Реферат на тему:

«Язык программирования Java»

Выполнил:

Студент группы 21-ИВТЗ

НГТУ им.Алексеева

Зинкин Дмитрий Александрович

2022

**Оглавление**

[1. JAVA 3](#_Toc106111234)

[2. Как же это работает? 3](#_Toc106111235)

[3. История создания 3](#_Toc106111236)

[4. Основные особенности. 5](#_Toc106111237)

[5. Основные характеристики. 6](#_Toc106111238)

[6. Простота использования языка. 7](#_Toc106111239)

[7. Безопасность. 8](#_Toc106111240)

[8. Высокая производительность. 8](#_Toc106111241)

[9. Многопоточность. 9](#_Toc106111242)

[10. Вывод. 9](#_Toc106111243)

# **JAVA**

Технология Java – это объектно-ориентированная, платформо-независимая, многопоточная среда программирования. Это основа «умных» Web – и сетевых сервисов, она позволяет вам надежно и безопасно наращивать информационную структуру вашего предприятия благодаря платформенной независимости. Все виды систем могут взаимодействовать друг с другом – начиная со смарт карт и заканчивая суперкомпьютерами – независимо от аппаратной платформы и системного программного обеспечения.

# **Как же это работает?**

Когда программный продукт, написанный на ЯП Java, компилируется, он представляется в виде байт-кода.

Виртуальная машина Java может интерпретировать этот байт-код на любой платформе, на которой установлена виртуальная машина Java. Это означает, что нет необходимости в портировании программ.

Java выпущен компанией Sun Microsystems и официально выпущенный 23 мая 1995 года.

Так называют не только сам язык, но и платформу для создания приложений уровня предприятий на основе данного языка.

Изначально язык программирования назывался Oak и разрабатывался Джеймсом Гослингом для бытовой электротехники, но в последствии был переименован в Java и стал использоваться для написания клиентских приложений и серверного программного обеспечения. Назван он так в честь марки кофе, любимого программистами, поэтому на официальной эмблеме языка Java изображены чашка с дымящимся кофе.

# **История создания**

Язык Java зародился как часть проекта создания передового программного обеспечения (ПО) для различных бытовых приборов. Реализация проекта была начата на языке С++, но вскоре возник ряд проблем, наилучшим средством борьбы с которыми было изменение самого инструмента - языка программирования. Стало очевидным, что необходим платформо-независимый язык программирования, позволяющий создавать программы, которые не приходилось бы компилировать отдельно для каждой архитектуры и можно было бы использовать на различных процессорах под различными операционными системами.

Рождению языка Java предшествовала довольно интересная история. В 1990 году разработчик ПО компании Sun Microsystems Патрик Нотон (Patrick Naughton) понял, что ему надоело поддерживать сотни различных интерфейсов программ, используемых в компании, и сообщил [исполнительному директору](https://pandia.ru/text/category/direktor_ispolnitelmznij/) Sun Microsystems и своему другу Скотту МакНили (Scott McNealy) о своем намерении перейти работать в компанию NeXT. МакНили, в свою очередь, попросил Нотона составить список причин своего недовольства и выдвинуть такое решение проблем, как если бы он был Богом и мог исполнить все, что угодно.

Нотон, хотя и не рассчитывал на то, что кто-то обратит внимание на его письмо, все же изложил свои претензии, беспощадно раскритиковав недостатки Sun Microsystems, в частности, разрабатываемую в тот момент архитектуру ПО NeWS. К удивлению Нотона, его письмо возымело успех: оно было разослано всем ведущим инженерам Sun Microsystems, которые не замедлили откликнуться и высказать горячую поддержку своему [коллеге](https://pandia.ru/text/category/koll/) и одобрение его взглядов на ситуацию в Sun Microsystems. Обращение вызвало одобрение и у высшего руководства компании, а именно, у Билла Джоя (Bill Joy), основателя Sun Microsystems, и Джеймса Гослинга (James Gosling), Нотона.

После команда из шести человек, с кодовым названием Green, ушла в самовольное изгнание, погрузившись в исследования бытовых устройств, таких как Nintendo Game Boys, устройств [дистанционного управления](https://pandia.ru/text/category/distantcionnoe_upravlenie/). Команда Green пыталась найти средство, с помощью которого можно было бы установить взаимодействие между этими устройствами. Вскоре стало ясно, что такие электроприборы, как [видеомагнитофоны](https://pandia.ru/text/category/videomagnitofon/), проигрыватели лазерных дисков, стереосистемы - все они были реализованы на разных процессорах. Это означало, если производитель захочет добавить телевизору или видеомагнитофону дополнительные функции или характеристики, он будет зажат в рамках средств, зашитых в аппаратное обеспечение. Эта проблема, в сочетании с ограниченностью памяти микросхем этих устройств, выдвинула новый подход к программированию ПО, который должен был стать ведущим на рынке бытовой электроники.

Вскоре компания Sun Microsystems преобразовала команду Green в компанию First Person. Новая компания обладала интереснейшей концепцией, но не могла найти ей подходящего применения. После ряда неудач неожиданно ситуация для компании резко изменилась: был анонсирован Mosaic - так родился World Wide Web, с которого началось бурное развитие Internet.

Нотон предложил использовать Oak в создании Internet- приложений. Так Oak стал самостоятельным продуктом, вскоре был написан Oak-компилятор и Oak-браузер "WebRunner". В 1995 году компания Sun Microsystems приняла решение объявить о новом продукте, переименовав его в Java (единственное разумное объяснение названию - любовь программистов к кофе). Когда Java оказалась в руках Internet, стало необходимым запускать Java-аплеты - небольшие программы, загружаемые через Internet. WebRunner был переименован в HotJava и компания Netscape встала на поддержку Java-продуктов.

# **Основные особенности.**

Программы на Java транслируются в байт-код, выполняемый виртуальной java-машиной (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор, но с тем отличием, что байтовый код, в отличие от текста, обрабатывается значительно быстрее. Достоинство подобного способа выполнения программ — в полной независимости байт-кода от ОС и оборудования, что позволяет выполнять Java-приложения на любом устройстве, которое поддерживает виртуальную машину.

Так же приложения переносимы на многие платформы, но об этом говорилось выше.

Java является объектно-ориентированным и одновременно простым языком программирования.

Цикл разработки программных средств с использованием Java значительно сокращается в силу того, что Java - интерпретируемый язык. Процесс компиляции-сборки-загрузки устарел - теперь программу надо только откомпилировать и сразу запускать.

Приложения надежны: Java контролирует обращения к памяти.

Приложения высокопроизводительны: несмотря на то, что язык Java - интерпретируемый, код Java программы оптимизируется до фазы исполнения.

Поддержка системы многопоточности позволяет создавать параллельно исполняемые взаимодействующие легковесные процессы.

Приложения настраиваемы под изменяющееся окружение: возможна динамическая загрузка программных модулей из любого места в сети.

# **Основные характеристики.**

Развитие Internet заставляет совершенно по-новому рассматривать процессы разработки и распределения программного обеспечения. Для того, чтобы выжить в мире электронного бизнеса и распространения данных, язык Java должен быть:

* безопасным,
* высокопроизводительным,
* надежным.

Работа на различных платформах гетерогенных сетей отбрасывает традиционную схему распределения ПО, версий ПО, модификации ПО, объединения ПО и т. д. Для решения проблем гетерогенных сред язык должен быть:

* нейтральным к архитектуре,
* переносимым,
* динамически подстраиваемым.

Разработчики Java с самого начала хорошо понимали, что язык, предназначенный для решения проблем гетерогенных сред, также должен быть:

* простым - его должны с легкостью использовать все разработчики
* ясным - разработчики должны без больших усилий выучить Java
* объектно-ориентированным - он использует все преимущества современных методологий разработки ПО и подходит для написания распределенных клиент-серверных приложений
* многопоточным - для обеспечения высокой производительности приложений, выполняющих одновременно много действий (например, в мультимедийных системах)
* интерпретируемым - для переносимости и большей динамичности.

# **Простота использования языка.**

Простота языка входит в ключевые характеристики Java: разработчик не должен длительное время изучать язык, прежде чем он сможет на нем программировать. Фундаментальные концепции языка Java быстро схватываются, и программисты с самого начала могут вести продуктивную работу. Разработчиками Java было принято во внимание, что многие программисты хорошо знакомы с языком С++, поэтому Java, насколько это возможно, приближен к С++.

Добавилась автоматическая сборка мусора, упрощающая процесс программирования, но несколько усложняющая систему в целом. В С и С++ управление памятью вызывало всегда массу проблем, теперь же об этом не придется много заботиться.

Но не известно к счастью или, к сожалению, в Java не включены редко используемые возможности С++, которые усложняли работу. Отказались от перегрузки операторов, но перегрузка методов осталась.

# **Безопасность.**

Java разработана для оперирования в распределенных средах, это означает, что на первом плане должны стоять вопросы безопасности. Средства безопасности, встроенные в язык, и система исполнения Java позволяют создавать приложения, на которые невозможно "напасть" извне. В сетевых средах приложения, написанные на Java, защищены от вторжения неавторизованного кода, пытающегося внедрить вирус или разрушить файловую систему.

# **Высокая производительность.**

Производительность всегда заслуживает особого внимания. Java достигает высокой производительности благодаря специально оптимизированному байт-коду, легко переводимому в машинный код. Автоматическая сборка мусора выполняется как фоновый поток с низким приоритетом, обеспечивая высокую вероятность доступности требуемой памяти, что ведет к увеличению производительности. Приложения, требующие больших вычислительных ресурсов, могут быть спроектированы так, чтобы те части, которые требуют интенсивных вычислений, были написаны на языке ассемблера и взаимодействовали с Java платформой.

# **Многопоточность.**

Для эффективной работы с потоками в Java реализован механизм семафоров и средств синхронизации потоков: библиотека языка предоставляет класс Thread, а система выполнения предоставляет средства диспетчеризации и средства, реализующие семафоры. Важно, что работа параллельных потоков с высокоуровневыми системными библиотеками Java не вызовет конфликтов: функции, предоставляемые библиотеками, доступны любым выполняющимся потокам.

# **Вывод.**

Благодаря Java работа по разработке программного обеспечения значительно упрощается, все старания направлены на достижение конечной цели.