

Module 1

- 1.Introduction part, Languages Programing, Java history, JDK, "hello World"
- 2.Project, package, Class, method
- 3.Variables. Keyboard input
- 4.Variables and data types
- 5.Consultation

INTRODUCTION TO **JAVA**

MATERIALS AND RESOURCES

- ▶ Books

- ▶ Head First Java (2nd edition) - Bert Bates, Kathy Sierra
- ▶ Grokking Algorithms - Aditya Y. Bhargava
- ▶ Thinking in Java (4th edition) - Bruce Eckel

- ▶ Documentation & help

- ▶ Oracle's Java Tutorial: <https://docs.oracle.com/javase/tutorial/>
- ▶ Q&A: <https://stackoverflow.com/questions/tagged/java>

GETTING STARTED

- ▶ Toolbox

- ▶ **JDK 19**

- <https://www.oracle.com/java/technologies/javase/jdk19-archive-downloads.html>

- ▶ **IntelliJ IDEA Community Edition:**

- <https://www.jetbrains.com/idea/download/>

- ▶ **Git:** <https://git-scm.com/downloads>

- ▶ Collaboration

- ▶ **GitHub:** <https://github.com/>

- ▶ Communication

- ▶ **Telegram**

GETTING STARTED

Java developer

1. Java от 8 до 17
2. Database
3. Spring as framework
4. Git
5. Unix systems
6. ...

QA engineer

1. Java Basic
2. Database
3. Tools for test
4. Git
5. Unix systems
6. ...

Front-end developer

1. Java Basic
2. Database
3. JavaScript, React or Angular
4. Git
5. Unix systems
6. ...

Классификация

1. (стажеры, Trainee), которые знают основы Java, но пока не умеют применять их на практике.
2. (младший разработчик, Junior). Имеет минимальный опыт работы, его работу проверяют коллеги.
3. (Java Middle). Этот специалист уже имеет опыт работы, он может самостоятельно писать код, который не нуждается в проверке, работает с заданиями средней сложности.
4. (старший разработчик, Senior). Легко справляется с любыми задачами, несет полную ответственность за сроки, написанный код, внесенные изменения.
5. (TeamLead, TeachLead). Специалист высокого уровня, который не только занимается написанием кода, но и раздает ТЗ, контролирует работу младших сотрудников.
6. (Architect). Самый высокооплачиваемый специалист, занимающийся проектированием, который может самостоятельно вытянуть весь проект

ПЛЮСЫ И МИНУСЫ

Плюсы

1. Вакансий Java-программистов много, поэтому найти работу не проблема.
2. Профессию можно освоить примерно за 1 год на курсах.
3. Возможность работать удаленно.
4. Перспективное профессиональное направление.
5. Работа в творческой команде.
6. Стабильная, высокая зарплата.

Минусы

1. Большая ответственность.
2. Без опыта сложно устроиться на работу.
3. Болят пальцы.

Личные качества

1. [Целеустремленность.](#)
2. [Сосредоточенность.](#)
3. [Аккуратность.](#)
4. [Пунктуальность.](#)
5. [Желание самообучаться.](#)

JAVA

OVERVIEW

BRIEF HISTORY

- ▶ **Green Team** (James Gosling, Mike Sheridan, Patrick Naughton)
- ▶ Initiated language project in June 1991
- ▶ Originally designed for **small embedded systems** in electronic appliances
- ▶ Initially called **Greentalk**, shortly after renamed to **Oak**
- ▶ A few years later renamed to **Java**

REASONS BEHIND NAMING

- ▶ Reflection of the essence of the technology
 - ▶ Revolutionary and dynamic
 - ▶ Easy to spell and fun to say
- ▶ Suggested names were Dynamic, Silk, Jolt, DNA
- ▶ Java is an island where first coffee was produced

PRINCIPLES

- ▶ Simple and robust
- ▶ Portable and platform independent
- ▶ High performant, multithreaded and secured
- ▶ Following object oriented language features

WHAT JAVA IS USED FOR

- ▶ Mobile phones
 - ▶ Google's **Android** OS
- ▶ Point of sale systems
 - ▶ **McDonald's** checkout computers
- ▶ Video games
 - ▶ **Minecraft** - one of the most popular games of all time
- ▶ Trading applications
 - ▶ **Murex** - application used for front to back connectivity is utilized by many banks
- ▶ Big data technologies
 - ▶ **Elasticsearch** - distributed search and analytics engine

RELEASE TIME LAPSE

- ▶ JDK 1.1 Major Changes
 - ▶ Extensive retooling for AWT event model
 - ▶ Introduced inner classes
 - ▶ Addition of JDBC and JavaBeans
- ▶ J2SE 1.2 Major Changes
 - ▶ Rebranded as Java 2
- ▶ J2SE 1.3 Major Changes
 - ▶ Bundled with Hotspot JVM
 - ▶ JavaSound, Java Naming and Directory Interface
 - ▶ Java Platform **Debugger** Architecture



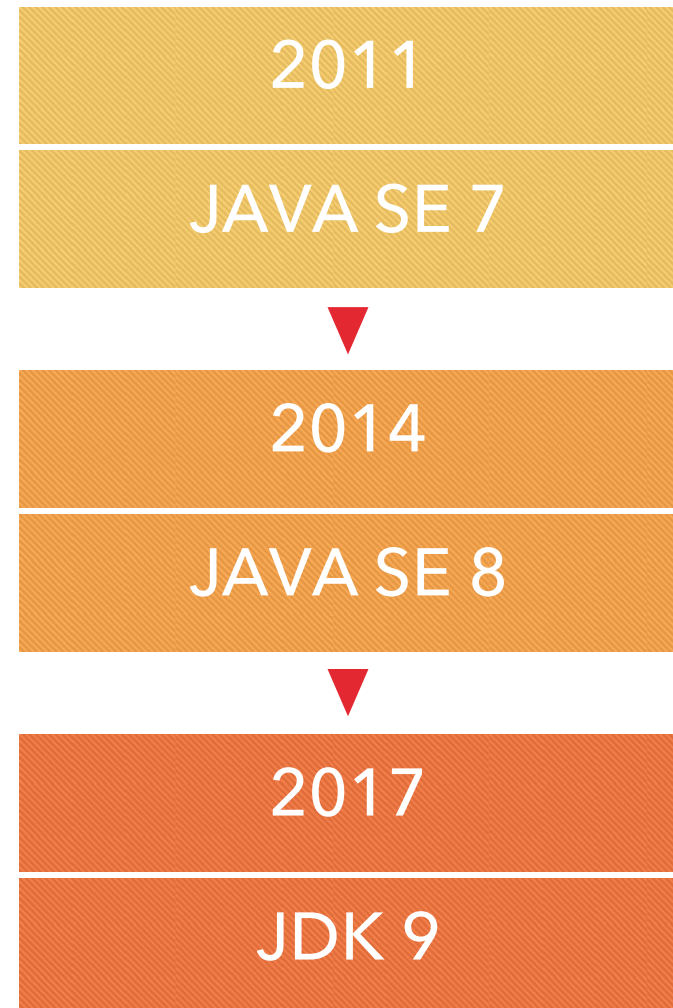
RELEASE TIME LAPSE

- ▶ JDK 1.4 Major Changes
 - ▶ First release developed under **Java Community Process**
 - ▶ Inclusion of **regular expressions**
- ▶ J2SE 5.0 Major Changes
 - ▶ For-each loop
 - ▶ Generics & Auto-boxing
 - ▶ Var-args
- ▶ Java SE 6 Major Changes
 - ▶ **Bundled** with database manager
 - ▶ **Facilitates** use of scripting languages with JVM



RELEASE TIME LAPSE

- ▶ Java SE 7 Major Changes
 - ▶ Strings in **switch**
 - ▶ JVM support for dynamic languages
- ▶ Java SE 8 Major Changes
 - ▶ Lambda expressions & Streams API
 - ▶ Default methods
 - ▶ Reworked dates API
- ▶ JDK 9 Major Changes
 - ▶ Module system
 - ▶ Extended Streams API

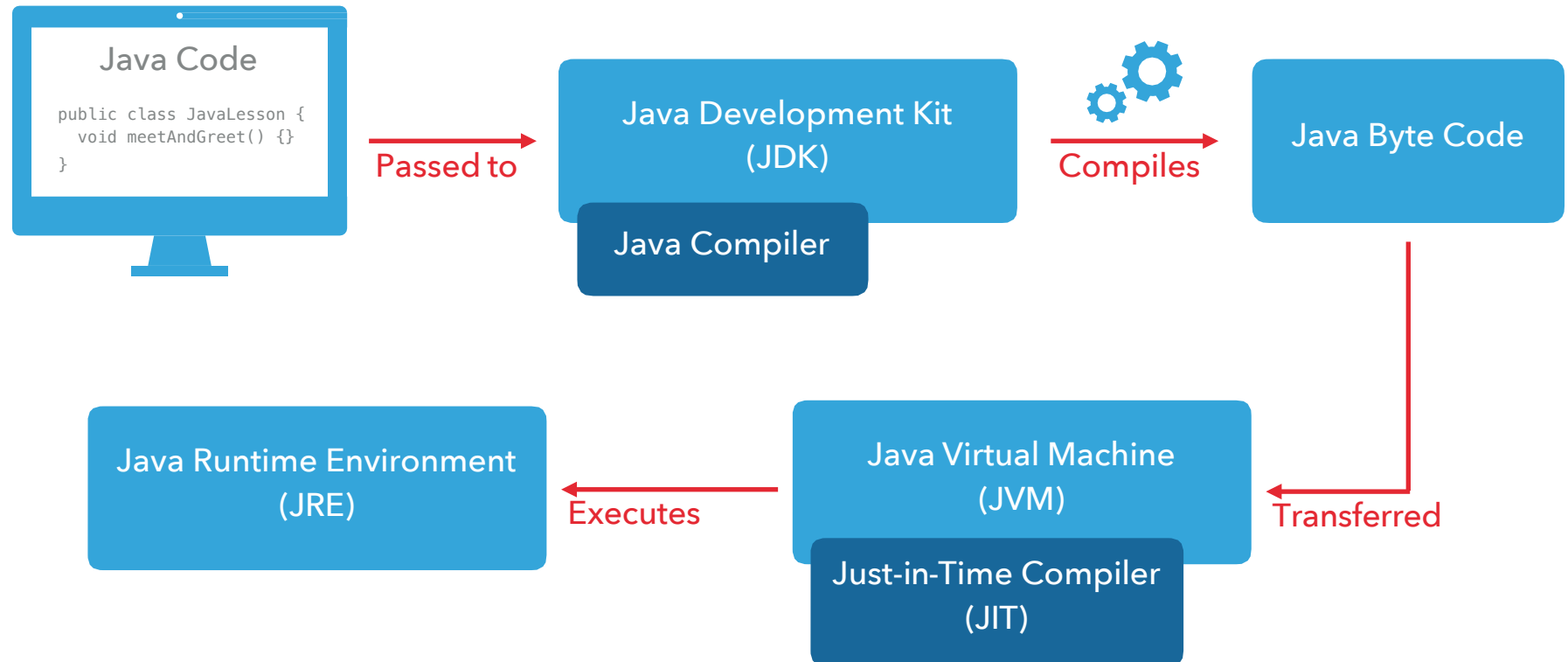


Java and evolution

- James Gosling, Mike Sheridan, Patrick Naughton
- Green Team
- Greentalk .gt
- Oak
- Java
- Sun Microsystem
- JDK 1.0 был выпущен 23 января 1996 года.
- Oracle Corporation

JDK Альфа и Бета (1995)
JDK 1.0 (23 января 1996 г.)
JDK 1.1 (19 февраля 1997 г.)
J2SE 1.2 (8 декабря 1998 г.)
J2SE 1.3 (8 мая 2000 г.)
J2SE 1.4 (6 февраля 2002 г.)
J2SE 5.0 (30 сентября 2004 г.)
Java SE 6 (11 декабря 2006 г.)
Java SE 7 (28 июля 2011 г.)
Java SE 8 (18 марта 2014 г.)
Java SE 9 (21 сентября 2017 г.)
Java SE 10 (20 марта 2018 г.)
Java SE 11 (сентябрь 2018 г.)
Java SE 12 (март 2019 г.)
Java SE 13 (сентябрь 2019 г.)
Java SE 14 (март 2020 г.)
Java SE 15 (сентябрь 2020 г.)
Java SE 16 (март 2021 г.)
Java SE 17 (сентябрь 2021 г.)
Java SE 18
Java SE 19
Java SE 20
Java SE 21

INTERNAL WORKFLOW



JAVA COMPONENTS: JVM & JRE & JDK

- ▶ Java Virtual Machine (**JVM**) is an **abstract computing machine** that **enables** computer to run a Java program
- ▶ Java Runtime Environment (**JRE**) is a software package that **contains** what is required **to run** a Java program. It includes a Java Virtual Machine implementation together with an implementation of the Java Class Library
- ▶ Java Development Kit (**JDK**) is a **superset** of a JRE and **contains tools** for Java programmers, e.g. a java compiler

JAVA COMPONENTS: JVM & JRE & JDK

- ▶ Виртуальная машина Java (JVM) – это абстрактная вычислительная машина, которая позволяет компьютеру запускать программу Java.
- ▶ Java Runtime Environment (JRE) – это программный пакет, содержащий все необходимое для запуска программы Java. Он включает в себя реализацию виртуальной машины Java вместе с реализацией библиотеки классов Java.
- ▶ Комплект для разработки Java (JDK) представляет собой расширенный набор JRE и содержит инструменты для программистов на Java, например. компилятор Java

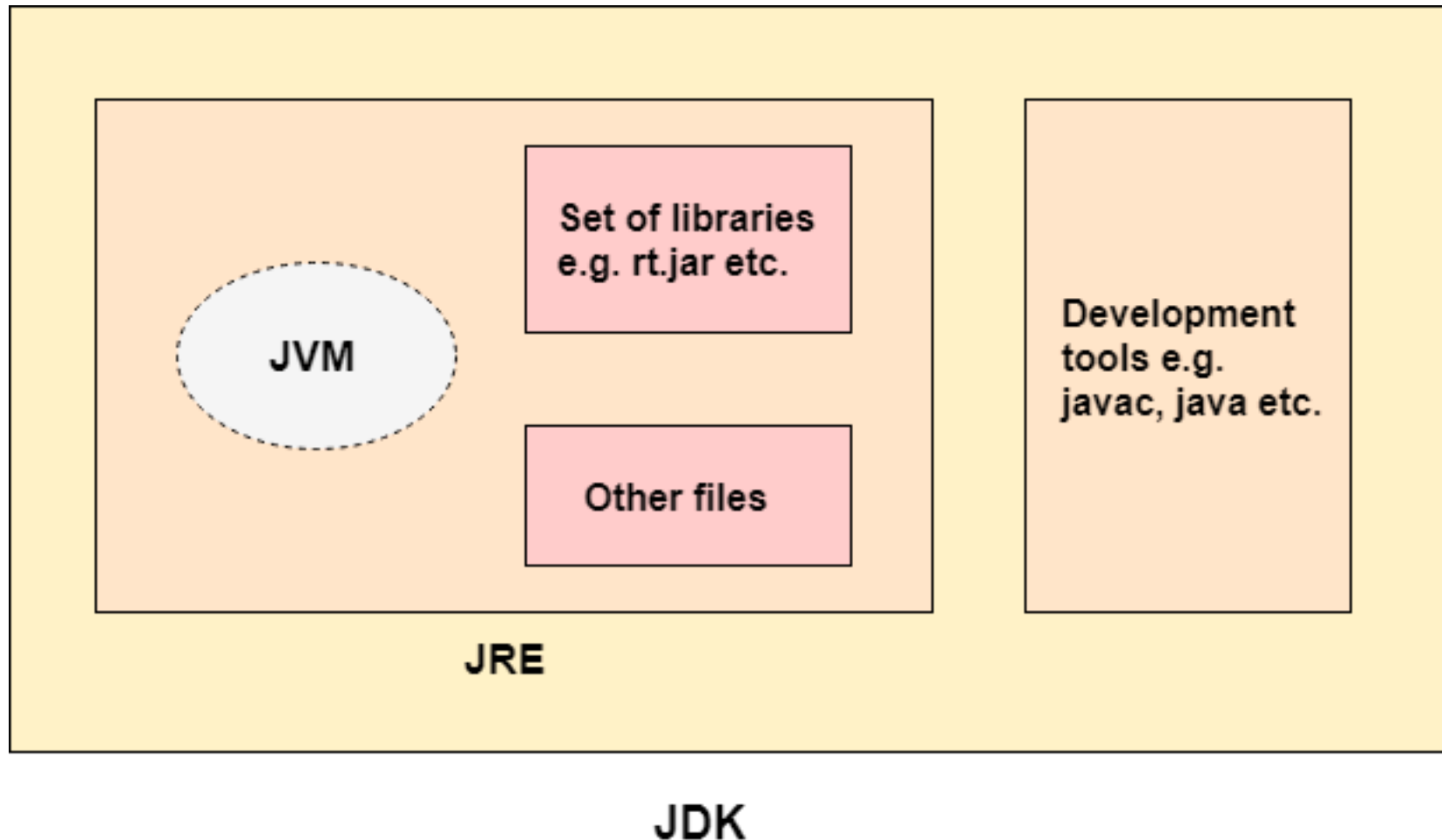
JAVA COMPONENTS: JVM & JRE & JDK

JVM (Java Virtual Machine) - это виртуальная машина Java - основная часть исполняющей системы Java, так называемой Java Runtime Environment (JRE). Виртуальная машина Java исполняет байт-код Java, предварительно созданный из исходного текста Java-программы компилятором Java (javac). JVM обеспечивает платформо-независимый способ выполнения кода. Программисты могут писать код не задумываясь, как и где он будет выполняться.

JRE (Java Runtime Environment) - это минимальная реализация виртуальной машины, необходимая для исполнения Java приложений, без компилятора и других средств разработки. Состоит из виртуальной машины и библиотек Java классов.

JDK (Java Development Kit) - это комплект разработчика приложений на языке Java, включающий в себя компилятор, стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему JRE.

JAVA COMPONENTS: JVM & JRE & JDK



ALGORITHMS

DEFINITION OF AN ALGORITHM

- ▶ An algorithm is a procedure or formula for solving a problem, based on **conducting a sequence** of specified actions
- ▶ In mathematics and computer science, an algorithm usually means a small procedure that solves a **recurrent** problem
- ▶ A computer program can be viewed as an **elaborate** algorithm

ALGORITHM BREAKDOWN: TAXI RIDE

1. Contact with the taxi service
2. Provide pickup & destination addresses
3. Wait for the cab
4. Get in the car
5. Enjoy your ride
6. Pay for the service upon arrival

ALGORITHM BREAKDOWN: PLANT A PLANT

1. Dig a hole
2. Put roots into soil
3. Cover the hole
4. If soil is wet then end
5. Water the plant