

JAVA – Язык программирования

Java

- Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры. Дата официального выпуска — 23 мая 1995 года.

Основные особенности языка

- Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

- В языке Java только 8 примитивных (скалярных, простых) типов: boolean, byte, char, short, int, long, float, double. Существует также вспомогательный девятый примитивный тип — void, однако переменные и поля такого типа не могут быть объявлены в коде, а сам тип используется только для описания соответствующего ему класса, для использования при рефлексии. Кроме того, с помощью класса Void можно узнать, является ли определённый метод типа void: `Hello.class.getMethod("main", Array.newInstance(String.class, 0).getClass()).getReturnType() == Void.TYPE`.
- Длины и диапазоны значений примитивных типов определяются стандартом, а не реализацией, и приведены в таблице. Тип `char` сделали двухбайтовым для удобства локализации (один из идеологических принципов Java): когда складывался стандарт, уже существовал Unicode-16, но не Unicode-32. Поскольку в результате не осталось однобайтового типа, добавили новый тип `byte`, причём в Java, в отличие от других языков, он не является беззнаковым. Типы `float` и `double` могут иметь специальные значения, и «не число» (NaN). Для типа `double` они обозначаются `Double.POSITIVE_INFINITY`, `Double.NEGATIVE_INFINITY`, `Double.NaN`; для типа `float` — так же, но с приставкой `Float` вместо `Double`. Минимальные и максимальные значения, принимаемые типами `float` и `double`, тоже стандартизованы.

Тип	Длина (в байтах)	Диапазон или набор значений
boolean	1 в массивах, 4 в переменных [33]	true, false
byte	1	-128..127
char	2	$0..2^{16}-1$, или 0..65535
short	2	$-2^{15}..2^{15}-1$, или -32768..32767
int	4	$-2^{31}..2^{31}-1$, или -2147483648..2147483647
long	8	$-2^{63}..2^{63}-1$, или примерно $-9.2 \cdot 10^{18}..9.2 \cdot 10^{18}$
float	4	$-(2-2^{-23}) \cdot 2^{127}..(2-2^{-23}) \cdot 2^{127}$, или примерно $-3.4 \cdot 10^{38}..3.4 \cdot 10^{38}$, а также , , NaN
double	8	$-(2-2^{-52}) \cdot 2^{1023}..(2-2^{-52}) \cdot 2^{1023}$, или примерно $-1.8 \cdot 10^{308}..1.8 \cdot 10^{308}$, а также , , NaN

Преобразования при математических операциях

В языке Java действуют следующие правила:

- Если один операнд имеет тип `double`, другой тоже преобразуется к типу `double`.
- Иначе, если один операнд имеет тип `float`, другой тоже преобразуется к типу `float`.
- Иначе, если один операнд имеет тип `long`, другой тоже преобразуется к типу `long`.
- Иначе оба операнда преобразуются к типу `int`.

- В языке Java имеются только динамически создаваемые объекты. Причём переменные объектного типа и объекты в Java — совершенно разные сущности. Переменные объектного типа являются ссылками, то есть неявными указателями на динамически создаваемые объекты. Это подчёркивается синтаксисом описания переменных. Так, в Java нельзя писать:

- `double a[10][20]; Foo b(30);`

- а нужно:

- `double[][] a = new double[10][20]; Foo b = new Foo(30);`

Объектными являются переменные любого типа, кроме примитивного. Явных указателей в Java нет. В отличие от указателей C, C++ и других языков программирования, ссылки в Java в высокой степени безопасны благодаря жёстким ограничениям на их использование, в частности:

- Нельзя преобразовывать объект типа `int` или любого другого примитивного типа в указатель или ссылку и наоборот.
- Над ссылками запрещено выполнять операции `++`, `--`, `+`, `-` или любые другие арифметические операции.
- Преобразование типов между ссылками жёстко регламентировано. За исключением ссылок на массивы, разрешено преобразовывать ссылки только между наследуемым типом и его наследником, причём преобразование наследуемого типа в наследующий должно быть явно задано и во время выполнения производится проверка его осмысленности. Преобразования ссылок на массивы разрешены лишь тогда, когда разрешены преобразования их базовых типов, а также нет конфликтов размерности.
- В Java нет операций взятия адреса (`&`) или взятия объекта по адресу (`*`). Амперсанд (`&`) означает всего лишь «побитовое и» (двойной амперсанд — «логическое и»). Однако для булевых типов одиночный амперсанд означает «логическое и», отличающееся от двойного тем, что цепь проверок не прекращается при получении в выражении значения `false`^[35] (напр. `a == b && foo() == bar()` не повлечёт вызовов `foo()` и `bar()` в случае, если `a != b`, тогда как использование `&` повлечёт в любом случае).

Благодаря таким специально введенным ограничениям в Java невозможно прямое манипулирование памятью на уровне физических адресов (хотя определено значение ссылки, не указывающей ни на что: `null`).

Если нужен указатель на примитивный тип, используются классы-обёртки примитивных типов: `Boolean`, `Byte`, `Character`, `Short`, `Integer`, `Long`, `Float`, `Double`.