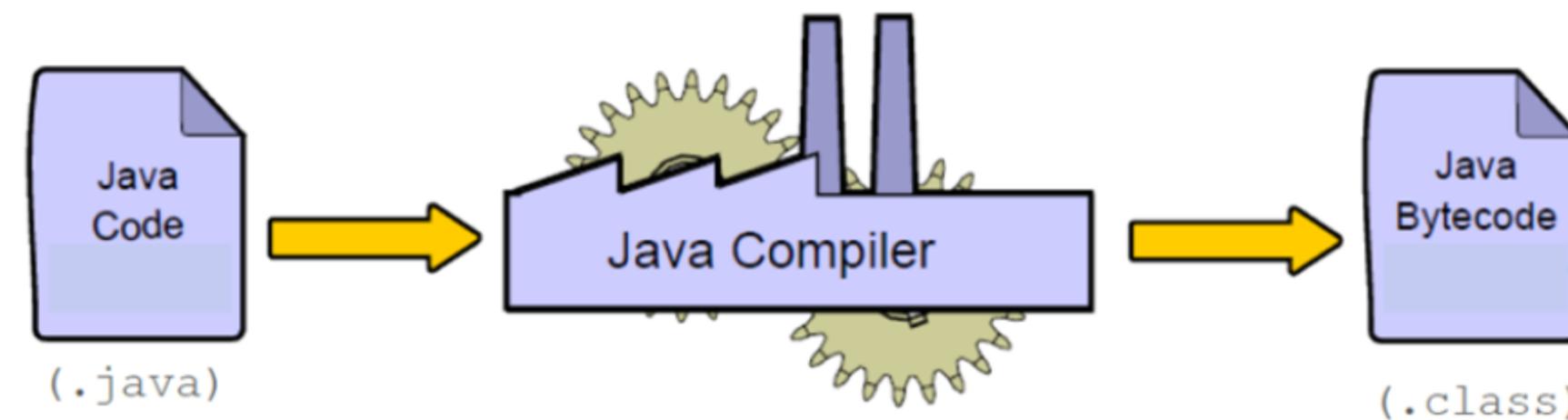


# C programos kompiliavimas



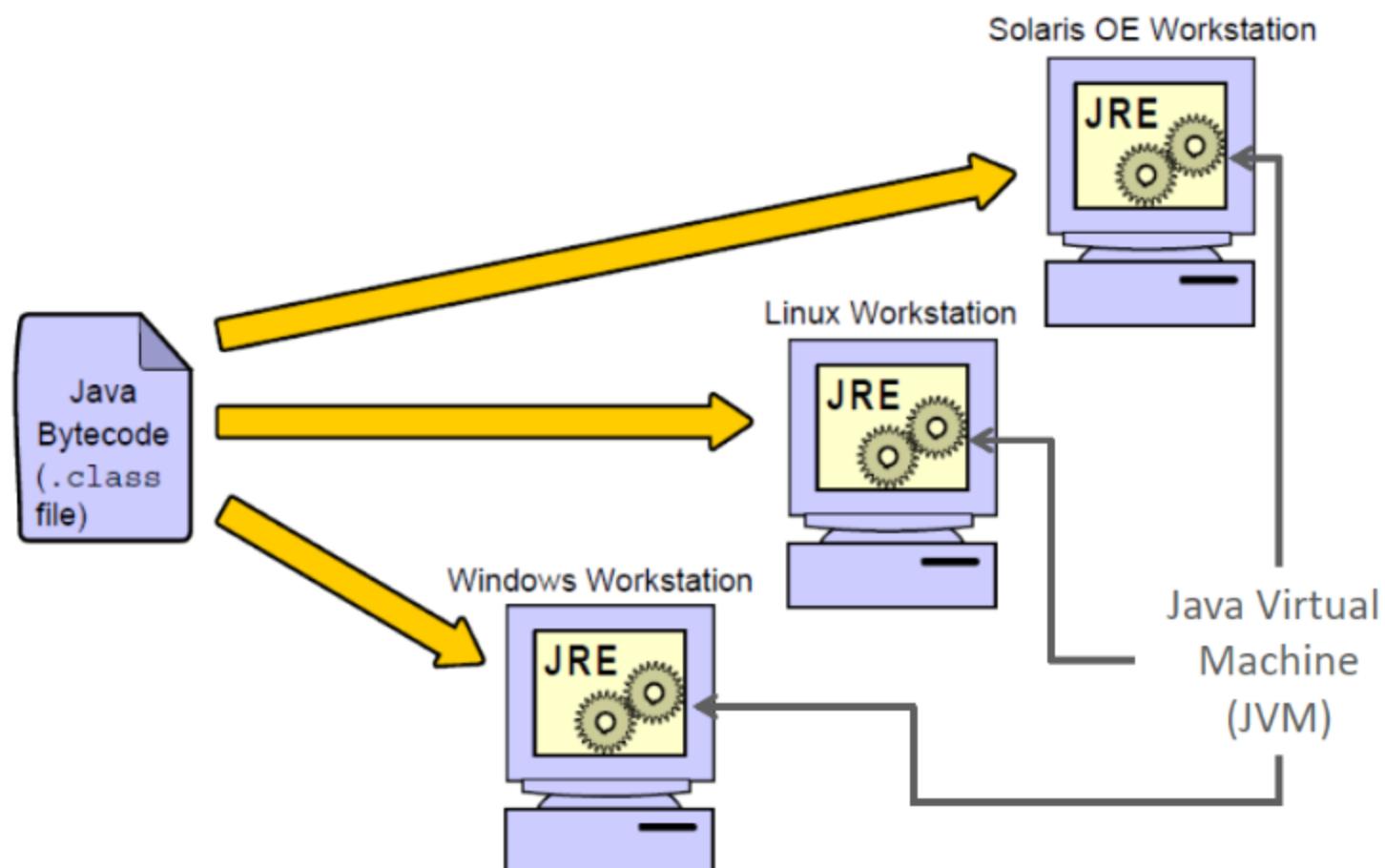
# Bytecode

- Java programa kompiliuojant paverčiama į JVM komandas, o ne konkretaus kompiuterio procesoriaus komandas.
- Sukompiliuota Java programa (bytecode) vienodai veiks ant visų kompiuterių, kur yra įdiegta JVM
- JVM atlieka baitinio kodo interpretavimą (vykdymą)



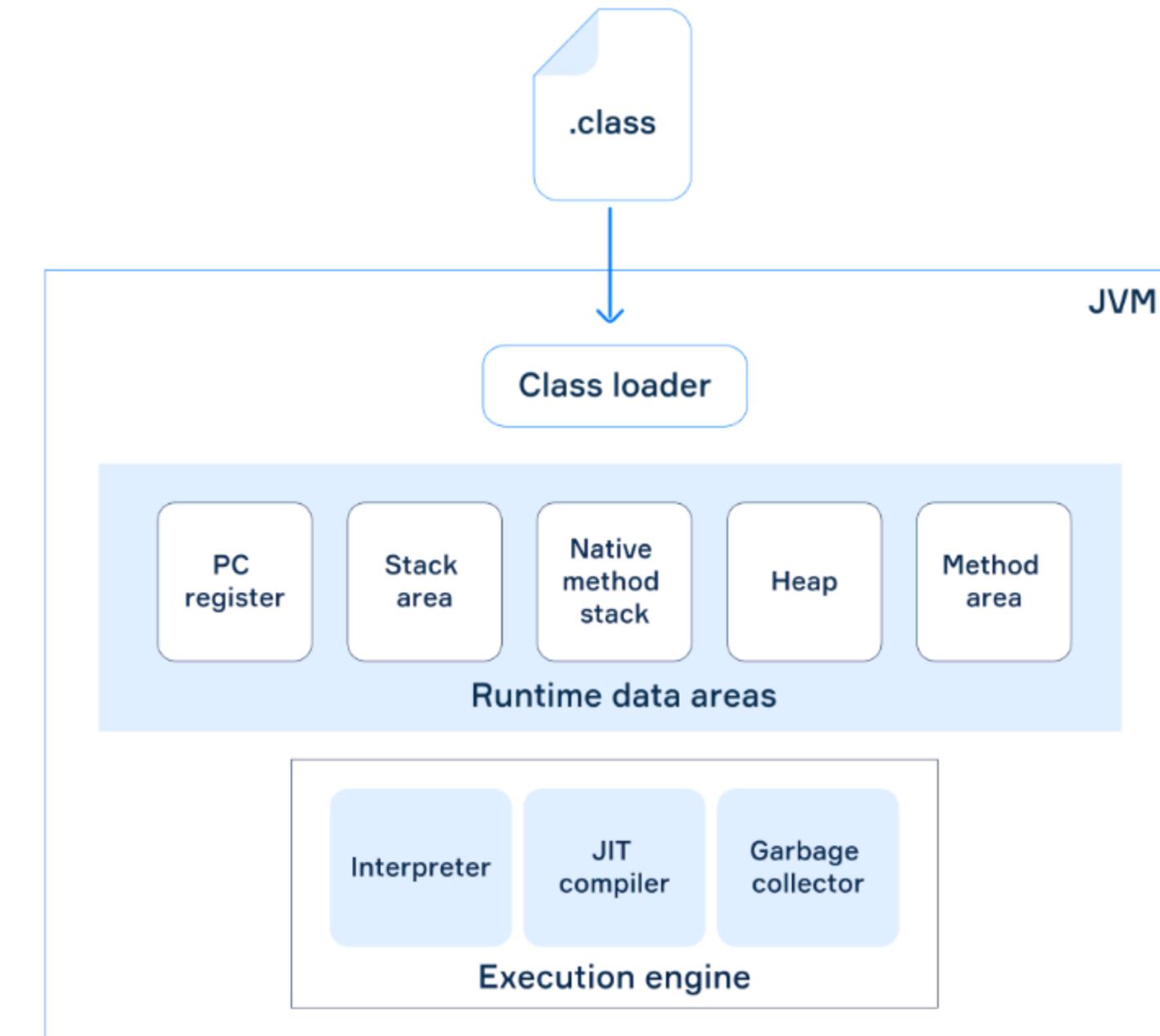
# Virtuali mašina (JVM)

- Java virtual machine
- Vykdo Java kodą
- Optimizuojas esamai sistemai
- Abstrakcija užtikrinanti programų nepriklausomumą nuo platformos

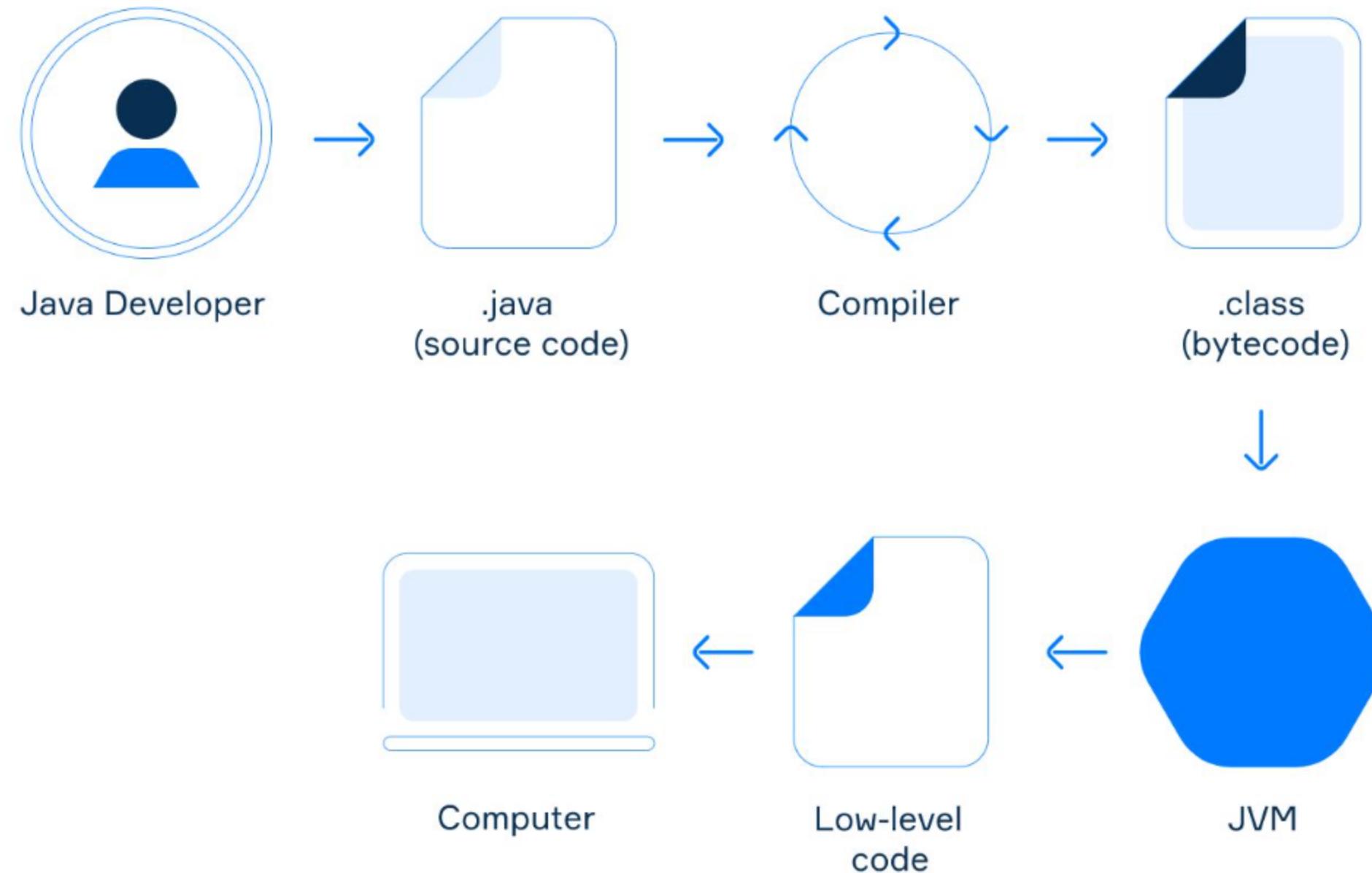


# JVM architektūra

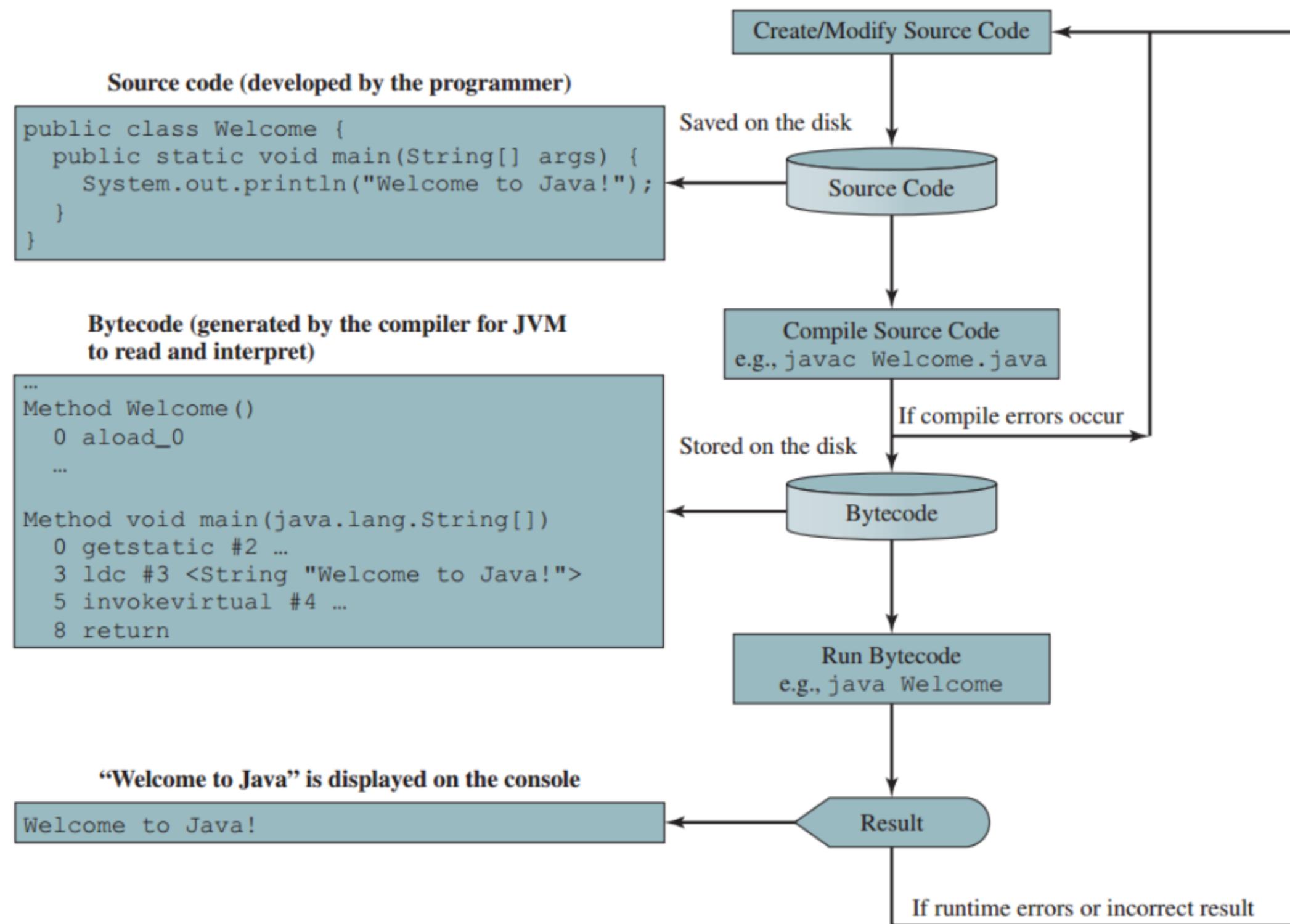
- **Class loader** - atsakinga už dinaminį Java klasų įkėlimą į JVM programos vykdymo metu
- **Runtime memory/data area** - atsakinga už duomenų, reikalingų programos vykdymui, valdymą
- **Execution engine** - atsakinga už baitkodo vykdymą



# Writing, compiling and running Java programs



# Writing, compiling and running Java programs



# Java multiplatform (WORA)

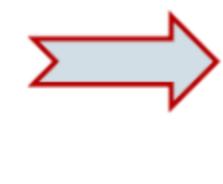
- Write once:



Java Source file

```
class Test{  
    public int foo(int i){  
        i=i*i;  
        return i;  
    } //end method foo  
} //end class Test
```

ORACLE®



javac compiler



Class file

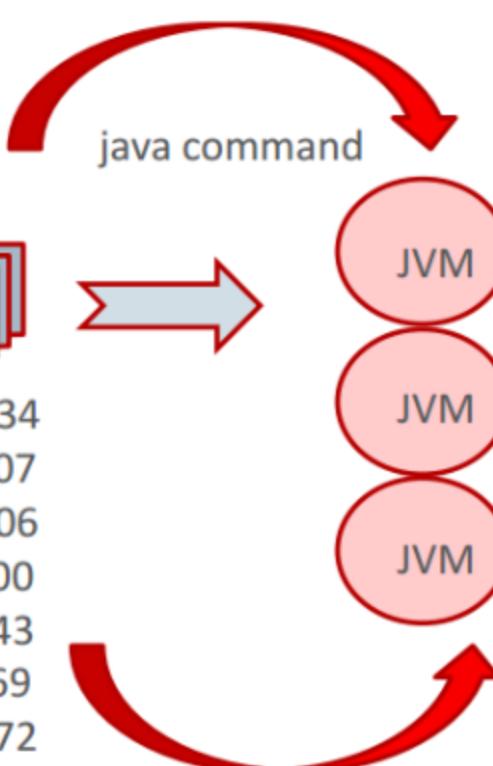
CA FE BA BE 00 00 00 34 00 0F 0A  
00 03 00 0C 07 00 0D 07 00 0E 01  
00 06 3C 69 6E 69 74 3E 01 00 03  
28 29 56 01 00 04 43 6F 64 65 01  
00 0F 4C 69 6E 65 4E 75 6D 62 65  
72 54 61 62 6C 65 01 00 03 66 6F  
6F 01 00 04 28 4900 07 00 00 00  
0A 00 02 00 00 00 03 00 04 00 04  
00 01 00 0A 00 00 00 02 00 0B

- Run anywhere:

Class file



CA FE BA BE 00 00 00 34  
00 0F 0A 00 03 00 0C 07  
00 0D 07 00 0E 01 00 06  
3C 69 6E 69 74 3E 01 00  
03 28 29 56 01 00 04 43  
6F 64 65 01 00 0F 4C 69  
6E 65 4E 75 6D 62 65 72



# JDK

- Norint sukompliuoti ir vykdyti Java kodą, reikia įsidiegti Java Development Kit (JDK).
- Pagrindinė JDK komanda yra javac, kuri kompliuoja Java kodą

```
C:\javaworks\helloworld>javac helloworld.java
```



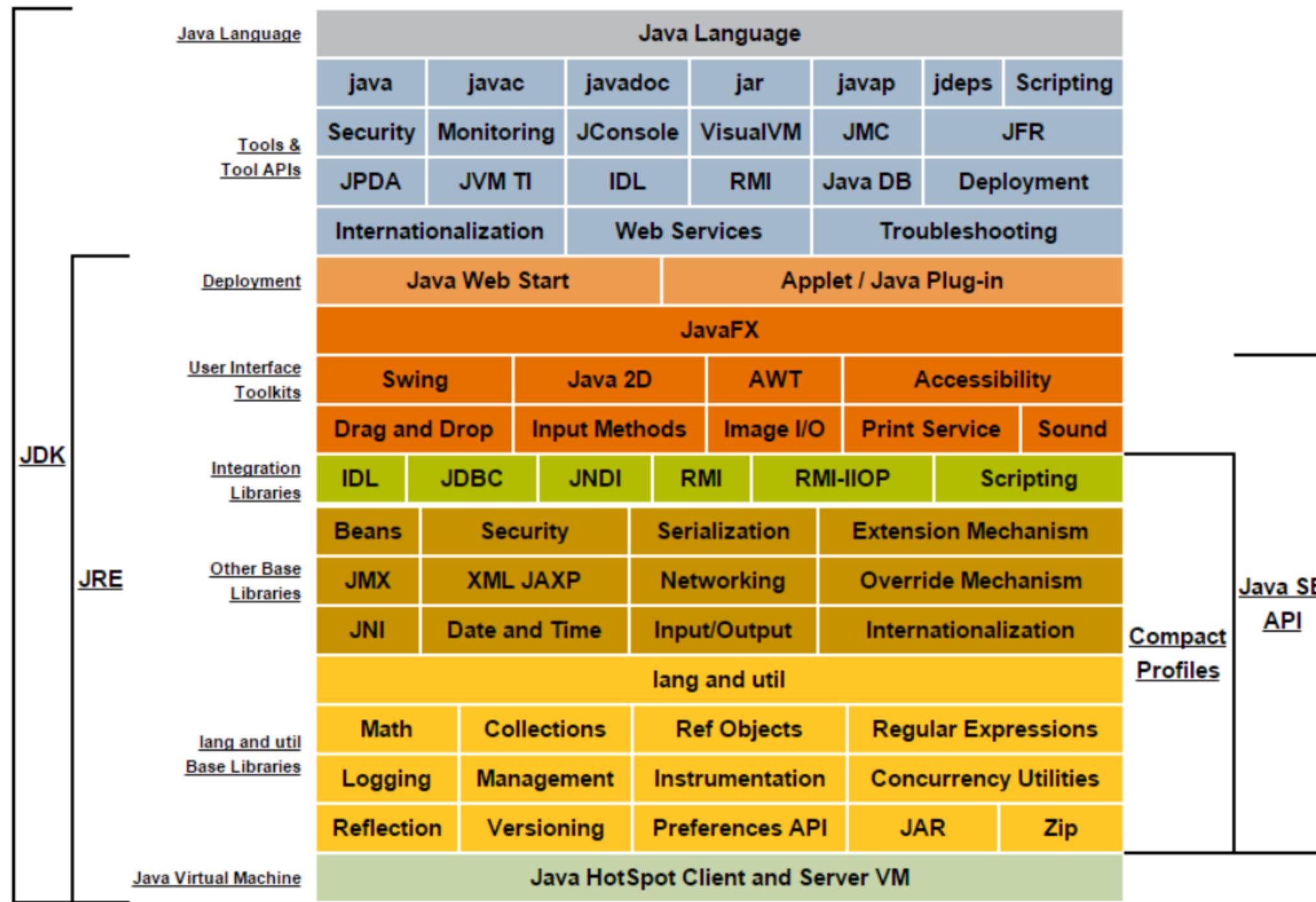
# JRE

- Java runtime environment (JRE) sudaro Java Virtuali Mašina(JVM), java platformos bazinės klasės ir pagalbinės Java platformos bibliotekos.
- JRE įeina į naršykles, operacines sistemas, duomenų bazių sistemas ir pan.  
Tiksliau, anksčiau buvo naršyklėse (vadinosi applets), tačiau dėl saugumo ši funkcija buvo pašalinta
- Pagrindinė JRE komanda yra **java**, kuri atlieka baitinio kodo vykdymą (interpretavimą):

```
C:\javaworks\helloworld>java helloworld  
Hello World!
```



# JRE yra JDK poaibis



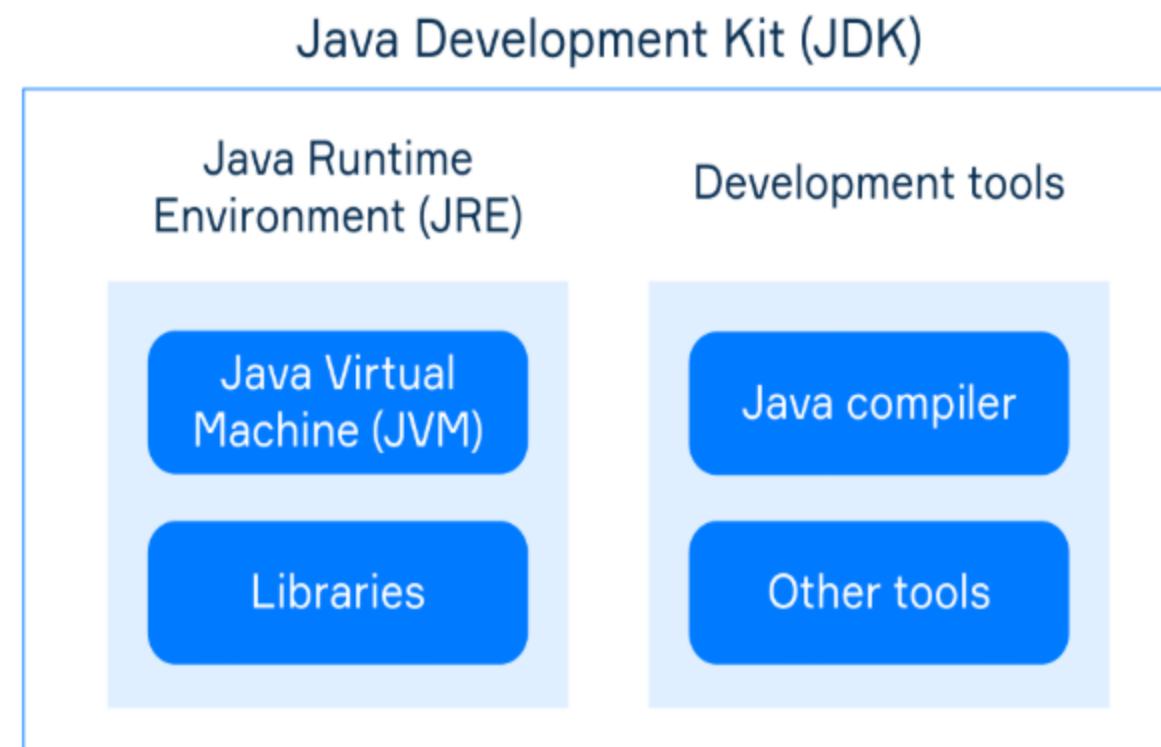
# JDK tools

- **java**: You can use the java command to launch a Java application.
- **javac**: You can use the javac tool and its options to read Java class and interface definitions and compile them into bytecode and class files.
- **javap**: You use the javap command to disassemble one or more class files.
- **javadoc**: You use the javadoc tool and its options to generate HTML pages of API documentation from Java source files.
- **jar**: You can use the jar command to create an archive for classes and resources, and to manipul
- **jdb**: You use the jdb command and its options to find and fix bugs in Java platform programs.ate or restore individual classes or resources from an archive.
- ...



# Recap

- **JVM** executes Java bytecode;
- **JRE** includes JVM and standard libraries: it is needed to run compiled programs;
- **JDK** includes JRE and development tools: it is needed to develop programs. As a developer, you need to install JDK.



# JDK vendors



GraalVM™



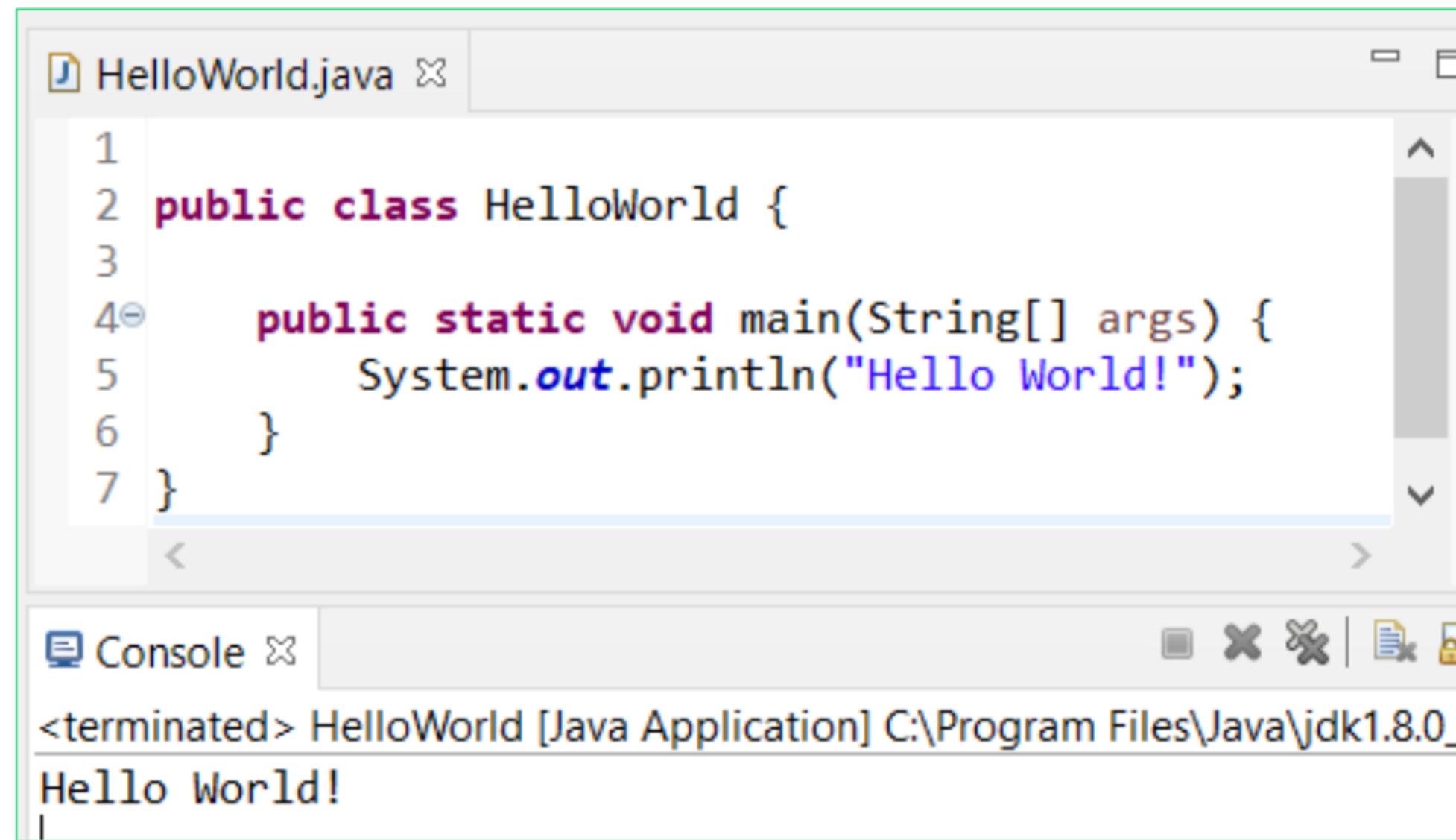
OpenJ9

OpenJDK

▪▪▪ AdoptOpenJDK



# Kaip atrodo Java kalba?



The screenshot shows a Java development environment with two windows:

- Code Editor (HelloWorld.java):** Displays the following Java code:

```
1
2 public class HelloWorld {
3
4     public static void main(String[] args) {
5         System.out.println("Hello World!");
6     }
7 }
```
- Console Window:** Displays the output of the program:

```
<terminated> HelloWorld [Java Application] C:\Program Files\Java\jdk1.8.0_
Hello World!
```



# Kodą galima rašyti taip...

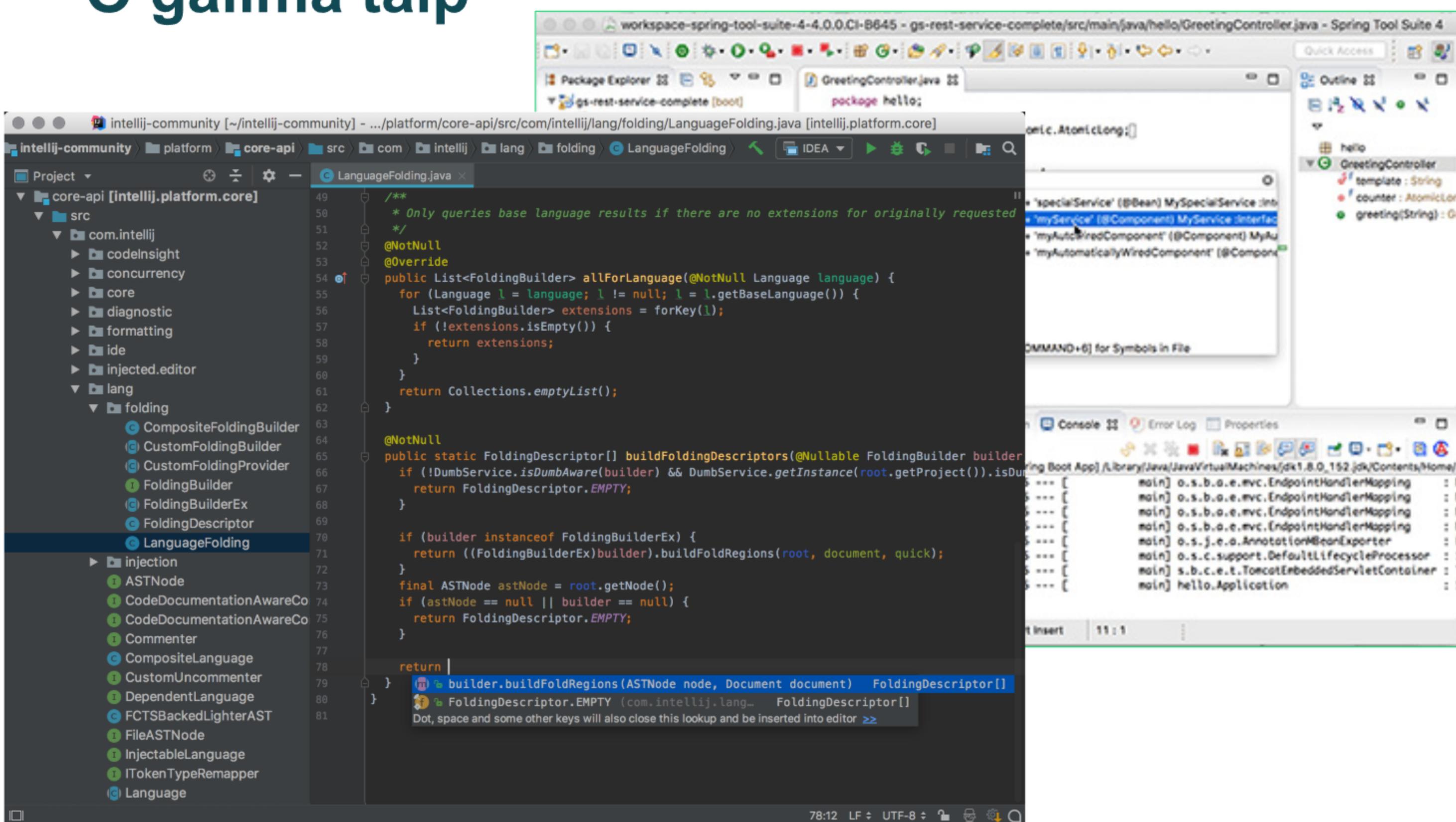


Untitled - Notepad

```
File Edit Format View Help
```

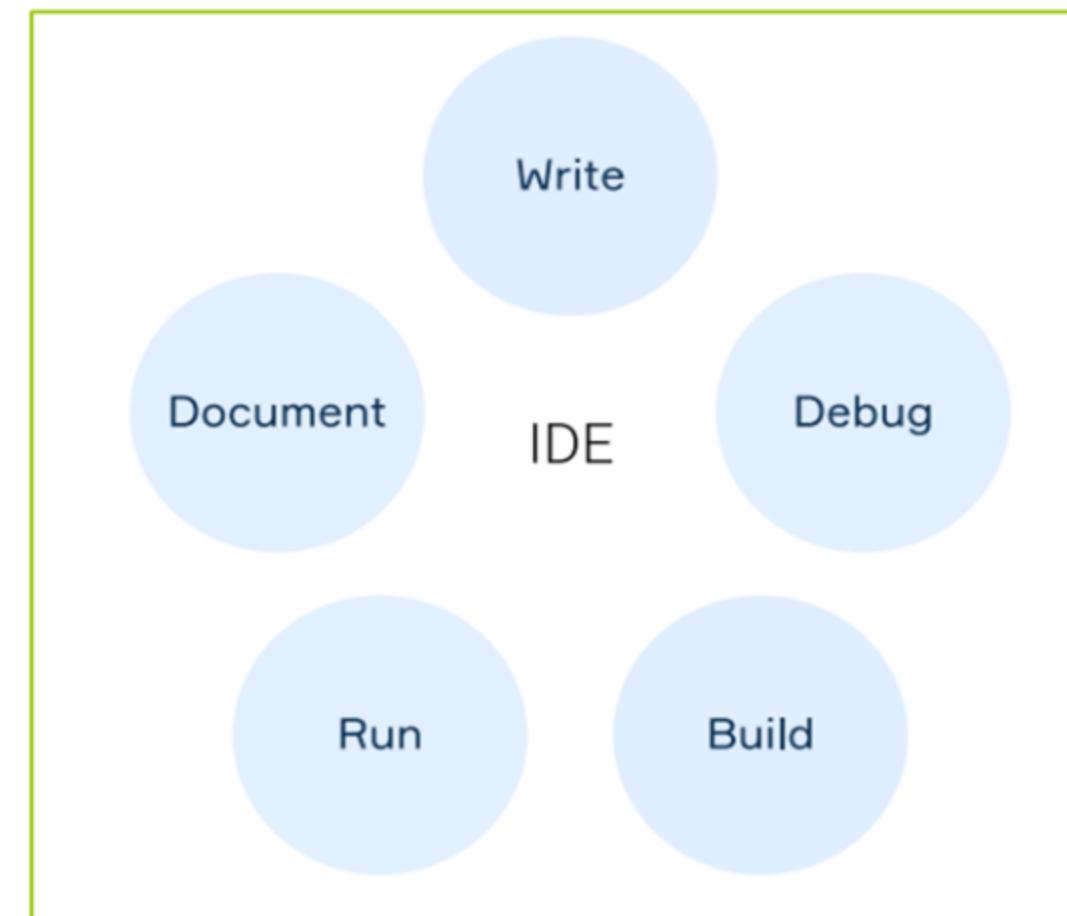
```
public class MyJavaProgram {  
    public static void main(String args[]){  
        System.out.println("This is a simple java program.");  
    }  
}
```

# O galima taip



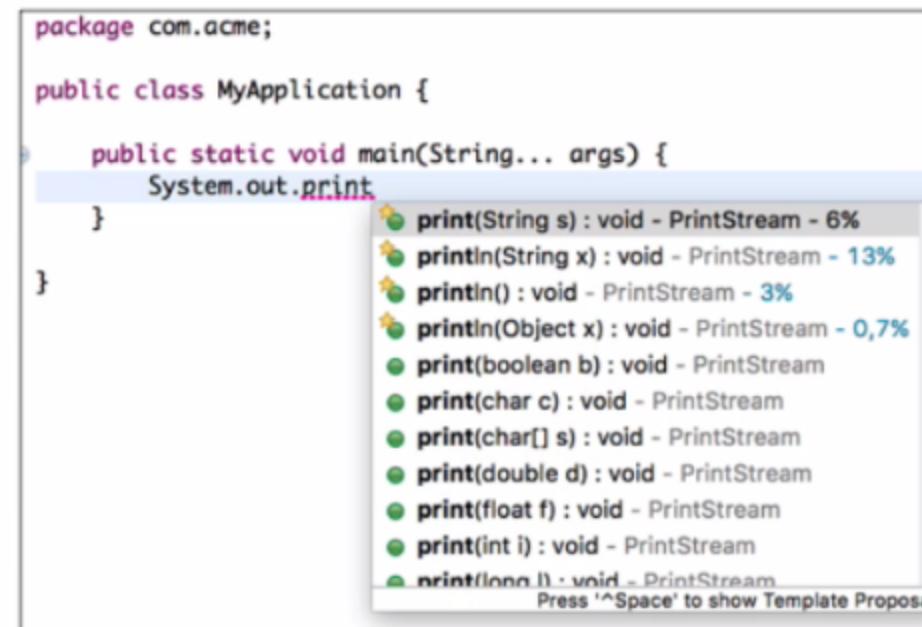
# IDE (Integruotos kūrimo aplinkos)

- Integrated Development Environment
- Palengvina kūrimo procesą
- Palengvina darbą su kodu (užuominos, šablonai, informacija apie klaidas...)
- Palengvina darbą su dideliais projektais
- Integruoja daug įrankių į vieną paketą (kompiliatorių, surinkėją, derintoją, versijų valdymo sistemą, kodo kokybės inspektavimo įrankius, ...)

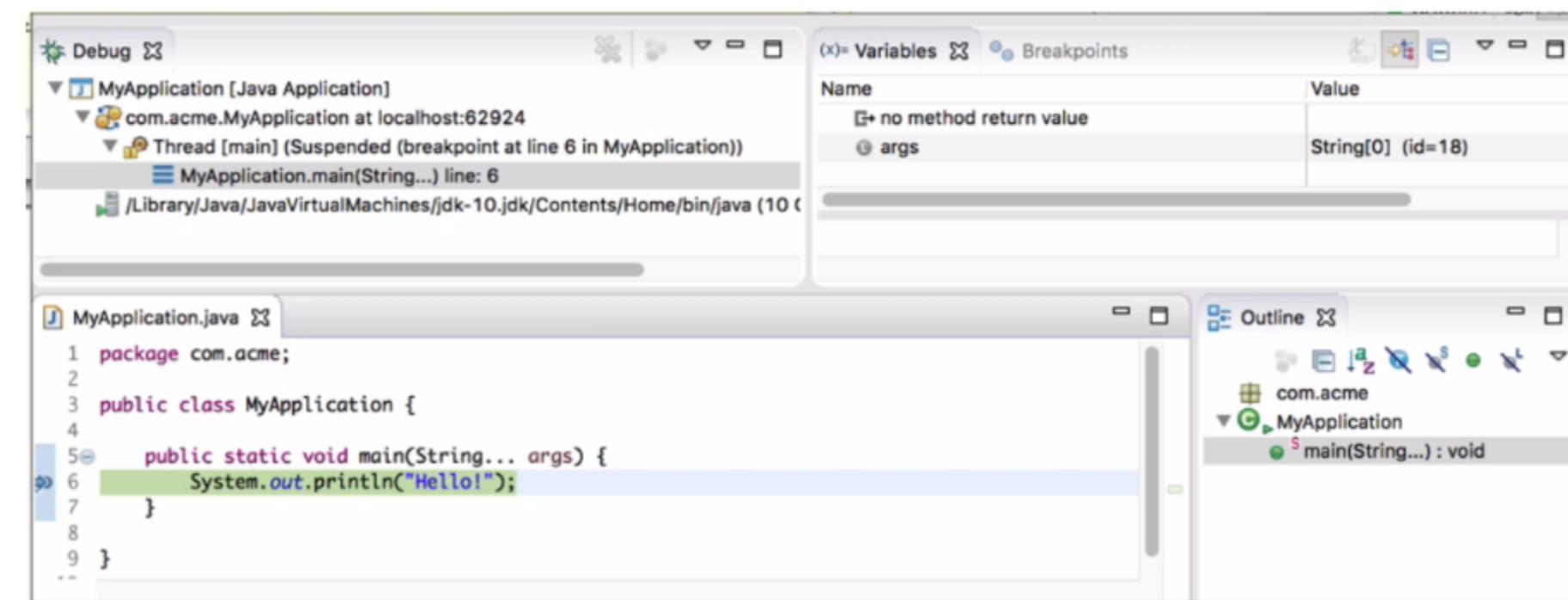
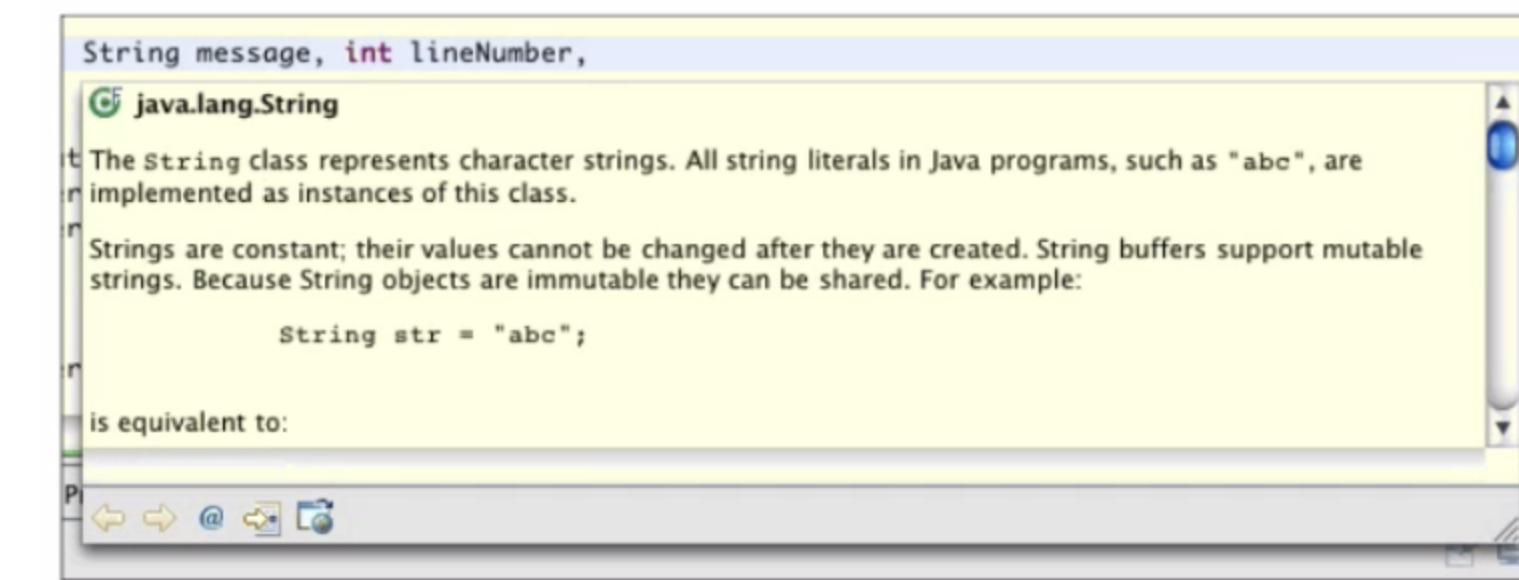


# IDE (Integruotos kūrimo aplinkos)

## Code completion



## Inline documentation



## Debugging

# IDE (Integruotos kūrimo aplinkos)



Eclipse

IntelliJ IDEA

NetBeans

BlueJ

JDeveloper

DrJava

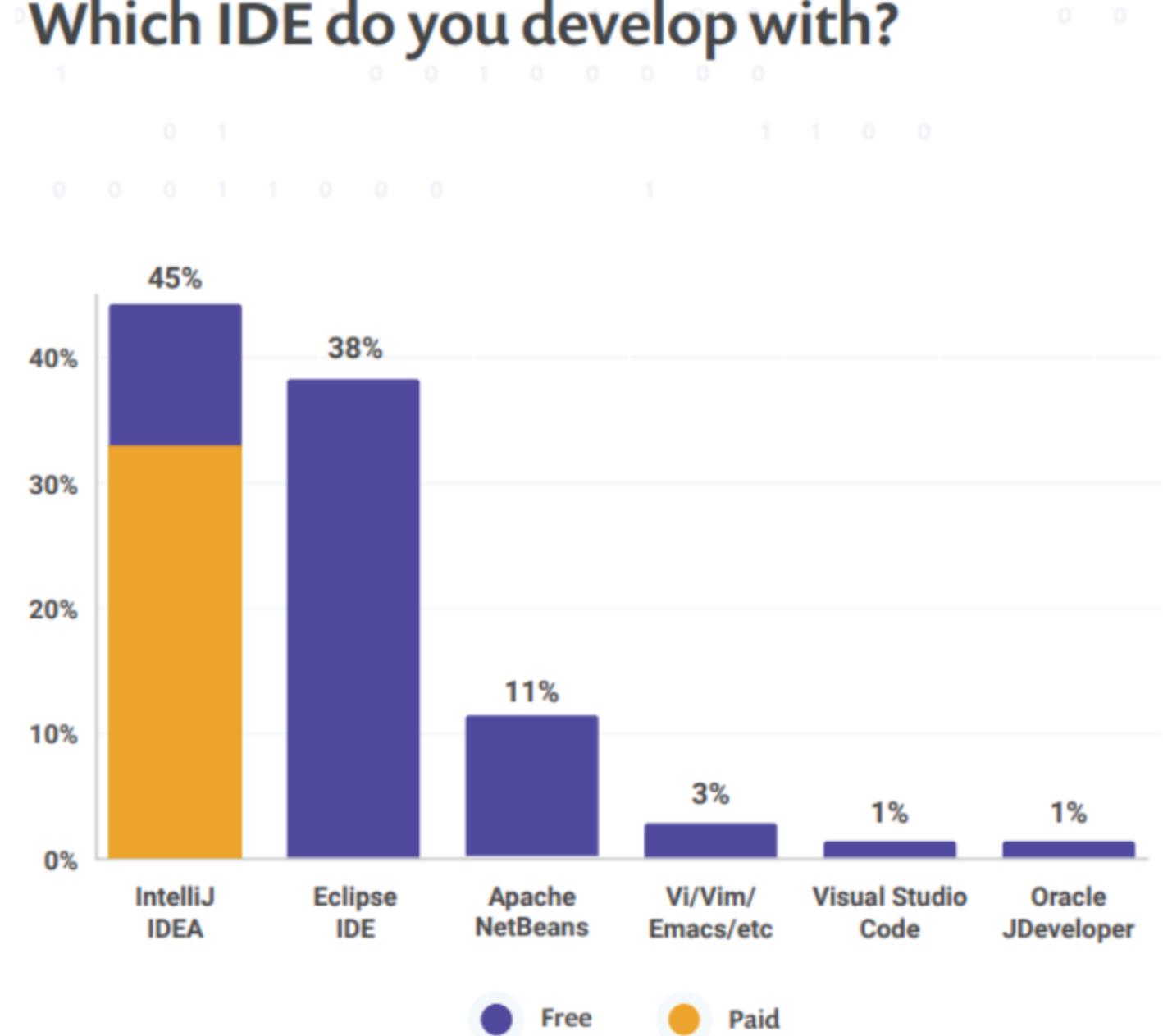
Android Studio

Other



# IDE (Integruotos kūrimo aplinkos)

Which IDE do you develop with?



# Hello World | Java

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

- Kiekviena Java kodo eilutė turi būti parašyta klasėje.
- Mūsų atveju, klasė vardu **PirmojiPrograma** (būtinai iš didžiosios raidės)
- Klasės vardas sutampa su failovardu.
- Didžiosios ir mažosios raidės Java kalboje skiriasi (**case-sensitive**), t.y. *MyClass* yra ne tas pats kas *myclass* arba *Myclass*.



# Skliaustai

- Skliaustų pora programoje sudaro bloką, kuris sugrupuoja programas komponentus.

```
public class Welcome { ←  
    public static void main(String[] args) { ←  
        System.out.println("Welcome to Java!");  
    } ←  
} ←  
                                Method block  
                                Class block
```



# main metodus

```
public class PirmojiPrograma {  
  
    public static void main(String[] args) {  
  
        // Sentence 1  
        // Sentence 2  
        // ...  
        // Sentence n  
  
    }  
}
```

- Nuo pagrindinio metodo (**main**) pradedama vykdyti programa.
- Programos vykdymo metu bus vykdomos visos kodo eilutės, esančios main metode.



# Išvedimo sakinys (1)

```
public class PirmojiPrograma {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
        System.out.println("Labas pasauli!");  
        System.out.println("...123@#$%^");  
    }  
}
```

Hello World!  
Labas pasauli!  
...123@#\$%^

Kiekviena komanda/sakinys  
atskiriamas **kabliataškiu**.



# Išvedimo sakinys (2)

```
public class PirmojiPrograma {  
  
    public static void main(String[] args) {  
  
        System.out.println("Sakinys 1"); System.out.println("Sakinys 2");  
  
        System.out.println("=====");  
  
        System.out.print("Sakinys 3"); System.out.print("Sakinys 4");  
    }  
}
```

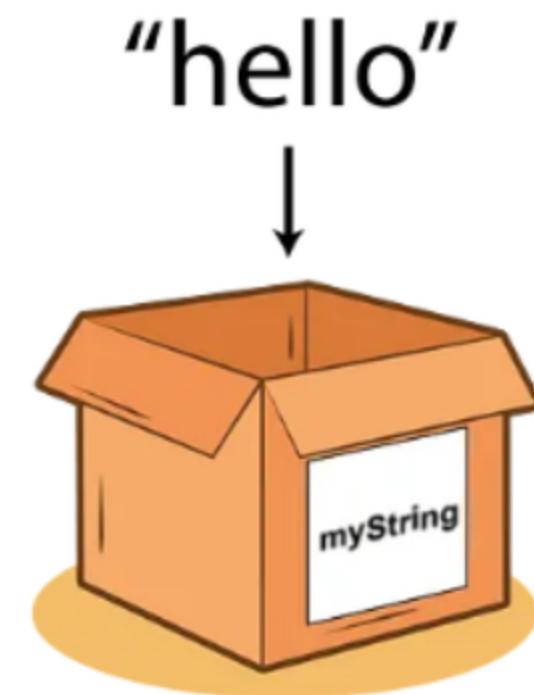
```
Sakinys 1  
Sakinys 2  
=====  
Sakinys 3Sakinys 4
```

System.out.println(); | System.out.print();



# Kintamasis

```
String myString = "hello";
```



- **Kintamasis tai vieta kompiuterio atmintyje, turinti vardą, kurioje saugomi duomenys, kol vykdoma programa. Vykstant programai paprastai tie duomenys yra keičiami.**



# Duomenų įvedimas

```
// Introduce the scanner tool used for reading user input
import java.util.Scanner;
public class Example {
    public static void main(String[] args) {
        // Create a tool for reading user input and name it scanner
        Scanner scanner = new Scanner(System.in);
        // Print "Write a message: "
        System.out.println("Write a message: ");
        // Read the string written by the user, and assign it
        // to program memory "String message = (string that was given as input)"
        String message = scanner.nextLine();
        // Print the message written by the user
        System.out.println(message);
    }
}
```



# String concatenation

```
String start = "My name is ";
String end = ", James Bond";
System.out.println(start + "Bond" + end);
```



# Savikontrolės klausimai

- Kuo Java skiriasi nuo kitų kalbų?**
- Kas yra WORA principas?**
- Kas yra JVM, JRE, JDK?**
- Kokias komandas naudojame norint sukompiliuoti ir paleisti java programą?**
- Kas yra baitkodas?**
- Kokie java išeities kodo (source code) ir baitkodo failų plėtiniai?**
- Kas yra IDE?**



# Instaliavimas

- Atsisiųskite ir instaliuokite JDK 17
  - <https://www.oracle.com/eg/java/technologies/downloads/#jdk17-windows>
- Pasitirkinkite komandinėje eilutėje ar pavyko
  - Komanda: javac –version

```
$ javac -version  
javac 17.0.6
```

- Atsisiųskite ir instaliuokite IntelliJ IDEA Community Edition
  - <https://www.jetbrains.com/idea/download/#section=windows>
- Parašykite ir paleiskite „Hello world!” programą

