**Список базовых типов данных**

[Java Syntax](https://javarush.ru/quests/QUEST_JAVA_SYNTAX)

[Уровень 10](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=10), Лекция 2

— 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\*109 .. 2\*109 | 0 | Целое число, 4 байта |
| long | 8 | — 9\*1018 .. 9\*1018 | 0L | Длинное целое, 8 байт |
| float | 4 | -10127 .. 10127 | 0.0f | Дробное, 4 байта |
| double | 8 | -101023 .. 101023 | 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\*106. Поэтому в памяти оно будет представлено как два числа «0.987654321» (**мантисса — 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 (**float**ing 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