Java — это не только язык программирования, но и экосистема инструментов, охватывающая почти все, что может понадобиться при программировании на Java. В нее входят:

Java Development Kit (JDK) — комплект разработчика Java. С помощью JDK и стандартного блокнота можно писать и запускать/ компилировать код на Java;

Java Runtime Environment (JRE) — исполняющая система Java. Механизм распространения программного обеспечения, состоит из автономной виртуальной машины Java, стандартной библиотеки Java (Java Class Library) и инструментов настройки.

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).

Integrated Development Environment (IDE) — интегрированная среда разработки. Инструменты, которые помогают запускать, редактировать и компилировать код.

Ключевые слова JAVA.

Для запоминания это:

* Примитивы (byte, short, int, long, char, float, double, boolean)
* Циклы и ветвления (if, else, switch, case, default, while, do, break, continue, for)
* Исключения (try, catch, finally, throw, throws)
* Области видимости (private, protected, public)
* Объявление \ Импорт (import, package, class, interface, extends, implements, static, final, void, abstract, native)
* Создание \ Возврат \ Вызов (new, return, this, super)
* Многопоточность (synchronized, volatile)
* instanceof, enum, assert, transient, strictfp, const, gotos

Имя или идентификатор переменной — это последовательность из строчных и заглавных латинских букв, цифр, а также символов «$» и «\_». Имя переменной может начинаться с любого из перечисленных символов, кроме цифры.

Технически возможно начать имя переменной также с «$» или «\_», однако это запрещено соглашением по оформлению кода в Java (Java Code Conventions). Кроме того, символ доллара «$», по соглашению, никогда не используется вообще. В соответствии с соглашением имя переменной должно начинаться именно с маленькой буквы (с заглавной буквы начинаются имена классов). Пробелы при именовании переменных не допускаются.

**Инициализация** (от англ. *initialization*, инициирование) —С точки зрения Java – выделение памяти под объект.

#### типы данных

**Примитивные**

* byte (целые числа, 1 байт, [-128, 127])
* short (целые числа, 2 байта, [-32768, 32767])
* int (целые числа, 4 байта, [-2147483648, 2147483647])
* long (целые числа, 8 байт, [-9223372036854775808,9223372036854775807])
* float (вещественные числа, 4 байта)
* double (вещественные числа, 8 байт)
* char (символ Unicode, 2 байта, [0, 65536])
* boolean (значение истина/ложь, используется int, зависит от JVM)

**Ссылочные**. В ссылочные типы входят все классы, интерфейсы, массивы.

1. byte, short, char в выражениях всегда повышаются до int
2. если в выражении участвует тип long – то именно к этому типу будет приведён результат
3. если в выражении участвует float – то результат приводится к float
4. если один из операндов имеет тип double – то к этому типу будет приведён весь результат
5. При выборе между длиной и возможностью сохранить дробную часть – будет выбрана дробная часть

При различных операциях может происходить повышение типов в порядке “усиления” к более информативному типу. Например складывая int и double получим тип double. Но есть и особенность, например сложив double (8 байт) и long (8 байт) Java оставит знаки после запятой (double), а не более “длинный” тип.

**При повышении типа byte>short; short>int; int>long; float>double; char>int информация не потеряется. При сужении возможна потеря информации**

#### boxing/unboxing

Это автоматическое преобразование из примитивных типов данных к ссылочным и наоборот.

Работа с объектами неизбежно влечет за собой накладные расходы по памяти и быстродействию. Чтобы избежать этого, используются переменные примитивных типов. Это, по сути, простые переменные, как в C или С++. byte занимает 1 байт памяти, int и float - по 4 байта, long и double - по 8 байт и т.д.

В отличие от операций с объектами, операции с переменными примитивных типов не требуют выделения/освобождения памяти и выполняются быстро - они в в большинстве случаев в конечном счете компилируются в простые процессорные инструкции, что позволяет программам на Java часто работать со скоростью, сравнимой с программами, написанными на простых компилируемых (непосредственно в машинный код) языках (типа тех же C, С++).

Почитать:

*Статьи сильно выходят за объем, затронутый на занятии. Это и к лучшему :)) если видите, что какая-то часть пока нами не изучена, можете вернуться к статье чуть позже.*

<https://habr.com/ru/post/49582/>

<https://compress.ru/article.aspx?id=23541> (первые 2 пункта)

<https://www.programiz.com/java-programming/variables-primitive-data-types>

<https://ru.hexlet.io/blog/posts/yazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda>

<https://java.com/ru/about/>