Java introduction

**Преимущества языка Java:**

1. Java – кроссплатформенный язык программирования. Один и тот же код можно запускать под управлением различных ОС: Linux, Windows, Solaris…Полезно, в случае , когда программы запускаются через интернет для последующего выполнения на разного рода ОС.
2. Java – это компилируемый язык программирования .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Есть 2 типа языков программирования:

- компилируемый – программа – компилятор преобразует исходный код в машинный код с особым заголовком или/и расширением, для последней идентификации этого файла, как исполняемого, ОС;

Интерпретируемый – для запуска программного кода на соответствующей ОС нужна программа интерпретатор;

Классификация языков программирования на компилируемые и интерпретируемые, является неточной и весьма условной, поскольку для любого языка программирования может быть создан как компилятор, так и интерпретатор.

Языки программирования принято разделять на компилируемые и интерпретируемые в силу типичных различий:

* скорость выполнения программы, скомпилированной в [машинный код](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), превосходит скорость интерпретируемой программы, как правило, в десятки и сотни раз;
* в случае использования компилятора, при внесении изменений в исходный код программы, прежде чем эти изменения можно будет увидеть в работе программы, необходимо выполнить компиляцию исходного текста.

Компилируемые языки обычно позволяют получить более быструю и, возможно, более компактную программу, и поэтому применяются для создания часто используемых программ.

1. синтаксис языка Java очень поход на синтаксис С++
2. Java – это полностью объектно-ориентированный язык программирования. Все сущности в языке Java являются объектами.
3. Java покрывает все слабые стороны языка С++:

- исключена возможность явного выделения и освобождения памяти: память в языке Java освобождается автоматически с помощью механизма сборки мусора;

- исключается возможность стереть данные из памяти вследствие неправильного использования указателей;

- исключена возможность перепутать операторы присваивания и равенства;

- исключено множественное наследование(в Java есть такое понятие как интерфейс).

**Характерные особенности языка Java**

*Простой*

*- синтаксис практически повторяет синтаксис языка С++:switch;*

*- краткость. Р*азмер основного интерпретатора и средств поддержки классов составляет около 40 Кбайт; стандартные библиотеки и средства поддержки потоков (особенно номное микроядро (self-contained microkernel)) занимают еще 17: Кбайт.  
Это огромный успех. Заметим, однако, что библиотеки средств поддержки графического пользовательского интерфейса значительно крупнее.

*Интерпретируемый*

Интерпретатор языка Java может пересылаться на любую машину и выполнять байт-код непосредственно на ней.

*Распределенный*

Java обладает большой библиотекой программ для передачи данных на основе таких протоколов TCP/IP (Transmission Control Protocol/Internet Protocol-протокол управления передачей/Интернет-протокол), как HTTP (Hypertext Transfer Protocol - протокол передачи гипертекстов) или FTP (File Transfer Protocol-протокол передачи файлов). Приложения, написанные на языке Java, могут открывать объекты и получать к ним доступ через сеть с помощью иURL-адресов (Uniform Resource Location - универсальный адрес ресурса) так желегко, как авлокальной сети.  
*Надежный*. Основное внимание в языке Java уделяется раннему обнаружению возможных ошибок, динамической проверке (во время выполнения программы), а также исключению ситуаций, подверженных ошибкам. Компилятор языка Java выявляет такие ошибки, которые в других языках обнаруживаются только на этапе выполнения программы. Кроме того, программисты, потратившие многие часы на поиски ошибки, вызвавшей повреждение памяти из-за неверного указателя, будут очень рады тому, что в языке Java такие проблемы возникнуть в принципе не могут.  
*Безопасный*  
*Машинонезависимый*

*В отличие от языков С и C++, в спецификации Java нет аспектов, зависящих от системы реализации. И размер основных типов данных, и арифметические операции над ними точно определены.*

Например, тип int в языке Java всегда означает 32-разрядное целое число. В языках С и C++ тип int может означать как 16-разрядное, так и 32-разрядное целое число, а также целое число произвольного размера, по выбору разработчика конкретного компилятора.

Short int<=int<=long int!!!!

 Фиксированный размер числовых типов позволяет избежать многих неприятностей, связанных с выполнением программ на разных компьютерах.

*Объектно-ориентированный.* **О**бъектно-ориентированное программирование - это метод программирования, в центре внимания которого находятся данные (т.е. объекты) и средства доступа к ним.

*Высокопроизводительный*

синхронными компиляторами (just-in-time compilers—JIT). Они транслируют байт-код в машинозависимый код, сохраняют результат в памяти, а затем вызывают его при необходимости  
*Многопоточный*

Обеспечивает лучшую интерактивность и выполнение программы.  
*Динамичный*

Во многих отношениях язык Java является более динамичным, чем языки С или C++. Он был разработан так, чтобы легко адаптироваться к постоянно изменяющейся среде. В библиотеки можно свободно добавлять новые методы и объекты, не причиняя никакого вреда. Язык Java позволяет легко получать информацию о ходе выполнения программы.  
*Не зависящий от архитектуры компьютера*

Компилятор генерирует объектный файл, формат которого не зависит от архитектуры компьютера, — скомпилированная программа может выполняться на любых процессорах под управлением системы выполнения программ языка Java.

**Язык Java и Интернет**

**Апплеты – java-программы, работающие под управлением веб-браузеров.**

программу Java Plug-in, позволяющую создавать наиболее современную среду для запуска программ на языке Java на основе броузеров Netscape и Internet Explorer. Загрузка апплета напоминает внедрение изображения в Web-страницу.

Начало работы с Java

* 1. <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> - качаем jdk
  2. Устанавливаем
  3. **Задание путей к выполняемым файлам: Меню->правой кнопкой мыши на «Этот компьютер»->Свойства->Дополнительные параметры системы->Дополнительно->Переменные среды->Системные переменные->Cоздаем новую системную переменную (JAVA\_HOME/путь к bin-file, обязательно название в кавычках если там содержаться пробелы”C:\Program Files\Java\jdk1.8.0\_45\bin”;**

Системные переменные ->ищем переменную PATH->в конец добавляем точку с запятой и прописываем тот же путь**”C:\Program Files\Java\jdk1.8.0\_45”**;

**Среда разработки программ -** Пакет Java SDK не имеет никаких средств, даже отдаленно напоминающих интегрированную среду разработки программ. Все команды выполняются путем их набора в окне оболочки.

Итак, у вас есть три возможных варианта выбора.

1. Использовать набор инструментальных средств Java SDK вместе со своим любимым текстовым редактором. Компиляция и запуск программы на выполнение в этом случае производится в режиме командной строки.

2. Использовать набор Java SDK и текстовый редактор, интегрированный вместе с этим пакетом. Эту и многие другие возможности предоставляют программы TextPad и Emacs. В этом случае компиляция и запуск на выполнение программ производятся внутри редактора.

3. Использовать интегрированную среду разработки программ, например, свободно распространяемую программу Sun ONE Studio Community Edition или  
одну из многих других как бесплатных, так и коммерческих сред (eclipse,maven…).

-Пока используем блокнот для написание простейшей программки «Hello Java»

Сразу оговорюсь о правилах синтаксиса Java:

[http://www.oracle.com/technetwork/java/javase/documentation](http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-135099.html)

Основы Java: имена переменных и методов

Начнем серию статей, посвященных основам Java, с правил и рекомендаций, которые предлагает нам Sun (или, вернее, уже Oracle), по именованию переменных и методов.

**Переменные**

И так. Вот перечень основных правил именования переменных:

* Имена переменных должны начинаться с буквы, знака $ или знака подчеркивания \_. Имена не могут начинаться с цифр!
* После первого символа в имени могут идти любые комбинации букв, цифр, знаков подчеркивания и $.
* Нет ограничения на количество символов, которое может содержать имя переменной.
* В качестве имен нельзя использовать зарезервированные в Java слова (перечень приведен ниже).
* Имена переменных в Java регистрозависимые - переменные с именами foo и FOO это две различные переменные.

Вот примеры *допустимых* имен:

int a;

int b1;

int \_c;

int $d;

int i\_am\_damn\_long\_variable\_identifier;

А вот примеры *недопустимых*:

int :e;

int /d;

int f#;

int .g;

int 14h;

Перечень зарезервированных слов в Java:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| abstract | boolean | break | byte | case | catch |
| char | class | const | continue | default | do |
| double | else | extends | final | finally | float |
| for | goto | if | implements | import | instanceof |
| int | interface | long | native | new | package |
| private | protected | public | return | short | static |
| strictfp | super | switch | synchronized | this | throw |
| throws | transient | try | void | volatile | while |
| assert | enum |  |  |  |  |

**Рекомендации по оформлению кода**

* *Классы и интерфейсы*. Первая буква в имени должна быть заглавной и если в имени содержится несколько слов, то каждую первую букву в последующих словах следует делать заглавной (проще говоря, используйте camelCase). Имена классов следует делать существительными. Например:

Cat

FootballTeam

Имена интерфейсам следует давать в форме прилагательных:

Runnable

Serializable

* *Методы*. Первую букву следует делать строчной и далее следовать рекомендациям camelCase. Имена следует давать в виде сочетания глаголов и существительных. Например:

getNumber

doCalculation

setCustomerName

* *Переменные.*Как и для методов, придерживайтесь camelCase и присваивайте говорящие имена, дабы было понятно для чего эта переменная без всяких комментариев:

labelText

myAwesomeString

* *Константы.*Константы в Java создаются с помощью зарезервированных слов static и final. Имена констант следует задавать только заглавными буквами, а слова в имени разделять знаком подчеркивания:

MY\_CONST

Напишем первую программу на языке java, для этого открываем блокнот и пишем

-public class Lesson\_1{

public static void main(String[] args){

System.out.println("Hello Java");

}

}

* Сохраним с расширением java!!!
* Win+R->cmd
* Прописываем путь к директории: консольные команды, которые понадобятся –это:

1) D: консоль понимает данную команду как переход на диск D

2) cd Class(Directory’s name)войти в директорию

3) dir - содержимое директории

- Определяем установлена ли Java: java –version

- если да, то заходим в директорию, смотрим ее содержимое, наш файл на месте;

- компилируем программу: javac ClassName.java

- запускаем программу: java ClassName

Скачиваем интегрированную среду разработки:

[https://www.jetbrains.com/idea/download/](https://vk.com/away.php?to=https%3A%2F%2Fwww.jetbrains.com%2Fidea%2Fdownload%2F)

<https://eclipse.org/home/index.php>