

Intro

Intro

Problem

- PC understand: 0110010101001010.
- Human understand: Languages (English, Belarusian, German).

Solution

- Use Programing Languages.
- For example: Java.

Why Java?

Profit

- Объектно-ориентированное программирование.
- Автоматическое управление памятью.
- Многопоточность.
- Стабильность и огромное сообщество.

Profit

- Независимость от платформы («Написать один раз и использовать везде»).
- Стандарт для корпоративных вычислительных систем.
- Java — язык высокого уровня с простым синтаксисом и плавной кривой обучения.
- Безопасность.

Loss

- Низкая производительность по сравнению с C/C++.
- Отсутствие нативного дизайна (работа с GUI).
- Платная коммерческая поддержка.

Where use?

- Приложения для Android
- Финансовые программы
- Автоматизация тестирования
- Программные продукты
- Программы для работы с большими данными

Java

Java



- **Java** - объектно-ориентированный язык программирования
- разработан компанией **Sun Microsystems**
- в последующем приобретённой компанией **Oracle**

Author

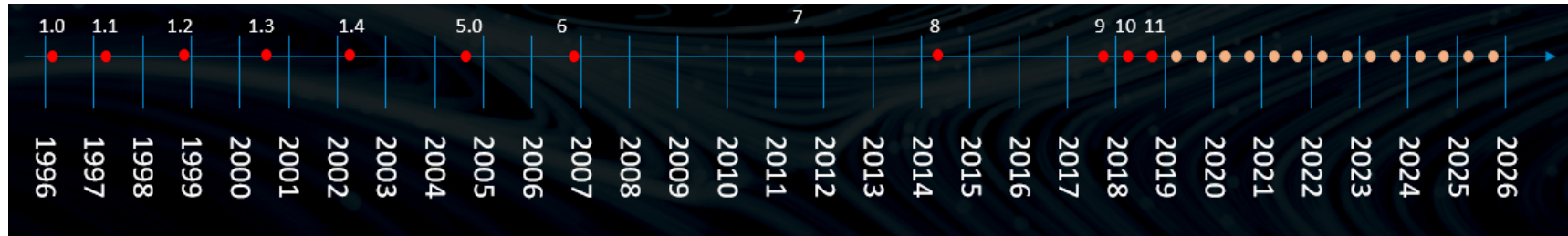


James Gosling

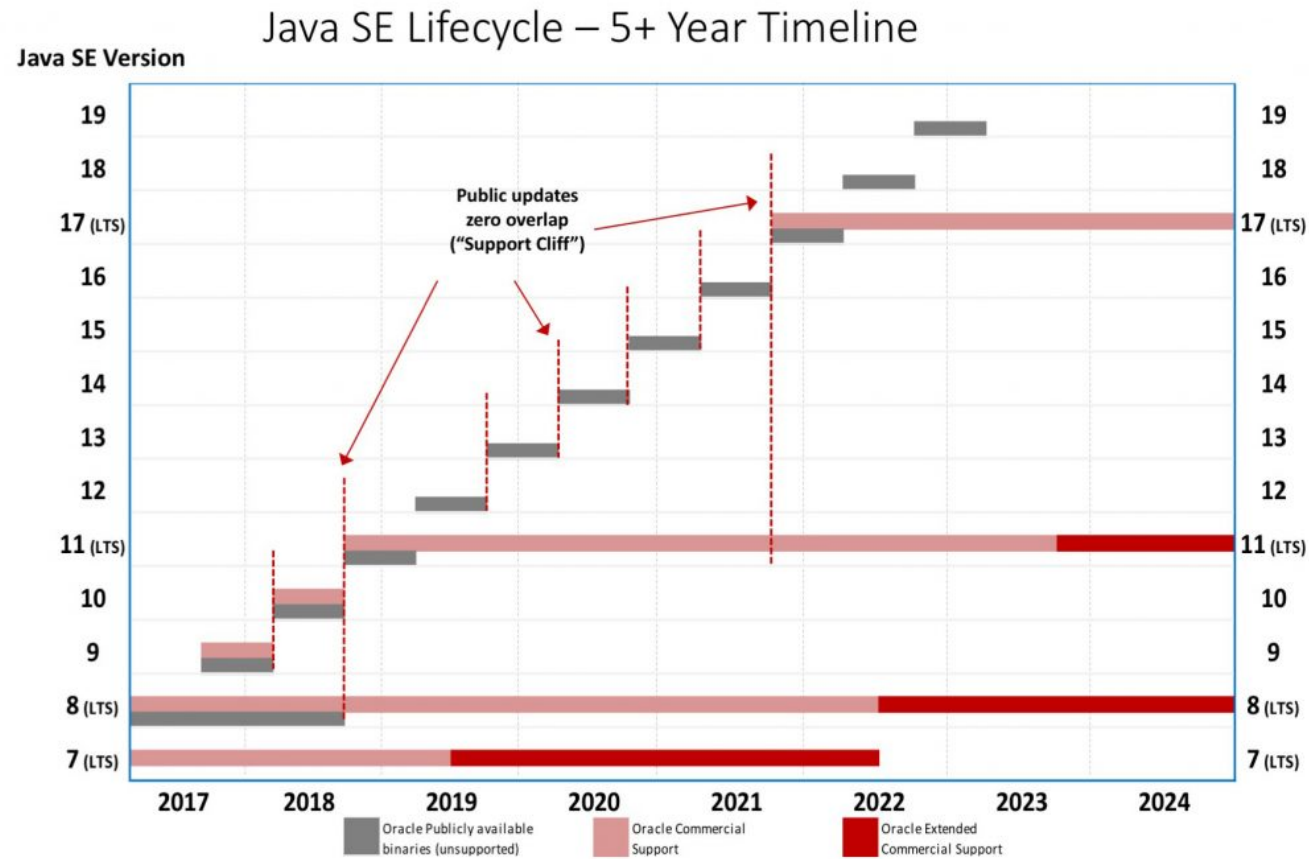
Java Platforms

- **Java Platform, Standard Edition (Java SE)**
- **Java Platform, Enterprise Edition (Java EE) → Jakarta EE**
- **Java Platform, Mobile Edition (Java ME)**
- **Java Card**
- **JavaFX → OpenJFX**

Java versions

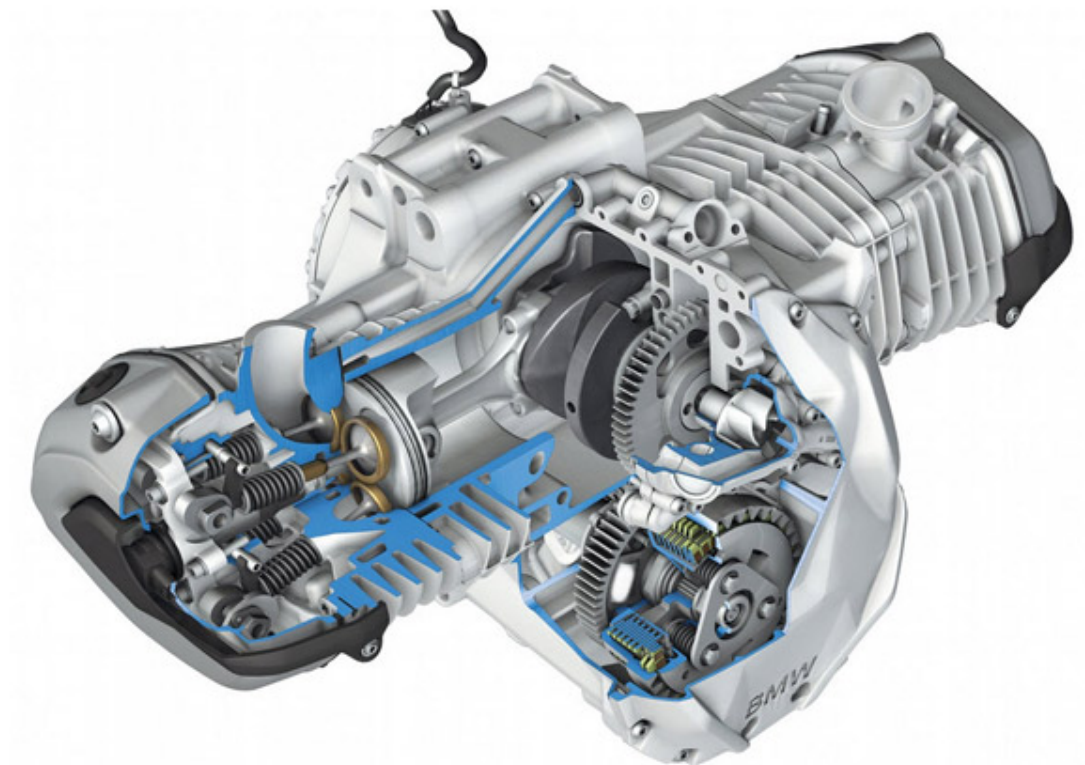


Java versions releases

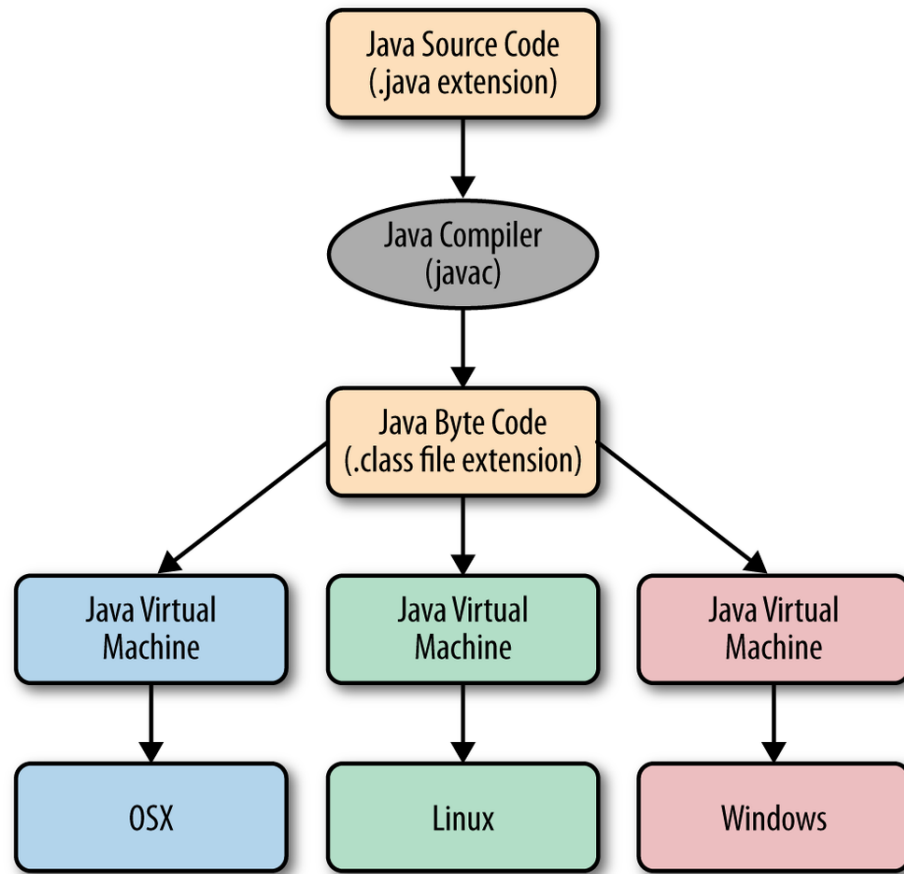


JVM

Java Virtual Machine (JVM) - ключевой компонент платформы Java



JVM



JRE

- **Java Runtime Environment (JRE)** - это исполнительная система Java, необходимая для исполнения Java-программ.
- **JRE** содержит:
 - минимальную реализацию **JVM**
 - Java **libraries**.

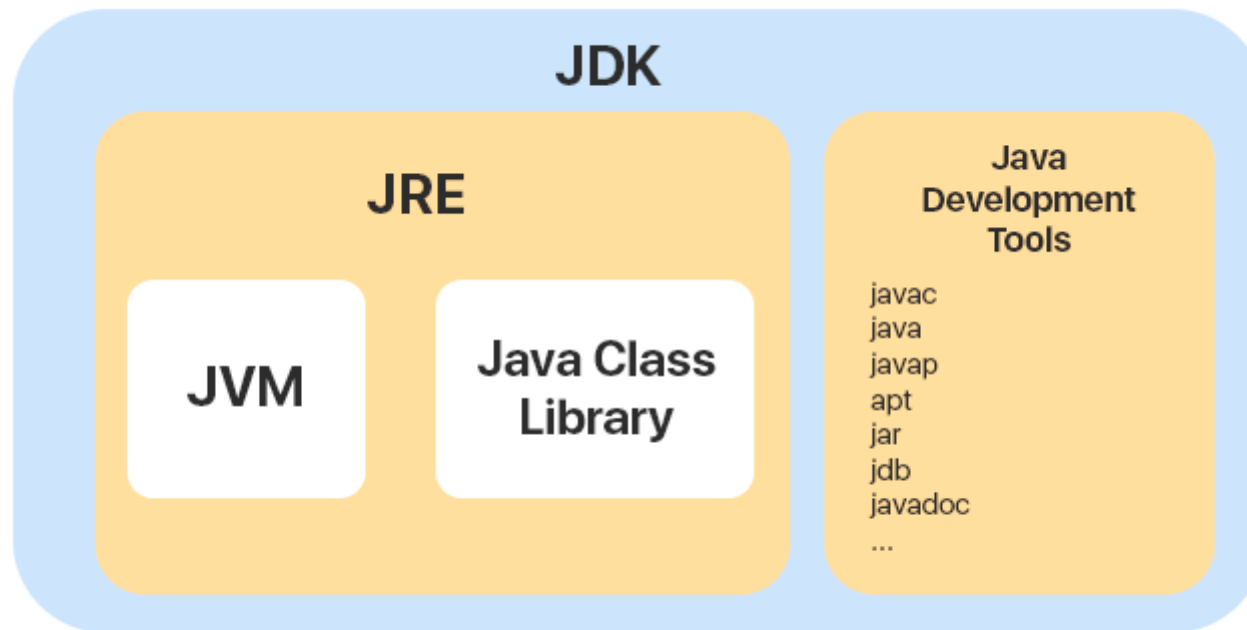
JDK

Java Development Kit (JDK) - бесплатно распространяемый комплект разработчика приложений на языке Java.

JDK включает в себя:

- исполнительную систему Java (**JRE**).
- компилятор Java (**javac**)
- стандартные библиотеки классов Java
- примеры
- документацию
- различные утилиты

Recap



Install JDK (OS Windows)

Проверка наличия JRE и JDK

- Открыть *PowerShell*.
 - Пуск → *PowerShell*.
 - **Shift + Mouse Right Click** in currency directory.
- Для проверки наличия **JDK**, необходимо выполнить команду:

```
javac --version
```

- Проанализировать результат работы команды.
- Для проверки наличия **JRE**, необходимо выполнить команду:

```
java --version
```

- Проанализировать результат работы команды.

Установка OpenJDK

- Скачиваем **OpenJDK 11.0.2** (*build 11.0.2*) с сайта **jdk.java.net** для *OS Windows*
- Распаковываем в **d:\opt\java\openjdk-11.0.2**

Установка системных переменных среды (с помощью UI)

- Зайти в *Этот компьютер* → *Свойства* → *Дополнительные параметры системы* → *Дополнительно* → *Переменные среды*.
- Добавить системную переменную **JAVA_HOME** с значением **d:\opt\java\openjdk-11.0.2**.
- Отредактировать системную переменную **PATH**, создав **%JAVA_HOME%\bin**.
- Проверить установку JDK.

Установка системных переменных среды (с помощью CLI)

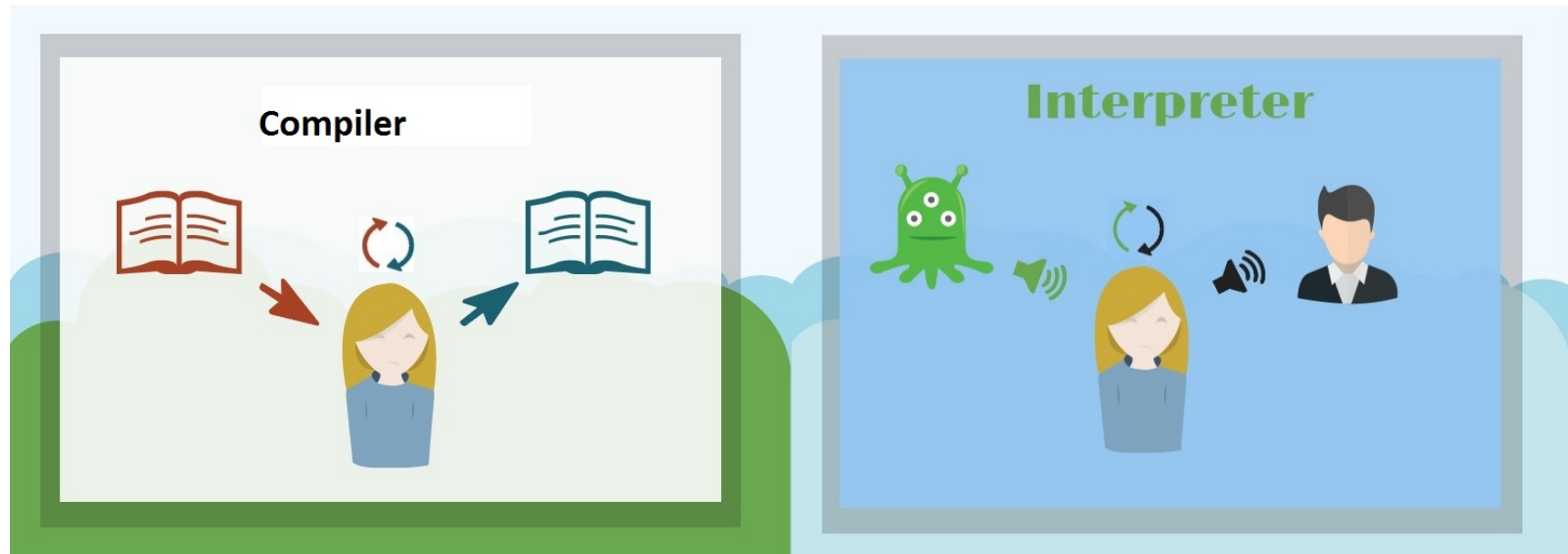
- Выполнить команды:

```
setx JAVA_HOME d:\opt\java\openjdk-11.0.2  
setx PATH "%PATH%;%JAVA_HOME%\bin"
```

- Перезапустить OS.
- Проверить установку JDK.

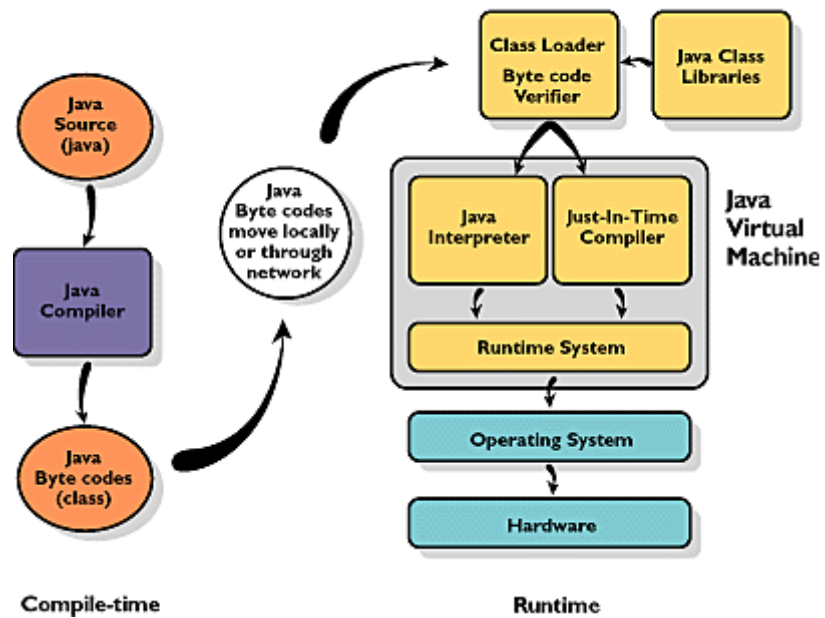
Compile and Run time

How computer understands Java?



Compile and Run time

Java Development and Runtime Environment



Write **source code**

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

Compile with **javac** to **byte code**

```
javac Hello.java
```

Have **byte** code

```
// class version 55.0 (55)
// access flags 0x21
public class Hello {

    // compiled from: Hello.java

    // access flags 0x1
    public <init>()V
        L0
            LINENUMBER 1 L0
            ALOAD 0
            INVOKESPECIAL java/lang/Object.<init> ()V
            RETURN
        L1
            LOCALVARIABLE this LHello; L0 L1 0
            MAXSTACK = 1
            MAXLOCALS = 1

    // access flags 0x9
    public static main([Ljava/lang/String;)V
        L0
            LINENUMBER 3 L0
            GETSTATIC java/lang/System.out : Ljava/io/PrintStream;
            LDC "Hello World!"
            INVOKEVIRTUAL java/io/PrintStream.println (Ljava/lang/String;)V
        L1
            LINENUMBER 4 L1
            RETURN
        L2
            LOCALVARIABLE args [Ljava/lang/String; L0 L2 0
            MAXSTACK = 2
            MAXLOCALS = 1
    }
}
```


Run with interpreter **java** on JVM

```
java Hello
```

```
java -cp . Hello
```

See in output

```
Hello World!
```

Very hard?

Integrated development environment (IDE)



IntelliJ IDEA



Eclipse



NetBeans