**Язык Java. Лекция 1. – История и особенности языка**

**1.1 Краткая история создания и основания для разработки**

Язык программирования Java изначально разрабатывался Sun Microsystems по инициативе Джеймса Гослинга и был выпущен в 1995 году как основной компонент платформы Java Sun Microsystems (Java 1.0 [J2SE]).

Последним выпуском Java Standard Edition является Java SE 8. С развитием Java и его широко распространенной популярностью были созданы различные конфигурации для различных типов платформ. Например, J2EE для корпоративных приложений, J2ME для мобильных приложений.

Новые версии J2 были переименованы в Java SE, Java EE и Java ME соответственно.

Джеймс Гослинг начал проект по языку Java в июне 1991 года для использования в одном из своих многочисленных проектов приставок. Язык, который первоначально назывался «Дуб» в честь дуба, стоявшего рядом с офисом Гослинга, также получил название «Зеленый» и впоследствии был переименован в Java из списка случайных слов.

Sun выпустила первую публичную реализацию под названием Java 1.0 в 1995 году. Она обещала **Write Once, Run Anywhere** (**Напиши один раз, Исполняй везде**), предоставляя бесплатное время выполнения на популярных платформах.

13 ноября 2006 года Sun выпустила большую часть Java как бесплатное программное обеспечение с открытым исходным кодом на условиях GNU General Public License (GPL).

8 мая 2007 г. Sun завершила процесс, сделав весь основной код Java свободным и открытым исходным кодом, **за исключением небольшой части кода, на которую Sun не имела авторских прав**.

**Что помогло языку Java «войти в каждый дом»? Кто-нибудь думал, что язык Java, первоначально предназначенный для устройств бытовой электроники, станет использоваться во многих других областях?**

Расширение влияния Java, как ни странно, началось с разработки браузеров. Главный разработчик Java Джеймс Гослинг перешел в Sun в 1984 году из исследовательского отдела IBM. До Java Гослинг разработал не имевший коммерческого успеха оконный интерфейс NeWS и GOSMACS — первую реализацию текстового редактора EMACS на языке С. После этого Гослинг и его коллеги начали изучать перспективы на рынке электроники и бытовой техники.

Группе была поставлена задача создать **распределенную систему**, которую можно было бы в качестве современной программной технологии продавать производителям бытовой электроники.



Рисунок 1.1. Инициаторы первого проекта Java.

В декабре 1990 года компания Sun создала рабочую группу для разработки новой технологии, куда вошел Патрик Нотон. Однако ему показалось, что возможностей C++ и С API не хватит для выполнения этой задачи. Поэтому Патрик Нотон начал проект под названием Stealth Project (**Ловкий, невидимый проект**) чтобы найти решение проблемы. Позже к нему присоединились Джеймс Гослинг и Майк Шеридан, и проект получил название Green Project. Когда стало понятно, что у проекта – большие перспективы, они пригласили еще 10 человек. Компания Sun поддержала начинание, предположив, что оно ознаменует «**новую волну» в ИТ-отрасли**.

**Команда вскоре задумалась о том, чтобы усовершенствовать язык С++.**

Однако он оказался избыточным и слишком ресурсоемким, чтобы на его базе создать язык, удовлетворяющий поставленным требованиям:

1. **Поддержка объектно-ориентированной парадигмы**;

2. **Возможность запуска одной и той же программы на разных операционных системах**;

3. Встроенные механизмы для работы с компьютерных сетей;

4. Обеспечение безопасности удаленного исполнения программ;

5. Использование преимуществ других объектно-ориентированных языков программирования.

**1.2 Подробная хронология создания языка**

Июнь 1991 г. — Гослинг начинает работу над интерпретатором Oak, который через несколько лет (при поисках торговой марки) переименован в Java.

19 августа 1991 г. — Коллектив разработчиков Green демонстрирует идеи базового пользовательского интерфейса и графическую систему сооснователям компании Sun Скотту Макнили и Биллу Джою.

17 октября 1991 г. — Шеридан и Нотон присваивают конструкторской философии своего коллектива девиз «1st Person», который со временем становится названием компании.

17 ноября 1991 г. — Офис проекта Green снова подключается к главной сети компании Sun линией на **56 Кбит/с**.

1 марта 1992 г. — К проекту Green присоединяется Джонатан Пейн, который позднее участвует в написании HotJava.

Лето 1992 г. — Интенсивная деятельность по доработке Oak, Green OS, пользовательского интерфейса, аппаратуры Star7 и соответствующих компонентов.

4 сентября 1992 г. — Завершена разработка устройства Star7; оно продемонстрировано Джою и Макнили.

1 октября 1992 г. — Из компании SunLabs переходит Уэйн Розинг, принимающий на себя руководство коллективом.

1 ноября 1992 г. — Организована корпорация FirstPerson.

15 января 1993 г. — Коллектив переезжает в Пало Альто в здание, где раньше находилась лаборатория Western Research Lab компании DEC и была основана исходная группа Hamilton Group (она же OSF).

15 марта 1993 г. — После ознакомления с результатами испытаний кабельного интерактивного телевидения, проведенных компанией Time Warner, корпорация FirstPerson сосредотачивается на этой тематике.

Апрель 1993 г. — **Выпуск первого графического браузера для Internet — Mosaic 1.0**, разработанного в центре NCSA.

14 июня 1993 г. — Компания Time Warner продолжает проводить свои испытания интерактивного кабельного ТВ с компанией SGI, несмотря на признанное превосходство технологии компании Sun и уверения, что Sun выиграла эту сделку.

Лето 1993 г. — Нотон пролетает 300 тысяч миль, продавая Oak всем, занимающимся **бытовой электроникой и интерактивным телевидением**; тем временем **темп, с которой люди получают доступ к Internet, головокружительно нарастает**.

Август 1993 г. — Через несколько месяцев многообещающих переговоров с компанией 3DO относительно разработки ОС для приставок, президент 3DO Трип Хокинс предлагает купить технологию. Макнили отказывается, и сделка срывается.

Сентябрь 1993 г. — К коллективу присоединяется Артур Ван Хофф, поначалу — чтобы создать **среду разработки приложений, предназначенных для интерактивного телевидения**, а потом разрабатывающий, главным образом, сам язык.

7 декабря 1993 г. — Экспертиза операций на высоком уровне в FirstPerson обнаруживает, что эта группа не имеет реальных партнеров или маркетинговой стратегии и неясно представляет себе дату выпуска.

8 февраля 1994 г. — Отменено публичное заявление компании FirstPerson о выпуске, которое должно было состояться на конференции Technology, Entertainment and Design (TED).

17 февраля 1994 г. — Исполнительным лицам компании Sun для разносторонней экспертизы представлен альтернативный **бизнес-план** корпорации FirstPerson по разработке **мультимедийной платформы для CD-ROM и онлайновой работы**.

25 апреля 1994 г. — Создана компания Sun Interactive; в нее переходит половина сотрудников FirstPerson.

Июнь 1994 г. — Начат проект Liveoak, нацеленный Биллом Джоем на использование Oak в **крупном проекте небольшой операционной системы**.

Июль 1994 г. — Нотон ограничивает область применения проекта Liveoak, просто переориентировав Oak на Internet.

16 сентября 1994 г. — Пейн и Нотон начинают писать Web Runner – **браузер-аналог Mosaic, позднее переименованный в HotJava**.

29 сентября 1994 г. — Прототип HotJava впервые продемонстрирован исполнительным лицам компании Sun.

11 октября 1994 г. — Нотон уходит в компанию Starwave.

Осень 1994 г. — Ван Хофф реализует компилятор Java на языке Java. Ранее Гослинг реализовывал его на языке С.

23 мая 1995 г. — Компания Sun официально представляет Java и HotJava на выставке Sun World '95.

23 мая 1995 г. – Netscape объявляет о намерении использовать Java при разработке браузера Netscape.

21 сентября, 1995 г. – В Нью-Йорке проходит конференция по Java-разработке

25 Сентября, 1995 г. – Sun объявляет о расширенном сотрудничестве с Toshiba.

26 Сентября, 1995 г. – Sun анонсирует пакет инструментов для разработки с использованием Java-технологий

30 Октября, 1995 г. – На Internet World Conference в Бостоне компании Lotus Development Corp., Intuit Inc., Borland International Inc., Macromedia Inc. и Spyglass Inc. объявляют о намерении лицензировать Java.

4 декабря, 1995 г. – Sun, Netscape и Silicon Graphics создают альянс для разработки инструментария для интернета.

4 декабря, 1995 г. – Borland, Mitsubishi Electronics, Sybase и Symatec объявляют о планах лицензировать Java.

6 декабря, 1995 г. – IBM и Adobe объявляют о планах лицензировать Java.

7 декабря, 1995 г. – Microsoft объявляет о планах лицензировать Java.

23 января, 1996 г. – релиз JDK 1.0

В 1997 году большинство версий Java оставались бесплатными, за исключением **Java Enterprise System**. В этом же году вышла существенно обновленная версия – Java 2, а также выделились отдельные платформы: J2SE, J2EE, J2ME.

[**https://vc.ru/story/20976-sun-story**](https://vc.ru/story/20976-sun-story) **- 28 лет Sun Microsystems: как создавалась, развивалась и была продана компания, разработавшая Java**

<http://progopedia.ru/language/java/#:~:text=Java%20—%20объектно-ориентированный%20язык%20программирования%2C,приложений%20и%20серверного%20программного%20обеспечения>

<https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Oak_%28programming_language%29> - Oak (язык программирования)

<https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history> - история версий Java вплоть до 2023 года

[Язык Java](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_(programming_language)) претерпел несколько изменений со времен [JDK](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_Development_Kit)1.0, а также многочисленные добавления [классов](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Class_(computer_science)) и пакетов в стандартную [библиотеку](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Library_(computer_science)). Начиная с версии J2SE 1.4, эволюция языка Java регулируется [Процессом сообщества Java](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_Community_Process) (JCP), который использует *запросы спецификации Java* (JSR) для предложения и уточнения дополнений и изменений к [платформе Java](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_platform). *Язык определяется спецификацией языка Java* (JLS); изменения в JLS управляются с помощью [JSR 901](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/www.jcp.org/en/jsr/detail?id=901).

В сентябре 2017 года Марк Рейнхольд, главный архитектор платформы Java, предложил изменить график выпуска на "один выпуск функций каждые шесть месяцев", а не на текущий двухлетний график.[[1]](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-6monthsReinhold-1)[[2]](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-6monthsServerSide-2). Это предложение вступило в силу для всех последующих версий и по-прежнему является текущим графиком выпуска.

Помимо языковых изменений, за прошедшие годы в [библиотеку классов Java](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_Class_Library) были внесены и другие изменения, которая выросла с нескольких сотен классов в JDK 1.0 до более чем трех тысяч в J2SE 5. Были введены целые новые [API](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/API), такие как [Swing](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Swing_(Java)) и [Java2D](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java2D), и многие из оригинальных классов и методов JDK 1.0 [устарели](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Deprecation). Некоторые программы позволяют конвертировать Java-программы из одной версии [платформы Java](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_(software_platform)) в более старую (например, Java 5.0, перенесенная в 1.4) ([см. Инструменты резервного копирования Java](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_backporting_tools)).

Что касается дорожной карты поддержки Oracle Java SE,[[3]](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-auto9-3) версия 19 является последней, а версии 17, 11 и 8 являются поддерживаемыми в настоящее время версиями [долгосрочной поддержки](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Long-term_support) (LTS), в которых клиенты Oracle будут получать поддержку Oracle Premier.

Java 8 LTS последнее [бесплатное общедоступное обновление программного](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Free_software) обеспечения для коммерческого использования было выпущено Oracle в марте 2022 года, в то время как Oracle продолжает выпускать бесплатные общедоступные обновления Java 8 для разработки[[3]](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-auto9-3) и личного использования на неопределенный срок.[[4]](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-Java8-4)

Java 7 больше не поддерживается публично. Для Java 11 долгосрочная поддержка не будет предоставляться Oracle для широкой публики; вместо этого ожидается, что более широкое сообщество [OpenJDK](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/OpenJDK), такое как [Eclipse Adoptium](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Adoptium) или [другие](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/OpenJDK#OpenJDK_builds), выполнят эту работу.[[5]](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-OracleJava11LTS-5)

Java 17 последняя (3-я) LTS была выпущена 14 сентября 2021 года[[6].](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-6)

Общая доступность Java 19 началась 20 сентября 2022 года[[7].](https://translated.turbopages.org/proxy_u/en-ru.ru.f7db7a95-63c15691-6a2dbf96-74722d776562/https/en.wikipedia.org/wiki/Java_version_history#cite_note-7)

В 1997 году большинство версий Java оставались бесплатными, за исключением Java Enterprise System. В этом же году вышла существенно обновленная версия – Java 2, а также выделились **отдельные платформы**: **J2SE, J2EE, J2ME**.

**1.3 Назначение и использование отдельных платформ Java**

Каждая платформа включает в себя:

• язык программирования;

• набор служебных программ (JVM, JRE, JDK и т.п.);

• набор библиотек.

**Java 2 Platform, Standard Edition (J2SE)** предназначается для использования **на рабочих станциях и персональных компьютерах. Является основой всех других Java-технологий.**

**Java Virtual Machine (JVM)** представляет собой программу, имитирующую работу некоторой гипотетической Java-машины.

**Java Runtime Environment (JRE)** представляет собой минимальный набор средств, необходимых для запуска Java-программ. Основу JRE составляют виртуальная машина Java и набор стандартных библиотек.

**Java Development Kit (JDK)** представляет собой набор средств, используемых в процессе разработки Java-программ: компилятор, отладчик, средства для автоматической генерации документации, комплект примеров и т.п. JDK работает с уже готовыми текстами программ и не содержит какого-либо редактора.

**Java Development Kit (JDK) представляет собой набор средств, используемых в процессе разработки Java-программ: компилятор, отладчик, средства для автоматической генерации документации, комплект примеров и т.п. JDK работает с уже готовыми текстами программ и не содержит какого-либо редактора.**

Интегрированные среды разработки для компиляции программ используют внешний JDK.

**JavaFX работает на основе Java.** На платформе JavaFX разработчики могут создавать и развертывать полнофункциональные интернет-приложения (RIA), одинаково стабильно функционирующие на различных платформах. JavaFX расширяет возможности Java, позволяя разработчикам использовать любые библиотеки Java в приложениях JavaFX. Разработчики могут расширить свои возможности в Java и воспользоваться технологией презентаций, обеспечиваемой JavaFX для создания увлекательных визуальных образов.

JavaFX — это по сути инструментарий GUI для Java.

Все наверняка слышали о **Swing**. Это библиотека для создания пользовательских, графических интерфейсов. В связи с тем, что наш горячо любимый Oracle еще не полностью отказался от Swing, он не считается устаревшим, и приложения на нем по-прежнему работают. **Однако он больше не модернизируется Swing, и ребята из Oracle дали нам понять, что за JavaFX будущее. Да и по сути, JavaFX использует компоненты Swing как поставщика услуг**.

**1.4 Сферы применения языка**

В настоящее время Java-технологии находят широкое применение в различных сферах. **Это, в первую очередь, вызвано особенностями создания и запуска Java-приложений**.

**А. Разработка приложений (application)**

Язык Java используется для разработки обычных прикладных программ, которые запускаются пользователем и имеют доступ ко всем ресурсам компьютера наравне с любыми другими программами.

Для запуска Java-приложений в операционной системе должна быть установлена среда исполнения Java (JRE).

**B. Разработка мидлетов (midlet)**

Язык Java используется для разработки специализированных программ, предназначенных для использования на мобильных устройствах.

Структура и возможности мидлетов обусловлены техническими особенностями мобильных устройств.

Для запуска мидлетов мобильное устройство должно поддерживать соответствующую возможность.

Приложение для профиля MIDP платформы Java ME. Примером мидлетов являются игры для мобильных телефонов. Приложения MIDP названы «мидлетами» по аналогии с терминами «апплет» и «сервлет».

Основной файл дистрибутива мидлета имеет расширение jar.

**C. Разработка апплетов (applet)**

Язык Java используется для разработки специализированных программ, обычно небольшого размера, запускаемых браузером внутри web-документа для вывода динамического содержимого и/или интерактивного взаимодействия с пользователем. Для запуска апплетов браузер должен содержать соответствующий модуль (plug-in).

**Java-апплет** —[прикладная программа](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), чаще всего написанная на [языке программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [Java](https://ru.wikipedia.org/wiki/Java) в форме [байт-кода](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4). Java-апплеты выполняются в веб-браузере с использованием [виртуальной Java машины](https://ru.wikipedia.org/wiki/Java_Virtual_Machine) (JVM), или в [Sun](https://ru.wikipedia.org/wiki/Sun_Microsystems)'s AppletViewer, автономном средстве для испытания апплетов. Java-апплеты были внедрены в первой версии языка Java в [1995 году](https://ru.wikipedia.org/wiki/1995_%D0%B3%D0%BE%D0%B4).

Java-апплеты обычно пишутся на языке программирования Java, но могут быть написаны и на других языках, которые компилируются в байт‐код Java, таких, как [Jython](https://ru.wikipedia.org/wiki/Jython).

Поддержка апплетов исключена из Java, начиная с версии 11.

Апплеты используются для предоставления интерактивных возможностей веб-приложений, которые не могут быть предоставлены [HTML](https://ru.wikipedia.org/wiki/HTML). Так как [байт-код](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4) Java платформо-независим, то Java-апплеты могут выполняться с помощью плагинов браузерами многих платформ, включая [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), [UNIX](https://ru.wikipedia.org/wiki/UNIX), [Apple Mac OS](https://ru.wikipedia.org/wiki/Apple_Mac_OS) и [GNU](https://ru.wikipedia.org/wiki/GNU)/[Linux](https://ru.wikipedia.org/wiki/Linux).

Такие программы с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), как applet2app[[1]](https://ru.wikipedia.org/wiki/Java-%D0%B0%D0%BF%D0%BF%D0%BB%D0%B5%D1%82#cite_note-1), могут быть использованы для преобразования апплета в самостоятельные программы на Java или исполняемые файлы Linux и Windows.

**D. Разработка JSP-страниц (Java Server Pages)**

**JSP-страницы представляют собой HTML-документы со вставками на языке Java**, которые используются для анализа пользовательских запросов и динамического формирования web-документов на основе результатов обработки этих запросов. Для обработки JSP-страниц Web-сервер должен содержать соответствующий модуль.

**E. Разработка сервлетов (servlet**)

На языке Java разрабатываются специализированные программы-сервлеты, функционирующие в рамках Web-сервера и имеющие доступ к его ресурсам: файлам, базам данных и т.п. Для запуска сервлетов используется **Web-сервер со специальным модулем** (контейнером сервлетов) или специальный сервер приложений.

Сервлет – это **Java-программа, которая в качестве своего суперкласса использует класс HttpServlet**. Сервлет используется для того, чтобы расширить возможности существующего сервера, в частности веб-сервера.

Технология сервлетов является фундаментом разработки web-приложений с помощью языка программирования Java.

**Сервлет** является интерфейсом [Java](https://ru.wikipedia.org/wiki/Java), реализация которого расширяет функциональные возможности [сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)). Сервлет взаимодействует с клиентами посредством принципа запрос-ответ.

Хотя сервлеты могут обслуживать любые запросы, они обычно используются для расширения [веб-серверов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). Для таких приложений технология Java Servlet определяет HTTP-специфичные сервлет классы.

Пакеты javax.servlet и javax.servlet.http обеспечивают интерфейсы и классы для создания сервлетов.

**1.4 Виртуальная Java машина (JVM)**

<https://ru.hexlet.io/blog/posts/chto-takoe-jvm-znakomstvo-s-virtualnoy-mashinoy-java>

<https://skillbox.ru/media/code/kak-ustroena-java-virtual-machine-i-ee-ekosistema/> !!!!

<https://docs.oracle.com/javase/specs/jls/se17/html/index.html> спецификация Java!!!!

Видео о структуре проекта

О JavaFX - [JavaFX — функции, преимущества, примеры кода (axiomjdk.ru)](https://axiomjdk.ru/announcements/2022/08/18/gid-po-javafx-razrabatyvaem-gui-na-java/?ysclid=lcyy0jhxp8534029395)

[Введение в Java FX (javarush.com)](https://javarush.com/groups/posts/2560-vvedenie-v-java-fx)

**1.5 Характеристика языка**

Выясним, как классифицируется язык программирования по следующим категориям:

* **по уровню**: это язык программирования высокого уровня. То есть, его представления данных близки к представлениям человека. (переменные, числа, символы), а не компьютера (адреса, регистры, машинные слова);
* инструкции (операторы) определяют законченное действие с точки зрения человека (вычислить выражение и сохранить полученное значение в переменной), а не в смысле единичной операции процессора (скопировать значение регистра в область памяти с указанным адресом);
* **по способу реализации**: компилируемый в интерпретируемый байт-код, то есть:
  + исходная программа компилируется в универсальный, не зависящий от архитектуры и операционной системы, код (байт-код);
  + код, который получился в результате компиляции исполняется не операционной системой, а интерпретатором байт-кода;
* **по парадигме - строго объектно-ориентированный:**
  + этот язык приспособлен для создания новых композитных типов данных – классов;
  + классы не являются программами;
  + все подпрограммы оформляются в виде методов – членов класса;
  + программирование на Java имеет **два направления**: программирование как создание новых типов и программирование как разработка исполняемых программ;
  + при создании программ можно (и нужно) использовать наследование, инкапсуляцию и полиморфизм.
  + исполняемая программа сводится:
    - к созданию экземпляров подходящих классов и
    - вызовов процедур в виде обращения к методам экземпляров или самих классов.
  + нет никаких исполняемых программных единиц, кроме классов;
* **по характеру – императивный**, то есть:
  + программа и подпрограммы-методы в Java являются последовательностью указаний, что надо сделать (компьютеру);
  + и **в чём-то процедурный**, так как вызов метода экземпляра класса представляет собой обращение к процедуре, а исполняемая программа считается главной (main) процедурой.

Руководство по Java !!!!! - [Руководство по Java (library-it.com)](https://library-it.com/wp-content/uploads/2020/12/rukovodstvo-po-java-evgeniy-popov.pdf)

О классах JDK - <https://translated.turbopages.org/proxy_u/en-ru.ru.4daf5c10-63c59350-027dddc9-74722d776562/https/docs.oracle.com/en/java/javase/16/docs/api/allclasses-index.html>



Рисунок 1.2. Важные коллекции классов JDK.

[Презентация на тему: Приведение типов данных в Java (studfile.net)](https://studfile.net/preview/5848111/page:3/#26)

Контрольные вопросы

1.Понятие Java. Понятие и назначение JRE и JDK. Версии JDK.

2.Cтруктура Java - программы, примеры. Компиляция и запуск программы из командной строки. Понятие и назначение JVM.

3.Простые типы данных Java: понятие, примеры. Преобразование простых типов данных в Java: правила и примеры.

4.Ссылочные типы данных Java: понятие, примеры. Классы Date и GregorianCalendar: назначение и примеры использования.

5.Консольный ввод-вывод данных в Java: характеристика и примеры.

<https://topjava.ru/blog/what-is-the-jdk> Что такое JDK? !!!!!

<https://topjava.ru/blog> смотреть блоги!!!!

**1.6 Подробнее о средствах разработки на Java**

Java Development Kit (**JDK**) является одним из трёх основных технологий, используемых в программировании на языке Java. К ним также относятся **JVM** (Java Virtual Machine) и **JRE** (Java Runtime Environment). Важно их различать, а **также понимать, как они связаны**:

* [JVM](https://topjava.ru/blog/what-is-the-jvm) отвечает за исполнение Java-программ;
* [JRE](https://topjava.ru/blog/what-is-the-jre) создает и запускает JVM;
* JDK позволяет разработчикам **создавать программы**, которые могут выполняться и запускаться посредством JVM и JRE;

Начинающие Java-разработчики часто путают JDK и JRE. Различие заключается в том, что **JDK представляет собой пакет инструментов** для ***разработки* программного обеспечения**, тогда как JRE представляет собой пакет инструментов для ***запуска*** **Java-кода**.

**1.6.1 Что такое JDK**

* Техническое определение: JDK — это реализация спецификации платформы Java, включающая в себя компилятор и библиотеки классов.
* **Обобщенное определение**: JDK — это программный пакет, который вы загружаете для создания Java-приложений.

В дополнение к JRE, который является средой, используемой для запуска Java-приложений, **каждый JDK содержит компилятор Java**.

Компилятор — это программа, способная принимать исходные файлы с расширением .java, которые являются обычным текстом, и превращать их в исполняемые файлы с расширением **class**.

Скоро мы рассмотрим, как работает компилятор, но сперва разберемся, как загрузить и настроить JDK.

Установка и настройка JDK не представляет никакой сложности. При загрузке JDK необходимо выбрать какую версию Java вы хотите использовать. На данный момент Java 8 является все еще наиболее часто используемой версией (хотя уже и вышла Java 19). **Java поддерживает обратную совместимость, поэтому просто загружайте последнюю версию**.

**1.6.2 Пакеты (платформы) JDK**

В дополнение к **версии** Java, необходимо выбрать пакет (платформу). В Java доступны следующие платформы: Java Enterprise Edition (Java EE), Java Standard Edition (Java SE) и Java Mobile Edition (Java ME).

Начинающие разработчики иногда не могут разобраться, какой именно пакет подходит для их проекта. Как правило, каждая версия JDK содержит инструменты Java SE.

Начинающих разработчиков не должен беспокоить выбор правильной версии JDK пакета, так как при необходимости, можно без проблем переключиться на другой пакет.

<https://timeweb.com/ru/community/articles/kak-ustanovit-java-development-kit>

**https://www.oracle.com/java/technologies/downloads/**

**https://www.oracle.com/java/technologies/downloads/#jdk19-windows**

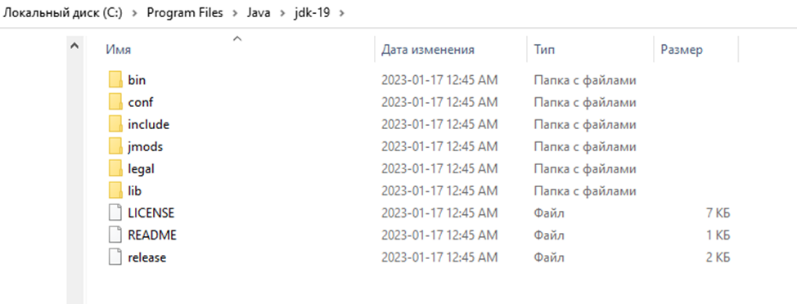
****

Рисунок 1.3. Установленная c сайта ORACLE JDK версии 19

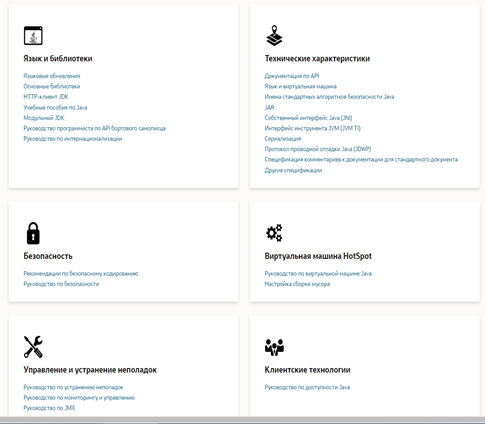


Рисунок 1.4. Часть справочной документации с сайта Oracle по JDK, переведённая на русский язык (появляется после загрузки JDK)

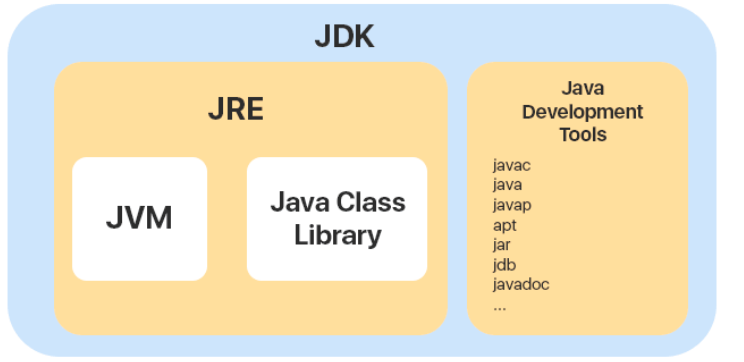


Рисунок 1.5. Структурная схема обеспечения разработки и выполнения программ на языке Java.

JDK содержит компилятор Java. Компилятор — это программа, способная принимать исходные файлы с расширением .java, которые являются обычным текстом, и превращать их в исполняемые файлы с расширением .class. Скоро мы рассмотрим, как работает компилятор, но сперва разберемся, как загрузить и настроить JDK.

<https://topjava.ru/blog/what-is-the-jdk>

**1.6.1 Что такое среда выполнения JRE?**

Для запуска и исполнения программы необходима специальная среда выполнения. Она загружает файлы классов, обеспечивает доступ к памяти и другим системным ресурсам.

В прошлом, большинство программ использовали операционную систему (ОС) в качестве среды выполнения. Программы запускались на любом компьютере, но для доступа к ресурсам полагались на параметры ОС. В качестве ресурсов выступали память, программные файлы и зависимости. С появлением среды выполнения Java все изменилось, по 1крайней мере, для Java-программ.

**1.6.2 Среда выполнения Java**

Программное обеспечение можно рассматривать **как набор слоев, расположенных поверх системного оборудования**. Каждый слой предоставляет службы, которые будут использоваться (и запрашиваться) слоями над ним.

**Среда выполнения Java — это уровень программного обеспечения, работающий поверх ОС и предоставляющий дополнительные службы, специфичные для Java**.

JRE сглаживает разнообразие ОС, **гарантируя, что программы Java могут работать практически на любой ОС без изменений**. Также автоматическое управление памятью — одна из самых важных функций JRE, гарантирующая, что программистам не придется вручную управлять распределением памяти.

Если коротко, то JRE — **это мета-OS для Java-программ**. Это классический пример *абстракции* — абстрагирование базовой ОС в согласованную платформу для запуска приложений Java.

**1.6.3 Как JRE работает с JVM**

Виртуальная машина Java — программное обеспечение, отвечающее за выполнение Java-программ. Пр. Ленина, 93

JRE — это программа, которая берет ваш Java-код, **объединяет его с необходимыми библиотеками и запускает JVM для его выполнения**.

JRE содержит программное обеспечение и библиотеки, которые требуются для работы вашей программы. **Например, загрузчик классов Java является частью JRE**. Эта важная часть программного обеспечения загружает скомпилированный Java-код в память и соединяет с соответствующими библиотеками.

В этом многоуровневом представлении JVM создается средой выполнения Java. С точки зрения пакета, JRE содержит JVM, как показано на рисунке 1.3.

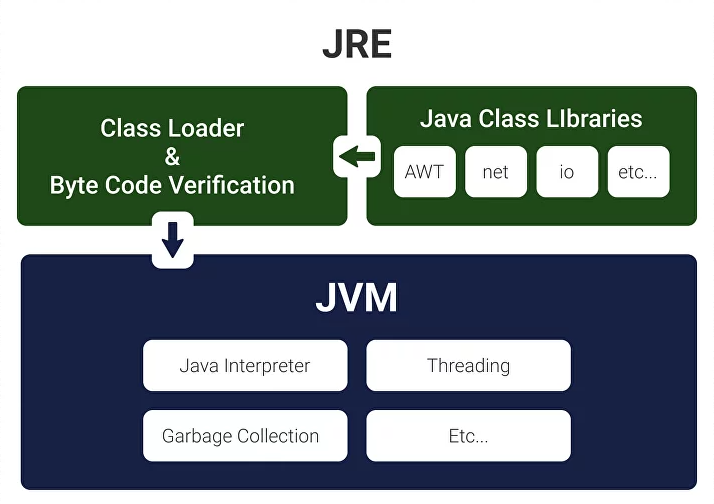


Рисунок 1.3. Структура состава и взаимодействие систем среды выполнения программы на языке Java