



Список базовых типов данных

Java Syntax (/quests/QUEST_JAVA_SYNTAX)
10 уровень (/quests/lectures/?quest=QUEST_JAVA_SYNTAX&level=10), 2 лекция (/quests/lectures/questsyntax.level10.lecture02)

ОТКРЫТА

— Привет, Амиго!

— Привет, Риша!

— Ты уже изучил основы синтаксиса Java, поэтому я хочу рассказать тебе некоторые вещи более углубленно.

— Сегодня я расскажу тебе немного про примитивные типы, а также — сколько памяти они занимают. Тебе это пригодится и, возможно, даже сегодня. Вот эти типы:

Тип
Размер, байт
Диапазон значений
Значение по умолчанию
Описание
byte
1
-128 .. 127
0
Самое маленькое целое — один байт
short
2
-32,768 .. 32,767
0
Короткое целое, два байта

int

4

— $2 \cdot 10^9$.. $2 \cdot 10^9$

0

Целое число, 4 байта

long

8

— $9 \cdot 10^{18}$.. $9 \cdot 10^{18}$

0L

Длинное целое, 8 байт

float

4

-10^{127} .. 10^{127}

0.0f

Дробное, 4 байта

double

8

-10^{1023} .. 10^{1023}

0.0d

Дробное, двойной длины, 8 байт

boolean

1

true, false

false

Логический тип (только true & false)

char

2

0..65,535

'\u0000'

Символы, 2 байта, все больше 0

Object

4

Любая ссылка или null.

null

Хранит ссылки на объекты типа Object или классов наследников

— Давай я остановлюсь на каждом типе подробнее.

— Тип byte – это самый маленький тип целых чисел. Каждая переменная этого типа занимает всего один байт памяти, поэтому он может хранить значения в диапазоне от -128 до 127.

— А зачем такой маленький тип нужен. Почему нельзя везде использовать int?

— Можно. Но если ты создаешь очень большие массивы, и тебе не нужно хранить там значения больше 100, то почему бы не воспользоваться таким типом. Так ведь?

— Тип short – ровно в два раза длиннее типа byte и тоже хранит только целые числа. Самое большое число, которое в него вмещается – это 32767. Самое большое отрицательное число – это -32768.

— Тип int ты уже знаешь. Он может хранить целые числа до двух миллиардов, как положительные, так и отрицательные.

— Тип float – создан для хранения вещественных (дробных) чисел. Его размер 4 байта.

— Все дробные числа хранятся в памяти в очень интересной форме.

— Например, число 987654.321 можно представить как $0.987654321 \cdot 10^6$. Поэтому в памяти оно будет представлено как два числа «0.987654321» (мантиса — значащая часть числа) и «6» (экспонента — степень десятки)

— А зачем так сделано?

— Такой подход позволяет хранить гораздо большие числа, чем int, используя всего 4 байта. Но при этом мы жертвуем точностью. Часть памяти расходуется на хранение мантиссы, поэтому такие числа хранят всего 6-7 знаков после запятой, остальные отбрасываются.

-Такие числа еще называют «числа с плавающей запятой» или «числа с плавающей точкой (floating point number)». Отсюда, кстати, и название типа – float.

— Ясно.

— Тип double – это такой же тип, как и float, только в два раза длиннее – он занимает восемь байт. (double – двойной, по-английски). И предельный размер мантиссы и количество значащих цифр в нем больше. Если тебе нужно хранить вещественные числа – старайся использовать именно этот тип.

— Тип char – гибридный тип. Его значения можно интерпретировать и как числа (их можно складывать и умножать) и как символы. Так было сделано потому, что хоть символы и имеют визуальное представление, для компьютера они в первую очередь просто числа. И работать с ними как с числами гораздо удобнее. Тут еще есть одно замечание: тип char строго положительный – отрицательных значений он хранить не может.

— Тип boolean – логический тип и может хранить всего два значения: true (истина) и false (ложь).

— Тип Object, хоть и представлен в таблице, примитивным типом не является. Это базовый класс для всех классов в Java. Во-первых, все классы считаются унаследованными от него, а значит, содержат его методы. А во-вторых, ему можно присваивать ссылки на объекты любого типа. В том числе и null – пустую ссылку.

— Узнал очень много нового. Спасибо за урок, Риша

[< \(/quests/lectures/questsyntax.level10.lecture01\)](/quests/lectures/questsyntax.level10.lecture01)[×10 > \(/quests/lectures/questsyntax.level10.lecture03\)](/quests/lectures/questsyntax.level10.lecture03)



0



0



0



0



0

+25

Комментарии (23)

популярные

новые

старые

Никита

GovnoCoder 10 уровень, Минск

четверг, 20:00

С размером `boolean` вроде не так все просто, в литературе часто вообще не указывают его размерность, а где-то указывают размер от 1 бита до 4 байт. Я так понял все зависит от JVM.

Ответить

0

Джонни 22 уровень

8 октября 2017, 14:43

"Тип"	Размер (байт)	Диапазон значений
<code>char</code>	2	0..65,535"

Введите этот код:

```
1 System.out.print((char) 111111111);
```

Ребята, нам явно что-то недоговаривают.

Ответить

+4

Artem 34 уровень, Санкт-Петербург

19 декабря 2017, 17:53

А еще, говорят,
"Рассмотрим распространенную Sun JVM и прочтем в спецификации виртуальной машины, что `boolean` типа в ней нет как такового, вместо него используется `int`" - с Хабра спер :)

так что да, еще и про Булеан недоговаривают.

Ответить

0

Джонни 22 уровень

27 декабря 2017, 21:32

А вот и ответ на мой вопрос:
<https://youtu.be/Sli18YQjfBM?t=1m28s>
Тут, правда, про Юникод, но суть понятна.

Ответить

+1

Beslan Bavizhev 31 уровень

20 января, 19:30

```
1 char c = (char) 111111111;  
2 int n = c;  
3 System.out.println(c);  
4 System.out.println(n);  
5 System.out.println(111111111 % 65536);
```

Я думаю понятно без комментариев.

Ответить

+1

Джонни 22 уровень

22 января, 03:41

Да, да, оказалось обычным приведением типов :)

Ответить

0

Spark 11 уровень, Минск

25 июля 2017, 16:58

Самое большое отрицательное число – это -32768.
Учитывая, что в этом случае рассматриваем целое отрицательное число, то наибольшим отрицательным будет -1....

Ответить

+24

Sergio 19 уровень

6 сентября 2017, 10:54

-1 - самое большое отрицательное ЦЕЛОЕ число, если быть точным

Ответить

+4

ывыфв 13 уровень

7 сентября 2017, 17:28

чё вы дотошные такие?

Ответить

+11

Shohrux Yusupov 12 уровень

12 сентября 2017, 15:49

...

MaKaRi4 согласен. Нужно концентрироваться на нужном. А не искать тему для троллига. Всем известно что -1 больше -2 и тд

Ответить

+1

Дмитрий 31 уровень

10 октября 2017, 06:51

...

Не знаю почему, но так поржал с этого комментария)))

Ответить

+1

Spark 11 уровень, Минск

22 октября 2017, 17:55

...

Пояснишь, что смешного ?

Ответить

0

Vladislav Kovyazin 31 уровень

6 июля 2017, 10:10

...

Про long забыли рассказать)

Ответить

+3

Logikarma 13 уровень

10 ноября 2017, 18:25

...

а зачем? в таблице основная инфа : длинное целое, 8 байт

Ответить

0

Евгений Орлянкин 13 уровень, Харьков

19 июня 2017, 16:48

...

А откуда информация что boolean 1 байт и Object 4, ведь нигде в спецификации этого не сказано? Или есть скрытые источники?

Ответить

0

Daniel Bakel 17 уровень, Харьков

29 августа 2017, 09:26

...

boolean 1 bit

Ответить

0

Dimenterious Fox 11 уровень

29 августа 2017, 16:13

...

Вот тут можно почитать подробнее про размер boolean:
<https://stackoverflow.com/questions/1907318/why-is-javas-boolean-primitive-size-not-defined>.
 Строго говоря, он зависит от реализации виртуальной машины.

Ответить

+1

Николай Кравцов 10 уровень, Гомель

18 октября 2017, 23:40

...

Boolean по объему занимаемой памяти соответствует типу int, если смотреть спецификацию для JVM Sun, а кто как реализует boolean в своих JVM, это вопрос разработчиков JVM... в спецификации Oracle ничего нет..
 Если интересно - почитайте [boolean — боремся за Java память...](#)

Ответить

0

Антон Бельский 17 уровень, Минск

13 марта 2017, 12:58

...

Хмм... Кажется Риша забыла как пользоваться бритвой.

Ответить

+16

Виктор Ясельник 13 уровень

27 апреля 2017, 16:55

...

ахаха

Ответить

0

Vasilii 12 уровень

1 марта 2017, 15:01

...

Хоть бы вскользь упомянули, что для денежных расчетов float и double не подходят

Ответить

+5

Alexey Smirnov 29 уровень, Москва

8 апреля 2017, 13:42

...

И вот почему: из-за погрешностей округления по ходу вычислений накапливается ошибка.

Что делать? Использовать тип данных BigDecimal

Ответить

+5

Николай Кравцов 10 уровень, Гомель

18 октября 2017, 23:41

...

Или писать свои классы, где работать с целыми числами

...

[javarush.ru/](#) [G+ \(https://plus.google.com/114772402300089087607\)](#) [Twitter \(https://twitter.com/javarush_ru\)](#) [Facebook \(https://www.facebook.com/javarush.ru\)](#)



Программистами не рождаются
© 2018