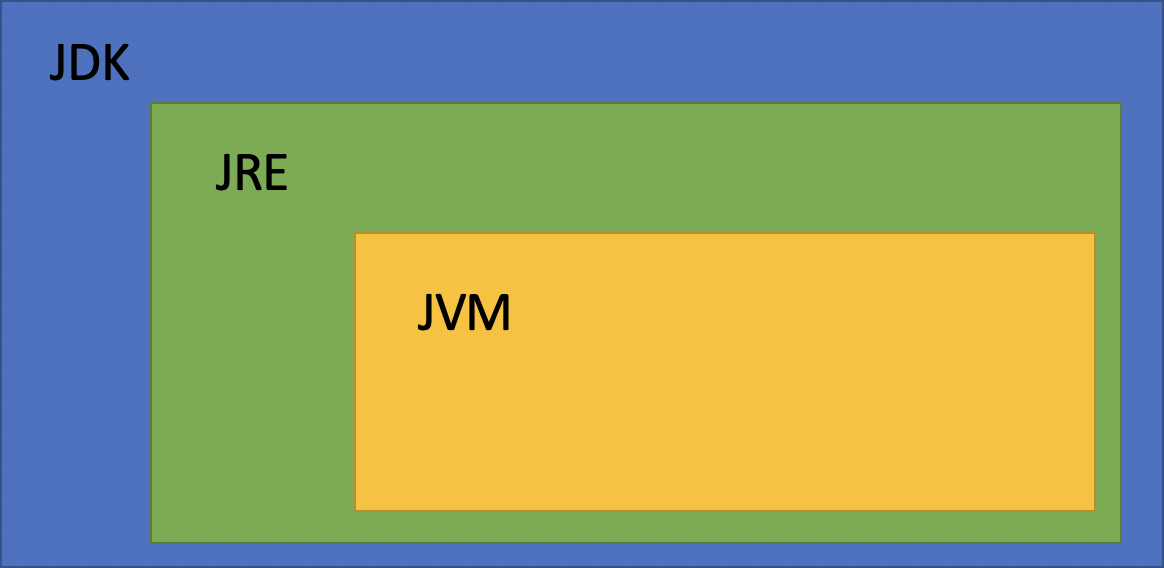
1. **Объясните понятия JRE, JDK, JavaSE, JavaME, JavaEE.**

JRE является реализацией JVM, которая предоставляет платформу для выполнения Java-программ. JRE состоит из виртуальной машины Java, бинарных файлов и других классов. JRE не содержит инструменты для разработки (компилятор Java, отладчик и т.д). Если вы хотите запустить любую Java программу, вы должны установить JRE.

JRE - это Java Runtime Environment. В переводе по-русски JRE - это среда выполнения для языка Java. Это минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора и других средств разработки. JRE состоит из виртуальной машины — Java Virtual Machine — и библиотеки Java-классов.

Набор программ и пакетов классов JRE содержит все необходимое для выполнения байт-кодов, в том числе интерпретатор java (в прежних версиях — облегченный интерпретатор jre) и библиотеку классов. Это часть JDK, не содержащая компиляторы, отладчики и другие средства разработки. Именно Oracle JRE или его аналог, созданный другими фирмами, присутствует в тех браузерах, которые умеют выполнять программы на Java, в операционных системах и системах управления базами данных.

JDK — это Java Development Kit. Это бесплатно распространяемый комплект разработчика приложений на языке Java. Он включает в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE). Распространитель комплекта - это компания Oracle Corporation (ранее называлась Sun Microsystems). В состав JDK не входит интегрированная среда разработки на Java, поэтому разработчик, использующий только JDK, вынужден использовать внешний текстовый редактор и компилировать свои программы, используя утилиты командной строки.



##### **Чем отличаются среда JRE и пакет JDK?**

|  |  |
| --- | --- |
| **JRE  (Среда выполнения Java)** | **JDK (Инструментальный пакет для разработчиков Java)** |
| Это реализация виртуальной машины Java\*, в которой фактически выполняются Java-программы. | Это пакет программного обеспечения, которое можно использовать для разработки приложений на основе Java. |
| Среда выполнения Java — это подключаемый модуль, необходимый для выполнения Java-программ. | Инструментальный пакет для разработчиков Java нужен для разработки Java-приложений. |
| JRE имеет меньший объем, нежели JDK, поэтому требует меньше места на диске. | Пакету JDK необходимо больше места на диске, так как в нем содержится среда JRE вместе с различными средствами разработки. |
| Возможность бесплатной загрузки и поддержки JRE предоставляется на веб-сайте [java.com](http://www.java.com/) | Возможность бесплатной загрузки и поддержки JDK предоставляется на веб-сайте [oracle.com/technetwork/java/javase/downloads/](http://www.oracle.com/technetwork/java/javase/downloads/index.html) |
| Содержит JVM, библиотеки ядра и другие дополнительные компоненты для выполнения приложений и апплетов, написанных на Java. | Содержит JRE, набор классов API, компилятор Java, Webstart и дополнительные файлы, необходимые для создания апплетов и приложений на Java. |

##### **Что такое Java Enterprise Edition (Java EE)?**

[**Java EE**](http://www.oracle.com/technetwork/java/javaee/index.html) — независимая от платформы, ориентированная на Java среда для разработки, построения и развертывания корпоративных веб-приложений в интерактивном режиме. Java EE содержит несколько компонентов Java Standard Edition (Java SE). Платформа Java EE состоит из набора служб, API-интерфейсов и протоколов, предоставляющих функциональные возможности для разработки многоуровневых веб-приложений.

Java EE упрощает разработку приложений и снижает потребность в программировании и обучении программированию, обеспечивая создание стандартизованных, повторно используемых модульных компонентов и поддерживая уровень автоматической обработки многих прикладных аспектов программирования.

**Java Standard Edition (Java SE)** – это стандартное издание Java, именно под него разрабатываются клиентские приложения. Приложения могут работать самостоятельно или в качестве апплетов веб-браузера.

**Java Micro Edition (ME)** - редакция для разработки приложений для микрокомпьютеров (мобильных телефонов, Palm и т.д.). Сейчас получила широкую популярность, в связи с развитием мобильных микропроцессорных устройств. В нее входят "облегченные" стандартные классы и классы для написания мидлетов (Midlets). Мидлеты - это аналоги аплетов, но только приспособленные специально для небольших устройств. В них также поддерживается графика, звук, реакция на события (нажатие кнопок и т.д.). Java ME наверно наиболее полно соответствует начальному предназначению Java - платформы для написания программ для бытовых устройств.

1. **Перечислите основные свойства и преимущества платформы Java**.

В этом документе приведено одиннадцать характерных особенностей Java. Этот язык:

* простой;
* объектно-ориентированный;
* распределенный;
* надежный;
* безопасный;
* не зависящий от архитектуры компьютера;
* переносимый;
* интерпретируемый;
* высокопроизводительный;
* многопоточный;
* динамичный.

**+ Объектно-ориентированное программирование**

Java включает в себя **объектно-ориентированное программирование** (OOP) — концепцию, в которой вы не только определяете тип данных и его структуру, но и набор функций, применяемых к нему. Таким образом, структура данных становится объектом, которым можно управлять для создания отношений между различными объектами.

В чем плюсы ООП?

* При ООП можно повторно использовать объекты в других программах
* ООП предотвращает ошибки, поскольку объекты скрывают информацию, к которой не должно быть доступа
* ООП более эффективно организует структуру программ, в том числе больших
* ООП упрощает обслуживание и модернизацию старого кода

**+ Java — язык высокого уровня с простым синтаксисом и плавной кривой обучения**

Java — это язык высокого уровня, то есть он похож на человеческий язык. В отличие от языков низкого уровня, которые напоминают машинный код. Языки высокого уровня преобразуется с помощью компиляторов или интерпретаторов. Это упрощает разработку, делая язык более легким для написания, чтения и обслуживания.

Синтаксис Java основан на C ++, поэтому Java похожа на C. Тем не менее, синтаксис Java проще, что позволяет новичкам быстрее учиться и эффективнее использовать код для достижения конкретных результатов.

**+ Стандарт для корпоративных вычислительных систем**

Корпоративные приложения — главное преимущество Java с 90-х годов, когда организации начали искать надежные инструменты программирования не на C. Java поддерживает множество библиотек — строительных блоков любой корпоративной системы. Библиотеки помогают разработчикам создавать любые функции, которые могут понадобиться компании. Java широко распространен — это язык, который преподают в рамках введения в программирование в большинстве школ и университетов. Возможности интеграции Java впечатляют: большинство хостинг-провайдеров поддерживают Java. Более того, Java — язык, дешевый в обслуживании: работать с Java можно с любого компьютера, вне зависимости от конкретной аппаратной инфраструктуры.

**+ Безопасность**

Существует мнение, что Java — безопасный язык, однако это не совсем так. Сам язык не защищает вас от уязвимостей, но некоторые его функции устраняют распространенные уязвимости. Во-первых, в отличие от C, в Java нет указателей. Указатель — это объект, который сохраняет адрес ячейки памяти другого значения, что может вызвать несанкционированный доступ к памяти. Во-вторых, в Java есть **Security Manager**, созданная для каждого приложения политика безопасности, в которой можно указать правила доступа. Это позволяет запускать приложения Java в «песочнице» и устранять таким образом уязвимости.

**+ Независимость от платформы («Написать один раз и использовать везде»)**

«Написать один раз и использовать везде» (WORA) — популярная в IT-сфере фраза, с помощью которой Sun Microsystems описывает кросс-платформенные возможности Java. Можно создать Java-приложение на Windows, скомпилировать его в байт-код и запустить его на любой другой платформе, поддерживающей виртуальную машину Java (JVM). Таким образом, JVM служит уровнем абстракции между кодом и оборудованием.

**+ Язык для распределенного программирования и комфортной удаленной совместной работы**

Java создавался как язык для распределенного программирования: он имеет встроенный механизм совместного использования данных и программ несколькими компьютерами, что повышает производительность и эффективность труда.

**+ Автоматическое управление памятью**

Разработчикам Java не нужно вручную писать код для управления памятью благодаря автоматическому управлению памятью (AMM). AMM также используется [в языке программирования Swift](https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-swift-programming-language/) и при очистке памяти приложениями, которые автоматически обрабатывают распределение и освобождение памяти. Что именно это означает?

Эффективность программы напрямую связана с памятью. При этом объем памяти ограничен. При написании приложения на языках с ручным управлением памятью, разработчики рискуют забыть выделить память, что приведет к увеличению объема занимаемой приложением памяти и проблемам с производительностью. **Программы очистки памяти** ищут объекты, которые больше не используются программой, и удаляют их. Это влияет на работу процессора, однако умная оптимизация и настройка позволяют снизить это влияние.

**+ Многопоточность**

Поток — наименьшая единица обработки в программировании. Чтобы максимально эффективно использовать время процессора, Java позволяет запускать потоки одновременно, что называется многопоточностью.

Потоки используют одну и ту же область памяти, поэтому между ними можно быстро переключаться. Потоки независимы друг от друга: один поток не влияет на работу других потоков. Это особенно полезно в играх и программах с большим объемом анимации.

**+ Стабильность и сообщество**

Уже много лет развитию Java способствуют сообщество, поддержка Oracle и изобилие приложений и языков на JVM. Кроме того, постоянно выпускаются новые версии Java [с новыми интересными функциями](https://aboullaite.me/10-new-features-in-java-10/).

Сообщество разработчиков Java не имеет себе равных. Около 45% респондентов опроса [StackOverflow 2018](https://insights.stackoverflow.com/survey/2018/" \t "_blank) используют Java. У Java чрезвычайно большая экосистема хорошо протестированных библиотек и фреймворков для любых задач. Начинающий разработчик, скорее всего, выберет Java: на тему Java-программирования существует более 1000 курсов на Udemy и более 300 на Coursera.

1. **Является ли язык Java компилируемым или интерпретируемым?**

Обычно Java видится разработчику как язык, компилируемый в файлы .class до того, как код будет использоваться на JVM (виртуальной машине Java). Призадумавшись, многие разработчики также объясняют, что работа байт-кода начинается с интерпретации виртуальной машиной JVM, но несколько позже код также проходит динамическую компиляцию (JIT). Но на этом многие специалисты начинают «плыть», выстраивая несколько надуманную концепцию о том, что байт-код, в сущности, является машинным кодом для воображаемого или упрощенного процессора.

На самом деле байт-код виртуальной машины Java можно считать переходной формой между исходным кодом, пригодным для чтения человеком, и машинным кодом. В техническом отношении с точки зрения теории компиляции байт-код — это действительно своеобразный промежуточный язык (intermediate language), а не настоящий машинный код. Это означает, что процесс преобразования исходного кода Java в байт-код не является компиляцией в том смысле, в каком она понимается в языках C и С++. В свою очередь, javac не назовешь таким же компилятором, как gcc. В сущности, это генератор файлов классов для обработки исходного кода Java. Настоящим компилятором в экосистеме Java является динамический компилятор (JIT).

Некоторые специалисты (в том числе программисты Oracle) характеризуют систему Java как «динамически компилируемую». При этом акцентируется тот факт, что истинная компиляция — это динамическая компиляция во время исполнения, а не создание файла класса во время исполнения.

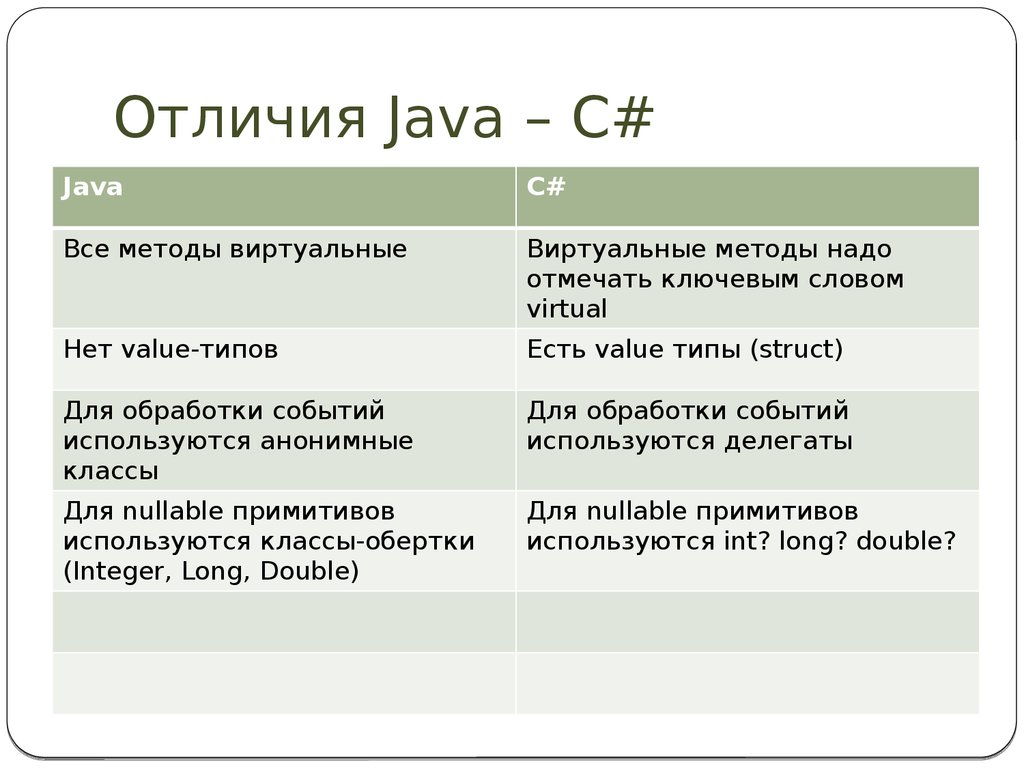
Поэтому верным ответом на вопрос «**Java — это компилируемый или интерпретируемый язык?**» будет: «**И такой и такой**»!

1. **Что такое механизм автоматической сборки мусора (garbage collector)?**

Java-программисту не нужно следить за распределением памяти, так как сборщик мусора управляет памятью автоматически. Сборщик мусора запускается виртуальной машиной Java (JVM). Сборщик мусора — это низкоприоритетный процесс, который запускается периодически и освобождает память, использованную объектами, которые больше не нужны.

## Запуск сборщика мусора

JVM обычно запускает сборщик мусора при низком уровне свободной памяти. Но работа сборщика мусора не гарантирует, что всегда будет оставаться достаточно свободной памяти. Если памяти недостаточно даже после восстановления, JVM генерирует исключение OutOfMemoryError. Обратите внимание, что перед генерированием исключения JVM обязательно запускает сборщик мусора как минимум 1 раз. Вы можете запросить запуск сборщика мусора, но вы не можете принудительно задавать это действие.

**5. В чем сходства и различия Java, C/C++, С#?** 



**6. Почему Java является платформой, а не языком программирования?**

Платформа [Java](https://ru.bmstu.wiki/Java) - это набор программ, которые облегчают разработку и запуск программ, написанных на языке программирования [Java](https://ru.bmstu.wiki/Java) . Платформа [Java](https://ru.bmstu.wiki/Java) будет включать механизм выполнения (называемый виртуальной машиной), компилятор и набор библиотек ; могут быть также дополнительные серверы и альтернативные библиотеки, которые зависят от требований. [Java](https://ru.bmstu.wiki/Java) не относится ни к одному процессору или операционной системе, поскольку платформы [Java](https://ru.bmstu.wiki/Java) были реализованы для широкого спектра аппаратных и операционных систем, чтобы позволить программам [Java](https://ru.bmstu.wiki/Java) работать одинаково на всех из них.

**7. Перечислите основные категории программ Java.**



**8. Какие существуют IDE для разработки программ?**

IDE – это Integrated Development Environment– Интегрированная Среда Разработки.

Для написания Java-программ существуют 3 популярных IDE:

* IntelliJ IDEA
* Eclipse
* Net Beans

Почти все программисты отдают предпочтение IntelliJ IDEA.