Оглавление

[1. Что такое виртуальная машина? 2](#_Toc34835599)

[2. К какому типу языка программирования относится Java? 2](#_Toc34835600)

[3. Из каких компонентов состоит Java (JDK, JRE, JVM)? 3](#_Toc34835601)

[4. Для чего используется JDK? 3](#_Toc34835602)

[5. Для чего используется JRE? 4](#_Toc34835603)

[6. Для чего используется VM? 4](#_Toc34835604)

[7. Расскажите про примитивные типы 4](#_Toc34835605)

[8. Опишите шаги для компиляции и запуска приложения в консоли (javac java). 6](#_Toc34835606)

[9. Что такое "оператор условия"? 6](#_Toc34835607)

[10. Какие типы операторов условия существуют? 6](#_Toc34835608)

[11. Расскажите про булевы операции || &&? Что такое short-circuit? 7](#_Toc34835609)

[12. Что такое тернарное условие? 7](#_Toc34835610)

[Тернарный оператор (или тернарная операция) в языке Java часто используется вместо оператора if-else. Он содержит три выражения - отсюда и происходит название оператора. 7](#_Toc34835611)

[Общая форма тернарного оператора: 7](#_Toc34835612)

[выражение1 ? выражение2 : выражение3 7](#_Toc34835613)

[13. Что такое циклы и для чего они используются? 7](#_Toc34835614)

[14. Для чего используется цикл for? 7](#_Toc34835615)

[15. Для чего используется цикл foreach? 8](#_Toc34835616)

[16. Для чего используется цикл while? 8](#_Toc34835617)

[17. Для чего используется цикл do while? 8](#_Toc34835618)

[18. Что такое массив? 9](#_Toc34835619)

[19. Как создать массив? 9](#_Toc34835620)

[20. Как присвоить значение ячейке массива? 9](#_Toc34835621)

[21. Как можно пройти по всем элементам массива? 9](#_Toc34835622)

[22. Как можно найти элемент в массиве? 9](#_Toc34835623)

[23. Что будет, если обратиться к несуществующей ячейки массива? 9](#_Toc34835624)

[24. Как удалить ячейку в массиве? 9](#_Toc34835625)

[25. Как отредактировать ячейку в массиве? 10](#_Toc34835626)

# Что такое виртуальная машина?

Java Virtual Machine (сокращенно Java VM, JVM) — 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 может также использоваться для выполнения программ, написанных на других языках программирования. Например, исходный код на языке Ada может быть скомпилирован в байт-код Java, который затем может выполниться с помощью JVM.

JVM является ключевым компонентом платформы Java. Так как виртуальные машины Java доступны для многих аппаратных и программных платформ, Java может рассматриваться и как связующее программное обеспечение, и как самостоятельная платформа. Использование одного байт-кода для многих платформ позволяет описать Java как «скомпилировано однажды, запускается везде» (compile once, run anywhere).

Виртуальные машины Java обычно содержат интерпретатор байт-кода, однако, для повышения производительности во многих машинах также применяется JIT-компиляция часто исполняемых фрагментов байт-кода в машинный код.

# К какому типу языка программирования относится Java?

Java - строго типизированный язык(НО ПРИ ЭТОМ ЕСТЬ var, который может быть объявлен локальной переменной. Тип выводится компилятором из присваемого значения). Каждая переменная в Java имеет конкретный тип, который определяет размер и размещение её в памяти; диапазон значений, которые могут храниться в памяти; и набор операций, которые могут быть применены к переменной.

Java – язык свободной формы относительно пробелов (отступов а-ля pyton)

Java — объектно-ориентированный язык программирования. Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее.

# Из каких компонентов состоит Java (JDK, JRE, JVM)?

**JVM** (Java Virtual Machine) - 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) - 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) - комплект разработчика приложений на языке Java, включающий в себя компилятор, стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему JRE.

# Для чего используется JDK?

это бесплатно распространяемый корпорацией Sun комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE). В состав JDK не входит интегрированная среда разработки на Java (IDE), поэтому разработчик, использующий только JDK, вынужден использовать внешний текстовый редактор и компилировать свои программы, используя утилиты командной строки.

Дополнительные возможности уже предоставляют современные интегрированные среды разработки на Java, такие, как NetBeans, Sun Java Studio Creator, IntelliJ IDEA, Borland JBuilder. Они опираются на сервисы, предоставляемые JDK, и вызывают для компиляции Java-программ компилятор командной строки из комплекта JDK. Поэтому эти среды разработки либо включают в комплект поставки одну из версий JDK, либо требуют для своей работы предварительной установки JDK на машине разработчика.

С некоторых пор фирма Sun предоставляет полные исходные тексты JDK, включая исходные тексты самого Java-компилятора.

# Для чего используется JRE?

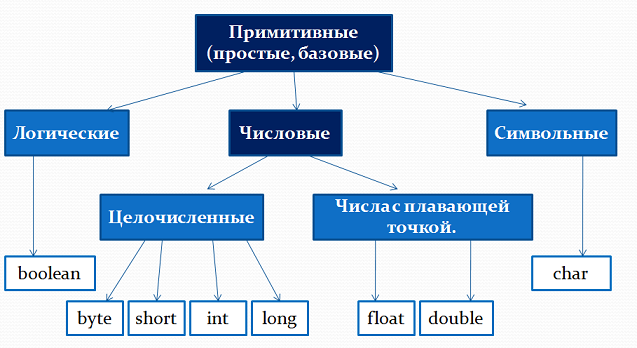
это исполнительная среда Java в которой выполняются программы, написанные на этом языке. Среда состоит из виртуальной машины – Java Virtual Machine(JVM) и библиотеки Java классов. JRE является частью JDK. По сути это минимальная реализация виртуальной машины, необходимая для исполнения Java приложений, без компилятора и других средств разработки. Именно JRE или его аналог других фирм используется в браузерах, умеющих выполнять программы на Java, операционных системах и системах управления базами данных. Хотя JRE входит в состав JDK, фирма SUN распространяет этот набор и отдельным файлом. Это вызвано тем, что установка JRE является необходимым и достаточным условием для выполнения Java программ. Однако для разработки программ JRE недостаточно, необходимо установить JDK, который может установить и JRE и дополнительные компоненты.

# Для чего используется VM?

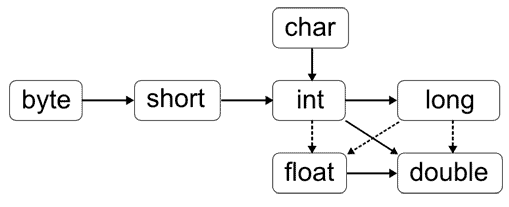
Виртуальная машина Java интерпретирует и исполняет байт-код Java. Байт код получают посредством компиляции исходного кода программы с помощью компилятора Java (стандартный - javac). В отличие от классических runtime-библиотек, библиотеки Java-классов входят в состав JRE.

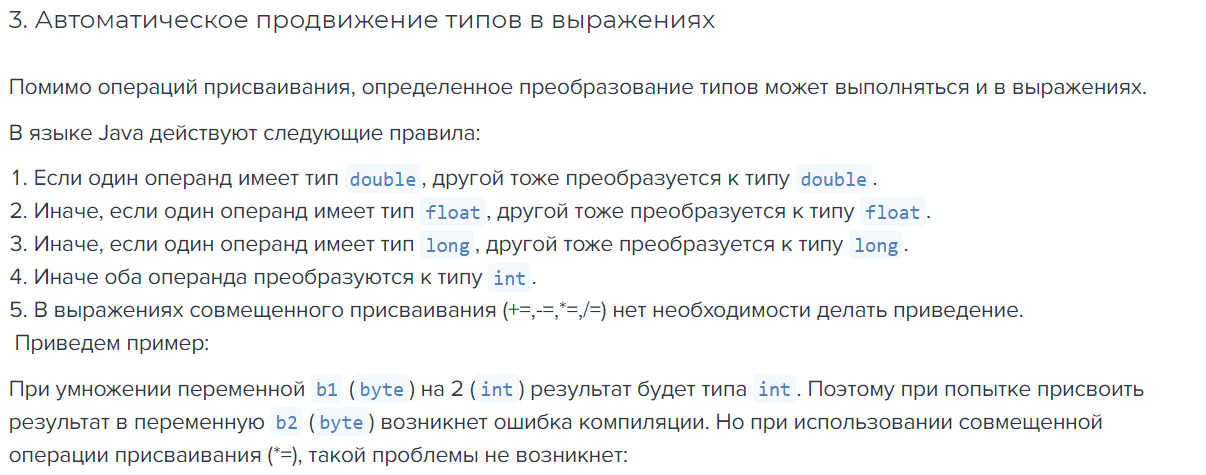
1. Расскажите про примитивные типы.

Примитивный (встроенный, базовый) тип — тип данных, предоставляемый языком программирования как базовая встроенная единица языка. Занимает память в стэке.







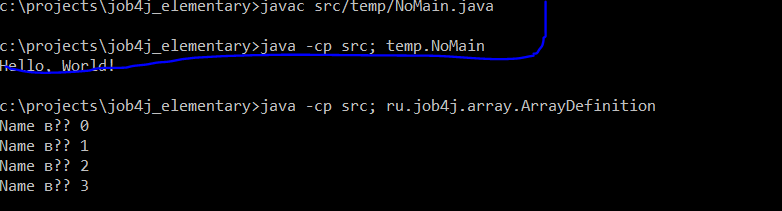


# Опишите шаги для компиляции и запуска приложения в консоли (javac java).

Для того, чтобы работать из консоли нужно изменить переменную окружения path. В этой переменной хранятся пути записанные через точку с запятой, в которых оболочка(консоль) будет искать исполняемые файлы(\*exe, \*.bat, и другие). Допишите в эту переменную ;C:Program FilesJavajdk1.6.0\_17bin.

Cначала вызываем javac classname.java

Создается файл класс, который далее запускается через java.



# Что такое "оператор условия"?

Операторы, которые позволяют выполнить определенный блок кода и пропустить другой.

# Какие типы операторов условия существуют?

If, else.

Switch case

Тернарные операторы

# Расскажите про булевы операции || &&? Что такое short-circuit?

ИЛИ/ И.

ИЛИ выполняет когда хотя бы одно верно, И когда верны оба.

short-circuit – операторы короткого замыкания. Это значит, что этим операторы проверяются только первую часть. И если она не верна, то вторую даже не проверяют. То есть это как раз && ||.

В отличие от них, операторы | & всегда проверяют два условия, вне зависимости от результатов в первой части – но это вроде как обычно только в битовых операциях используется.

# Что такое тернарное условие?

# Тернарный оператор (или тернарная операция) в языке Java часто используется вместо оператора if-else. Он содержит три выражения - отсюда и происходит название оператора.

# Общая форма тернарного оператора:

# выражение1 ? выражение2 : выражение3

# Что такое циклы и для чего они используются?

Циклы позволяют многократно выполнить блок операций или тела цикла.

В большинстве циклов указывается условие, которое проверяется.

# Для чего используется цикл for?

В основном мы используем цикл for, когда нам нужно работать со счетчиком, то есть знать например позицию массива, который мы пробегаем.

начало i = 0 Выполняется один раз при входе в цикл

условие i < 3 Проверяется перед каждой итерацией цикла. Если оно вычислится в false, цикл остановится.

шаг i++ Выполняется после тела цикла на каждой итерации перед проверкой условия.

тело alert(i) Выполняется снова и снова, пока условие вычисляется в true.

# Для чего используется цикл foreach?

На каждом последующем шаге итерации выбирается следующий элемент массива, который сохраняется в переменной цикла. Цикл завершается, когда все элементы массива окажутся выбранными. Цикл foreach позволяет проходить по каждому элементу коллекции

# Для чего используется цикл while?

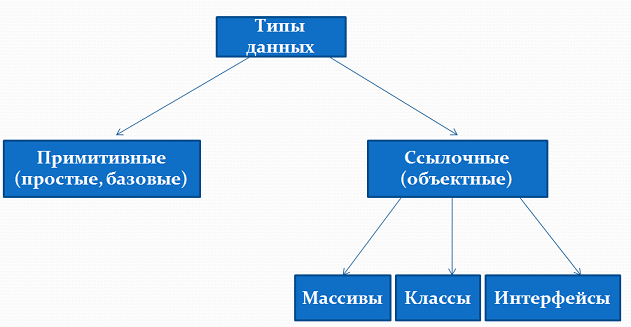
Код из тела цикла выполняется, пока условие истинно.

Можно считать аналогом for, но без реализации начало и шага цикла. Только условие.

# Для чего используется цикл do while?

Когда нам нужна минимум одна иттерация, например перед проверкой.

# Что такое массив?



Структура данных. Имеет ссылочный тип. Значения, записанные в массиве хранятся в куче Heap, а ссылка на область в памяти в стэке.

# Как создать массив?

Через оператор new.

# Как присвоить значение ячейке массива?

Используя синтаксис: имя массива [номер ячейки]

# Как можно пройти по всем элементам массива?

Через любой цикл. Например foreach.

# Как можно найти элемент в массиве?

Пробежаться по нему циклом или можно воспользоваться спец. Классом Arrays.binarysearch.

# Что будет, если обратиться к несуществующей ячейки массива?

Вылетит ошибка index out of bounds.

# Как удалить ячейку в массиве?

Никак.

# Как отредактировать ячейку в массиве?

Так же как и присвоить значение.