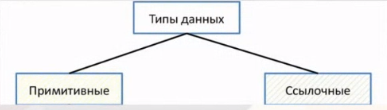
# Разбор основного синтаксиса :типы данных, переменные и константы, ввод и вывод данных

*Типы данных в Java: какие бывают и чем отличаются*

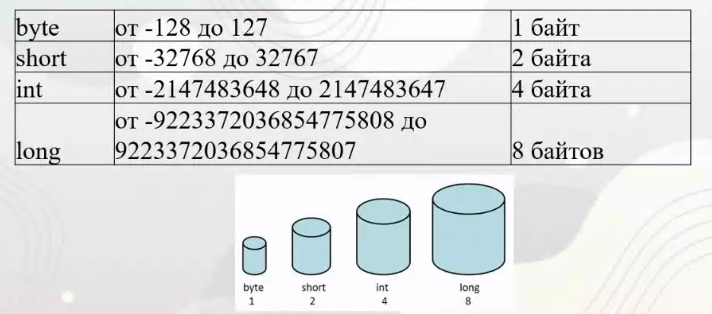
B Java типы данных делят на две большие группы: примитивные и ссылочные.

- Примитивные типы представляют простые значения, такие как целые числа, вещественные числа, логические значения и символы.

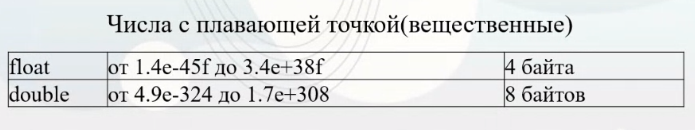
- Ссылочные типы представляют объекты, такие как классы, массивы, интерфейсы, строки и т.д.



Примитивные типы данных. Целые числа



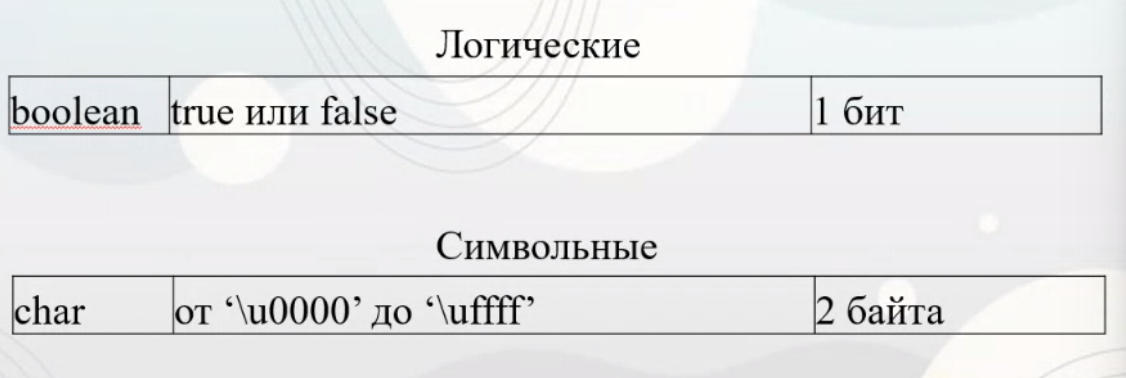
Числа с плавающей точкой.



Любое вещественное число может быть записано в памяти компьютера приближенно.

По умолчанию, числа с плавающей запятой в Java интерпретируются как double, поэтому если вы хотите присвоить такое число переменной типа float, вам нужно явно указать это с помощью F.

Логические и символьные



*Отличия ссылочных и примитивных типов данных*

1) Примитивные типы заранее определены языком и именуются ключевыми словами, а ссылочные типы создаются с помощью конструкторов классов

2)Примитивные типы занимают фиксированный объем памяти, в то время как ссылочные типы зависят от размера объекта, на который они ссылаются

3)Примитивные типы не имеют методов, а ссылочные типы имеют свойства и методы, которые определяются классом или интерфейсом

4)Примитивные типы не могут принимать значение null, а ссылочные типы могут принимать значение null, если на них не ссылается ни один обьект.

*Переменные в Java: как объявлять, инициализировать, присваивать и изменять значения переменных разных типов*

Объявление переменной — это операция, которая создает переменную и указывает ее тип данных.

Синтаксис:

Тип\_данных имя\_переменной;

Инициализация переменной - это операция, которая присваивает переменной начальное значение.

Синтаксис:

Имя\_переменной = значение;

Продолжение лекции 26.01

Присваивание переменной — это операция, которая изменяет значение переменной на новое.

Синтаксис:

Имя\_переменной = новое\_значение;

**Изменение значения переменной** — это операция, которая модифицирует значение переменной C помощью арифметических, логических или битовых операторов.

Синтаксис:

Имя\_переменной = имя\_переменной оператор выражение;

*Debug режим*

Debug режим — это специальный режим работы программы, в котором можно пошагово выполнять код, просматривать значения переменных, устанавливать точки останова и исправлять ошибки. Debug режим также называется отладкой или отладочным режимом. Для использования Debug режима нужно иметь специальный инструмент, называемый отладчиком.

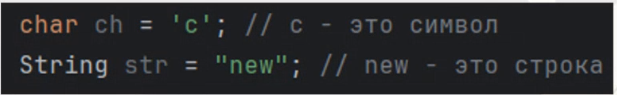
Отладчик — это программа, которая позволяет контролировать и анализировать выполнение другой программы.

*Точка останова*

Точка останова — это место в программе, где выполнение приостанавливается для вызова отладчика. Отладчик позволяет анализировать состояние программы и исправлять ошибки. Точка останова называется так, потому что она останавливает (прерывает) ход программы. Например, вы можете установить точку останова на определенной строке кода, и когда программа дойдет до этой строки, она остановится и перейдет к отладчику. Таким образом, вы можете проверить, правильно ли работает ваш код до этого момента.

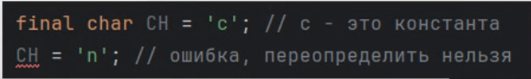
*Кавычки*

Кавычки в Java, в отличии от Python, имеют разное значение в зависимости от того, одинарные они или двойные. Одинарные кавычки используются для обозначения символов, а двойные кавычки - для обозначения строк.



*Константы в Java: как объявлять, инициализировать и использовать константы с помощью ключевого слова final*

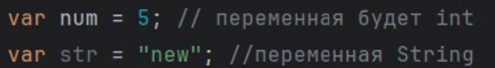
Константы в Java — это переменные, которые имеют фиксированное значение и не могут быть изменены в ходе выполнения программы. Константы используются для хранения важных данных, таких как числа, строки, символы и т.д. Константы могут быть объявлены в классе, интерфейсе или внутри метода. Для объявления константы в Java необходимо использовать ключевое слово **final**, за которым следует тип данных и имя константы.



Константы в программе принято писать большими буквами, чтобы отличать их от обычных переменных. Если имя константы состоит из нескольких слов, то они разделяются символом подчеркивания\_

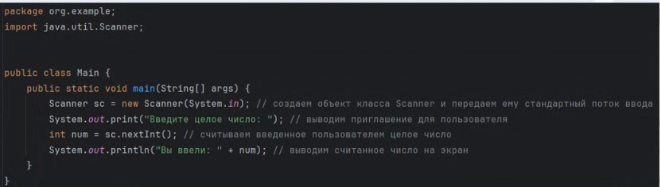


Ключевое слово var позволяет объявлять переменные без указания их типа. Тип переменной будет определяться компилятором на основе того значения, которое вы присвоите переменной при ее инициализации.

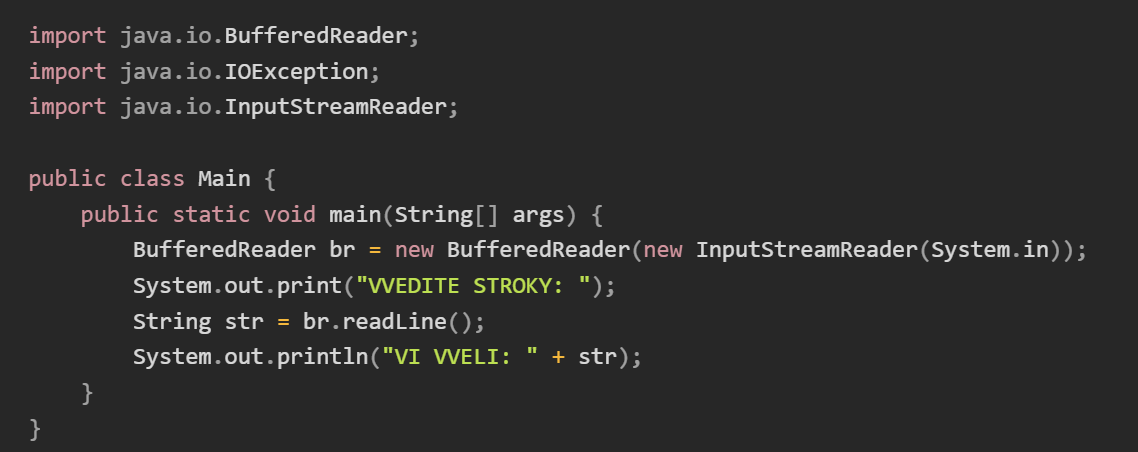


*Как считывать данные с клавиатуры, файла или сети с помощью классов Scanner, BufferedReader*

**Scanner** Этот класс позволяет считывать данные с любого потока ввода, такого как System.in (стандартный поток ввода, связанный с клавиатурой), FileInputStream (поток ввода, связанный с файлом) или Socket.getInputStream (поток ввода, связанный с сетевым соединением). Scanner имеет разные методы для считывания разных типов данных, таких как nextInt(), nextDouble(), nextLine() и т.д.

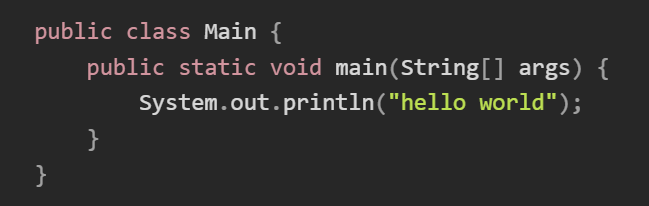


BufferedReader Этот класс позволяет считывать данные из любого потока ввода построчно, используя метод readLine(). Для того, чтобы использовать BufferedReader, нужно обернуть поток ввода в объект класса InputStreamReader, который преобразует байты в символы.



*Вывод данных в Јаvа: как выводить данные на экран и в файл с помощью классов System и PrintWriter*

Вывод данных на экран можно осуществлять с помощью класса System, который имеет статические поля out и err, представляющие стандартный поток вывода и поток ошибок. Для вывода данных на экран можно использовать методы println, print или printf



*Вывод данных в Java: как выводить данные на экран и в файл с помощью классов System и PrintWriter*

Вывод данных в файл можно осуществлять с помощью класса PrintWriter, который позволяет записывать текстовые данные в файл

