



Lesson 1

13.12.2021



2 место



1 место



3 место

1 Изначально язык был создан для программирования бытовых электронных устройств, но очень скоро стало ясно, что его возможности гораздо шире.

2 Создатели Java реализовали принцип WORA – “write once, run anywhere” (“пиши один раз и запускай везде”). Это позволяет запускать приложения везде, где есть среда исполнения JRE.

3 Это один из самых СТАБИЛЬНЫХ языков. Можно спокойно читать код 20-летней давности. Это одна из причин, по которой его любят корпорации 🧑‍💼, особенно, финансового сектора. Многие мировые инвестиционные банки, типа Goldman Sachs, Citigroup, Barclays используют Java. А еще Java используют в Amazon, eBay, Oracle, IBM, Intel.

4 Это один из наиболее ВЫСОКООПЛАЧИВАЕМЫХ языков. Средняя зарплата джуниора в Украине – \$785, миддла – \$2000, синьора – \$3600, может доходить до \$5-6 тыс и выше у архитектора.

5 Это очень ПОПУЛЯРНЫЙ язык. Например, согласно рейтингу TIOBE Java занимает 2-е место среди языков программирования. Этот рейтинг формируется на основе данных о количестве разработчиков, использующих определенный ЯП, курсов по его изучения и других показателей популярности, таких как количество поисковых запросов. А компания ORACLE утверждает, что общее количество всех устройств, работающих с Java, достигло 3-х миллиардов 😲.

6 Java – это язык разработки под Android 📱 (он вытесняется Kotlin, но Kotlin проблематично выучить без знания Java).

7 За 25 лет существования на Java было написано огромное количество кода, который нужно поддерживать, так что ВОСТРЕБОВАННОСТЬ Java разработчиков не вызывает сомнений.

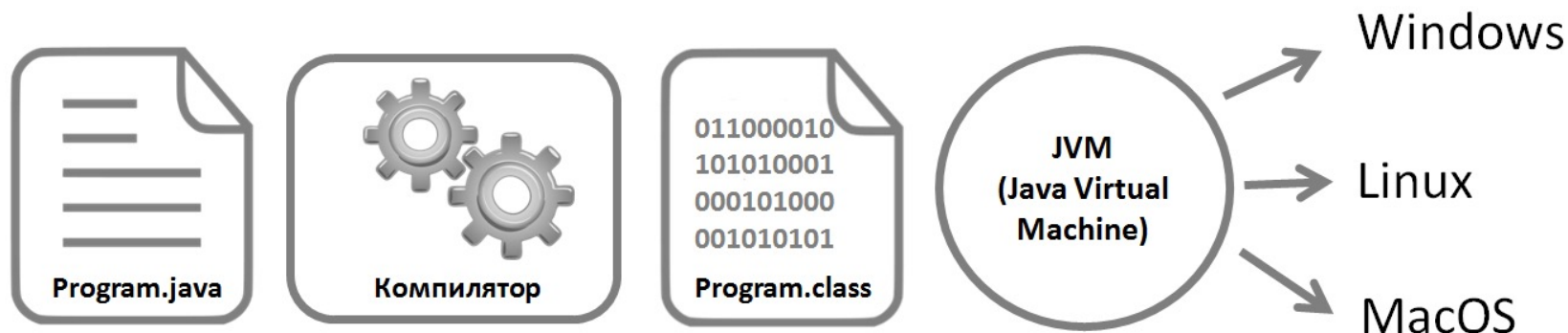


Почему Java

1. "Написано единожды, работает везде"
2. Дружественный синтаксис
3. Объектно-ориентированный язык
4. Управление памятью



Написано единожды, работает везде



Когда Вы будете писать программы на языке Java, они всегда будут сохраняться отдельными файлами. Причём эти файлы всегда будут иметь расширение .java.

Далее, когда Вы запускаете компилятор (javac), Ваш код из человекочитаемого превращается в так называемый байт-код (то есть в виде разных комбинаций 0 и 1) и код становится исключительно машиночитаемым. После этого появится еще один файл, который всегда будет иметь расширение .class.

Затем JVM (Java Virtual Machine) исполняет байт-код.



Дружественный синтаксис

Взяли всё самое лучшее от лучшей марки "велосипеда" - языков программирования С и С++

Выбросили всё, что посчитали лишним и не особо удачным в С и С++

Внесли новшества в новый язык программирования Java

Объектно-ориентированный язык

Всё вокруг нас является
объектом.



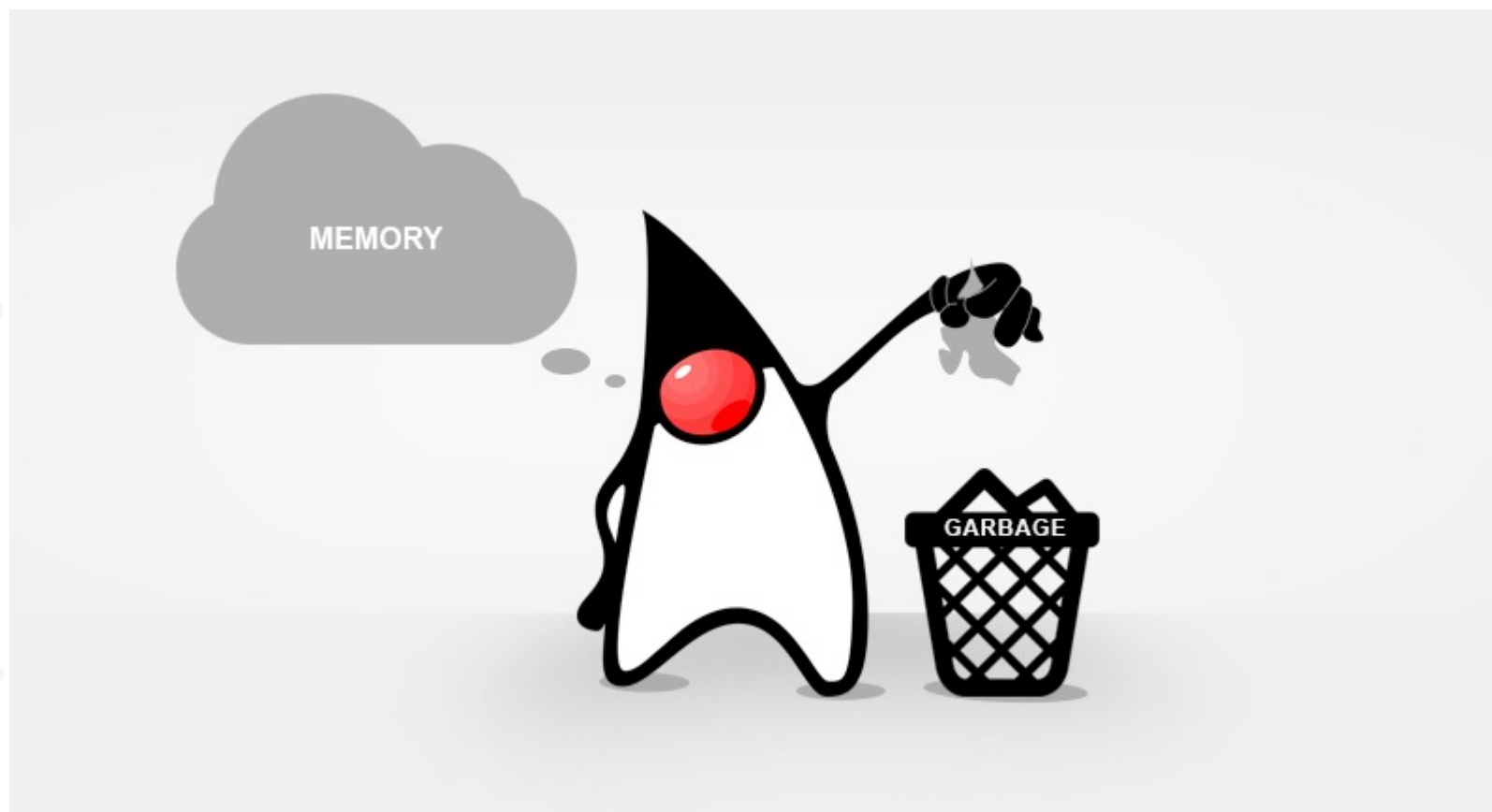
У объекта есть
свойства
(еще называют параметры)



У объекта есть
методы
(методы – это действия,
то есть что может
делать объект)



Управление памятью

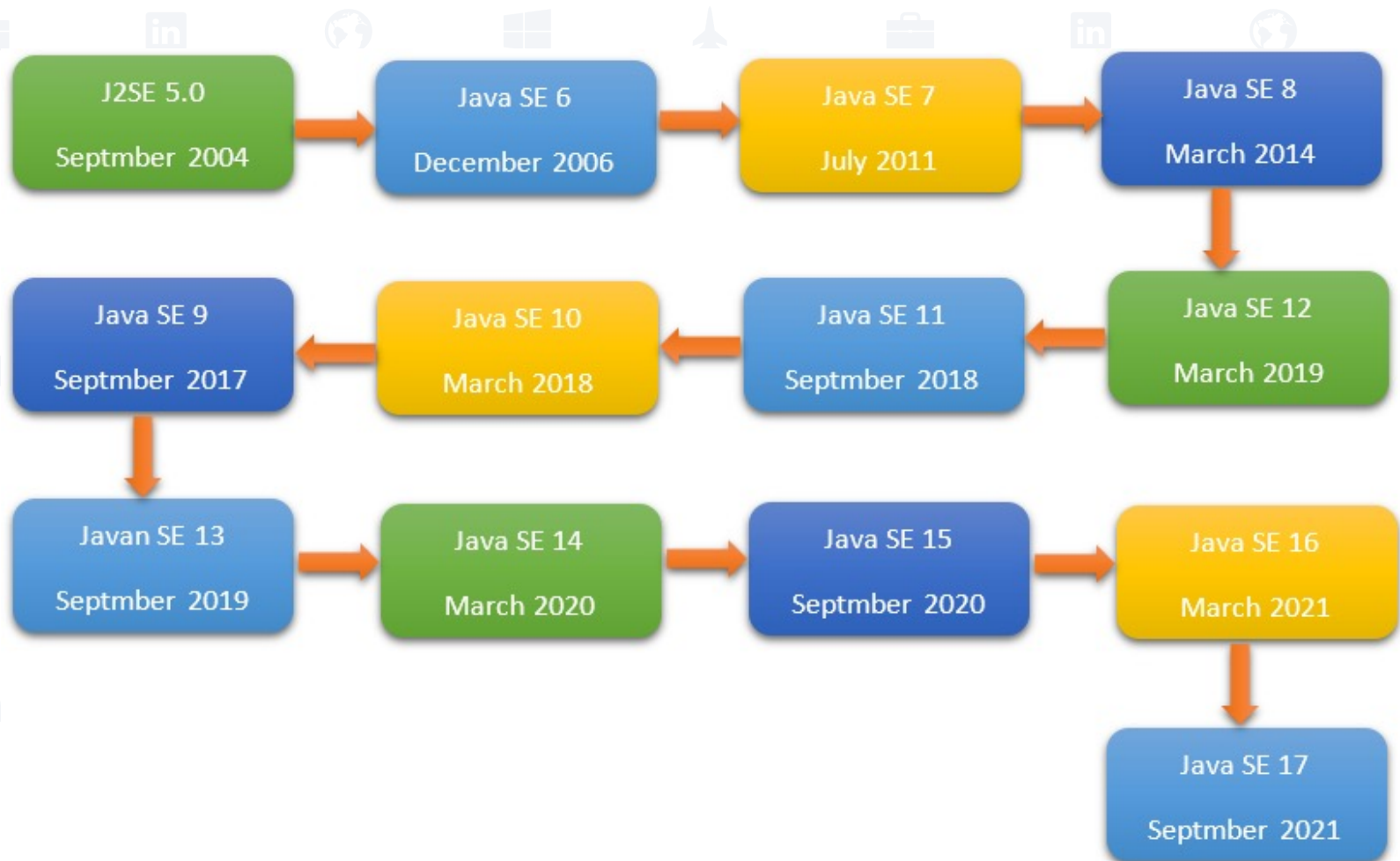


Java история версий

Версия	Дата выпуска	Конец бесплатной поддержки	Конец LTS
JDK Beta	1995	неизвестно	неизвестно
JDK 1.0	январь 1996		
JDK 1.1	февраль 1997		
J2SE 1.2	декабрь 1998		
J2SE 1.3	май 2000		
J2SE 1.4	февраль 2002	октябрь 2008	февраль 2013
J2SE 5.0	сентябрь 2004	ноябрь 2009	апрель 2015
Java SE 6	декабрь 2006	апрель 2013	декабрь 2018
Java SE 7	июль 2011	апрель 2015	июль 2022
Java SE 8 (LTS)	март 2014	<ul style="list-style-type: none"> Январь 2019 (коммерческая лицензия Oracle) Декабрь 2020 (лицензия для персонального использования Oracle) Сентябрь 2023 (для AdoptOpenJDK^[1]) 	март 2025
Java SE 9 ^{[2][3][4]}	сентябрь 2017	Март 2018 (OpenJDK)	—
Java SE 10	март 2018	Сентябрь 2018 (OpenJDK)	—
Java SE 11 (LTS)	сентябрь 2018	Сентябрь 2022 ^[1]	сентябрь 2026
Java SE 12	март 2019	Сентябрь 2019	—
Java SE 13	сентябрь 2019	Март 2020	—
Java SE 14	март 2020	Сентябрь 2020	
Java SE 15	сентябрь 2020	Март 2021	
Java SE 16	март 2021	Сентябрь 2021 (OpenJDK)	
Java SE 17	сентябрь 2021	Сентябрь 2030	

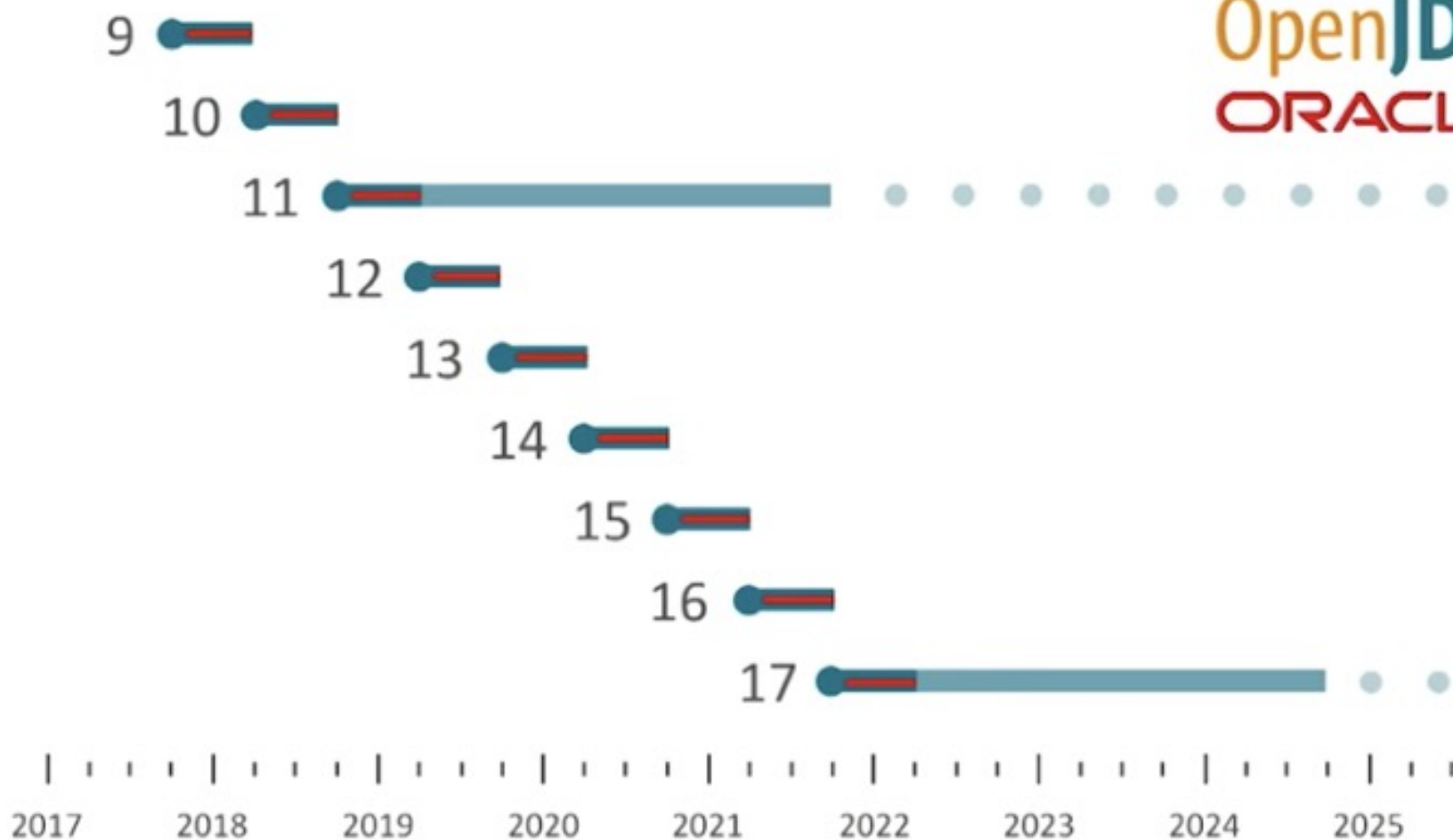


Java история версий



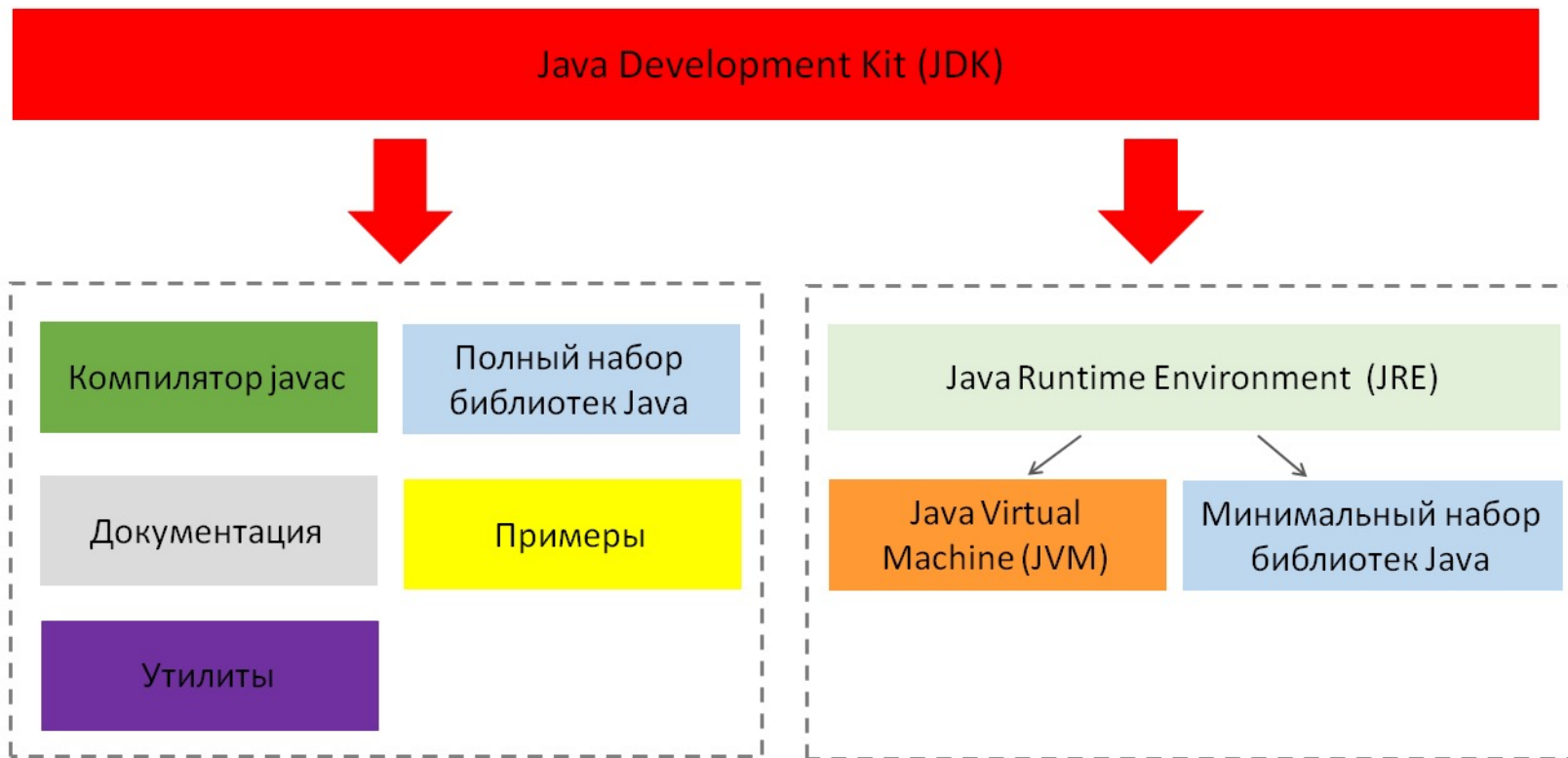


OpenJDK
ORACLE®






JDK vs JRE



*JRE достаточно для выполнения Java программ.
А для разработки программ JRE недостаточно.*



JDK: Software Development Kit: Это набор средств разработки для Java, в JRE, и компиляторы (compilers) и инструменты (как JavaDoc и Java Debugger) чтобы создавать и компилировать программы.

JRE: Java Runtime Environment (Среда выполнения для Java). По существу это виртуальная машина Java (Java Virtual Machine) на котором запущены ваши программы Java. Она так же включает браузер плагина для выполнения Applet.



Установка JDK

Шаг 1: Скачать установочный файл с сайта -

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

Шаг 2: Установить JDK из файла

Шаг 3: Установка переменных среды — JAVA_HOME и PATH



Чтобы настроить программное окружение на Ubuntu, необходимо сделать несколько шагов:

Шаг №1 - Запустить консоль (сочетание клавиш Ctrl + Alt + T), а также убедиться что вы обладаете правами устанавливать программы на компьютер

Шаг №2 - Обновить списки пакетов:

```
sudo apt-get update
```

Шаг №3 - Проверить версию Java, если ранее не устанавливали должно вернуть "The program java can be found in the following packages":

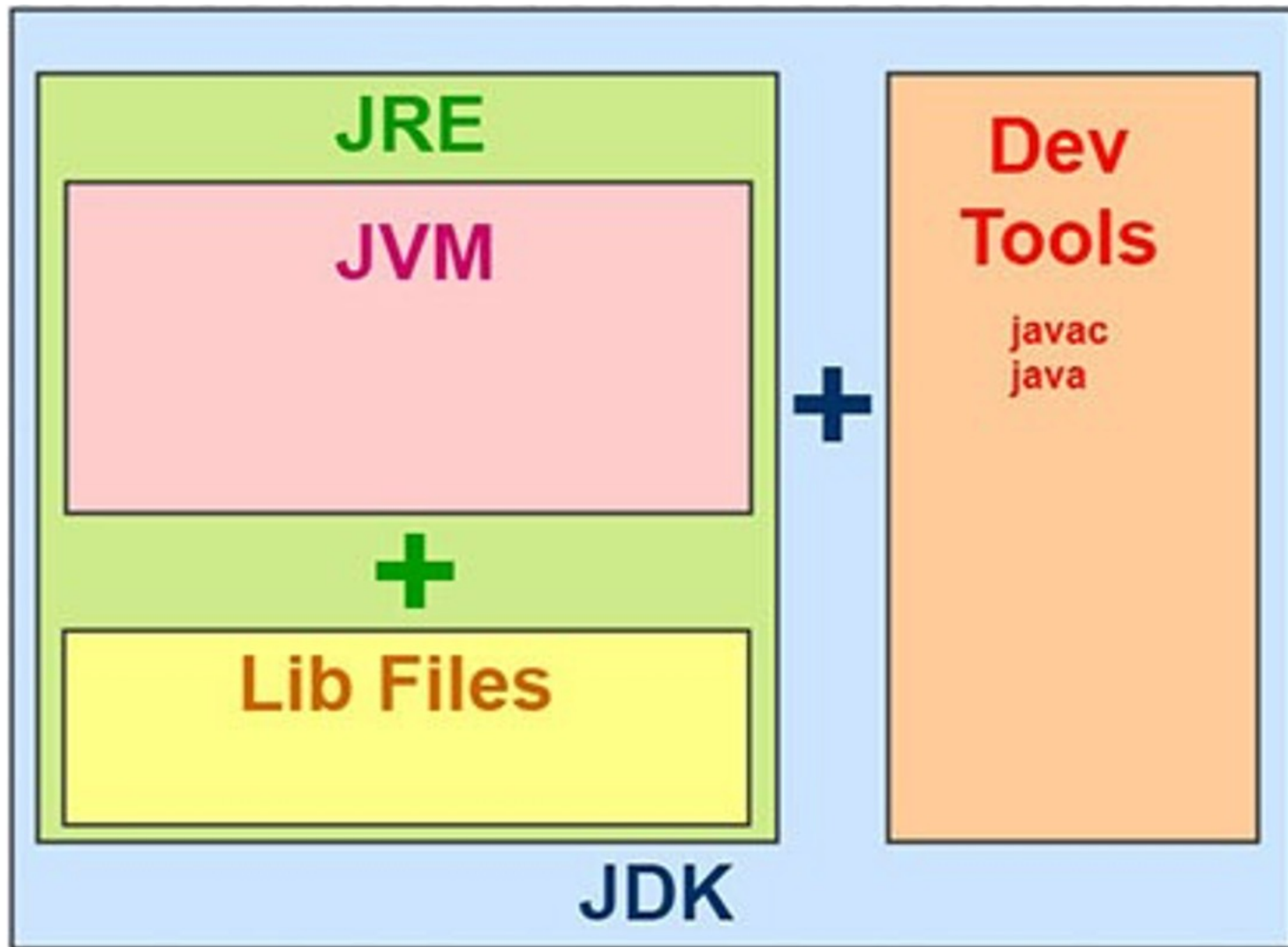
```
java -version
```

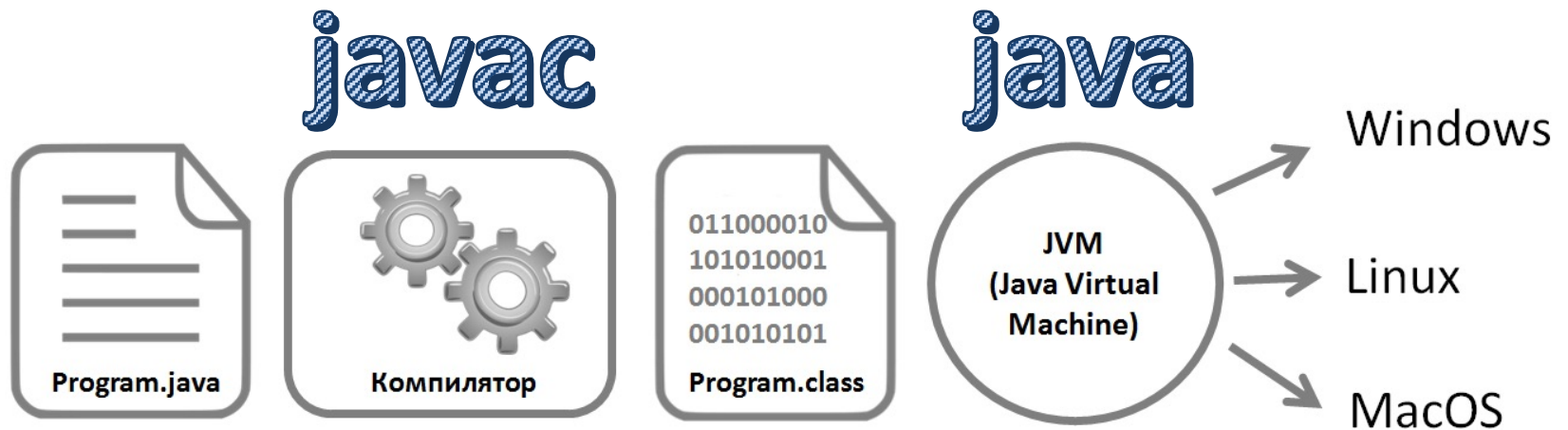
Шаг №4 - Дальше, на сегодняшний день, актуально установить JDK 8:

```
sudo apt-get install oracle-java8-installer
```

Шаг №5 - И снова проверить версию и убедиться что стоит JDK 8:

```
java -version
```





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





Система контроля версий (Version Control System, VCS) — программное обеспечение для облегчения работы с изменяющейся информацией.

VCS позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.



Для чего нужны VCS?

Хранение полной истории изменений

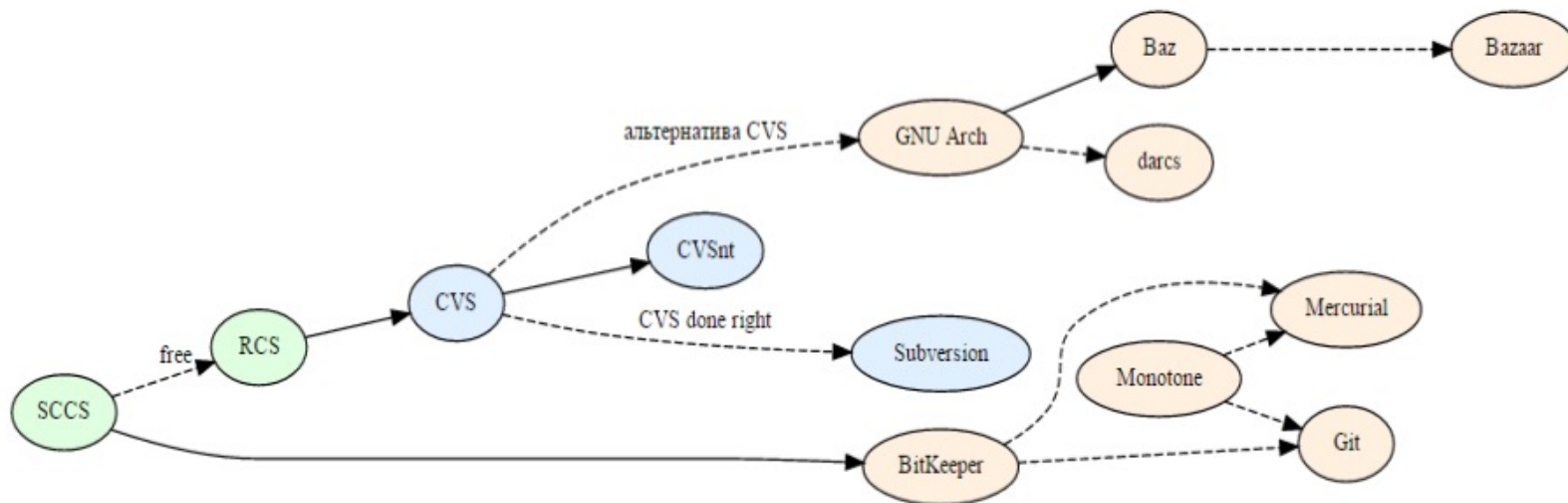
Описание причин всех производимых изменений

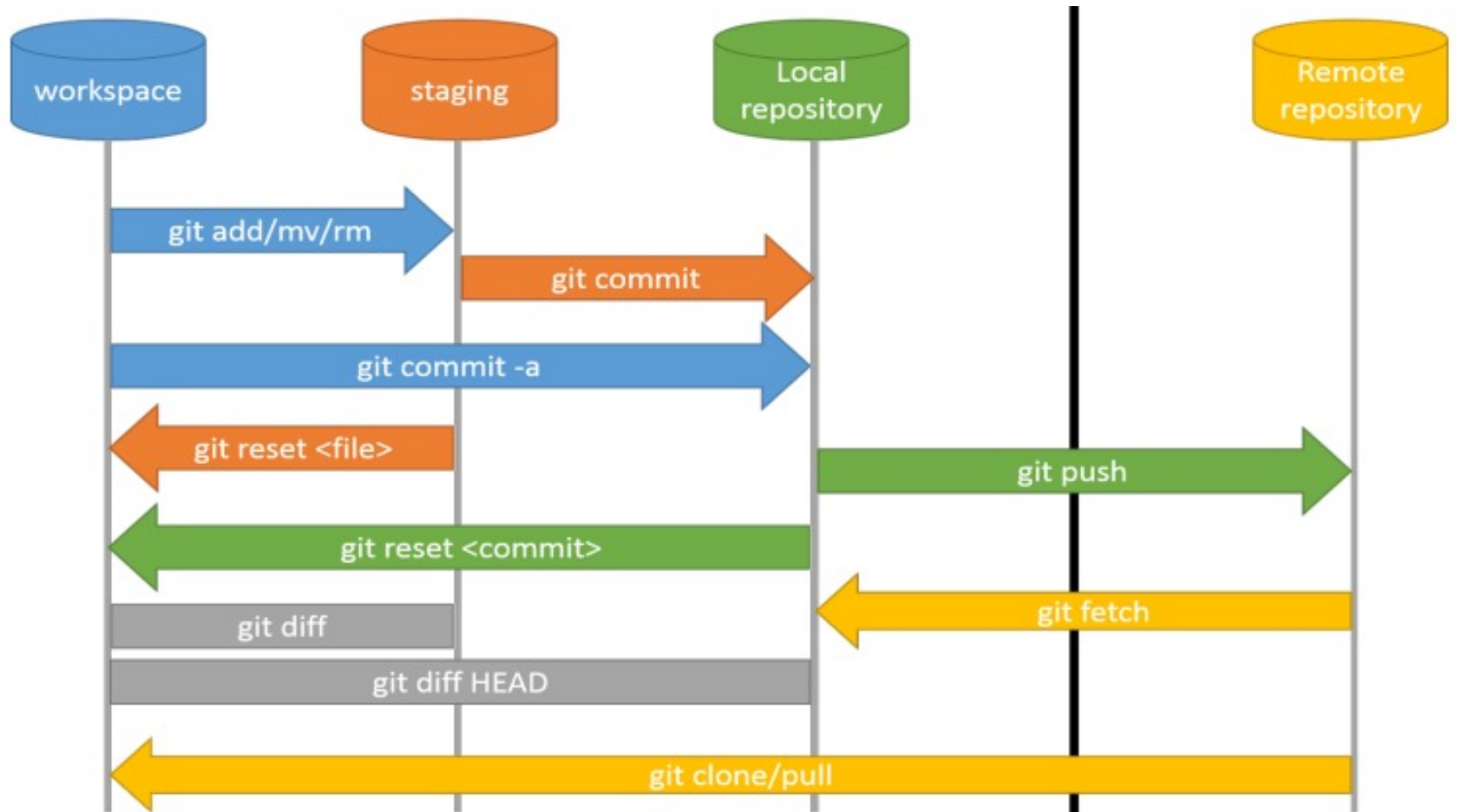
Откат изменений, если что-то пошло не так

Поиск причины и ответственного за появления ошибок в программе

Совместная работа группы над одним проектом

Возможность изменять код, не мешая работе других пользователей







Создание репозитория в существующем каталоге:

\$ git init

Клонирование существующего репозитория:

\$ git clone [url]

Добавление изменённых файлов в индекс:

\$ git add file [, file]

Фиксация изменений:

\$ git commit

Откат к предыдущему состоянию проекта:

\$ git reset – hard HEAD

Слияние веток:

\$ git merge

Обновление удалённого репозитория:

\$ git push

