

# Школа IT

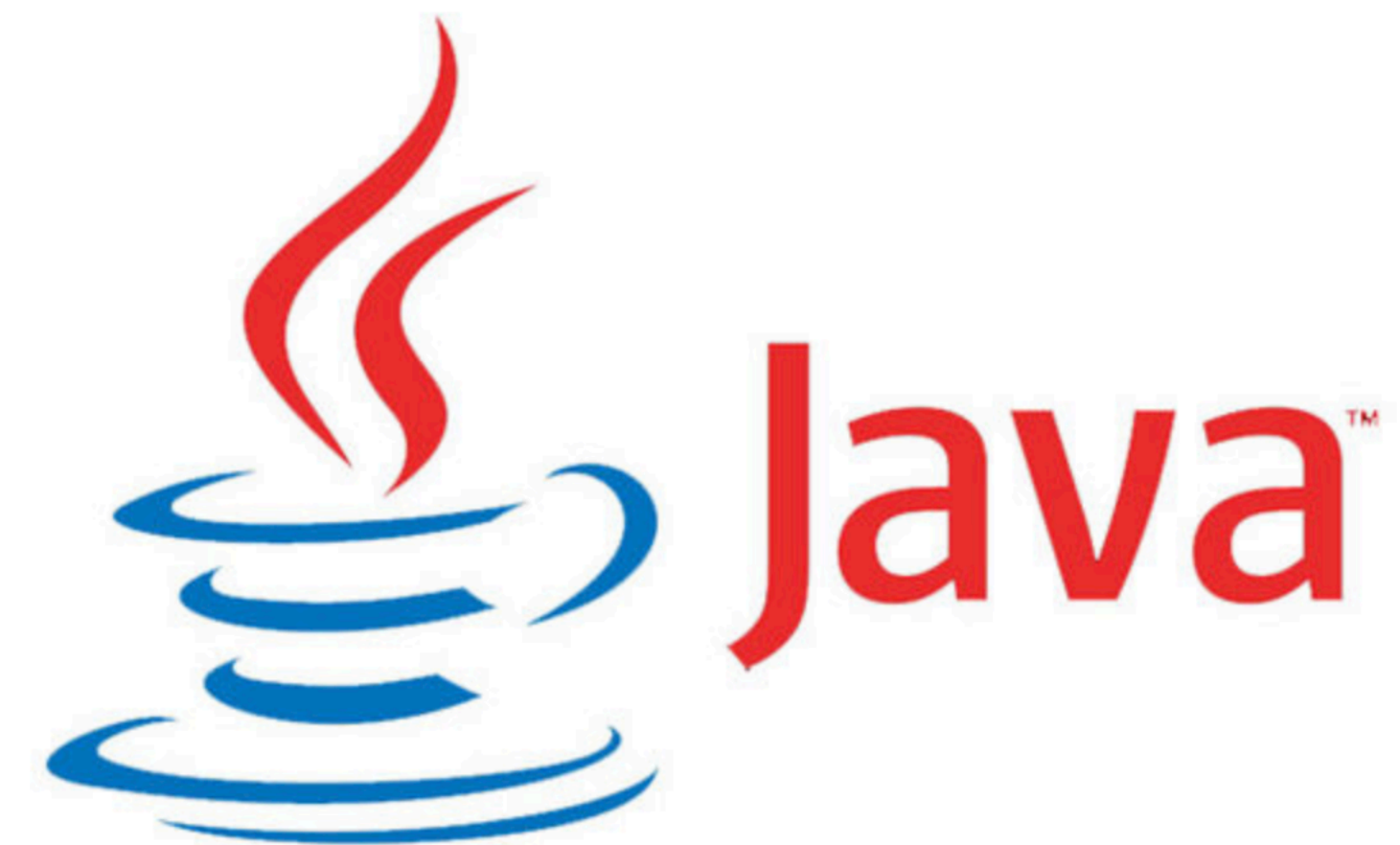
# Корпоративный университет

Java

Типы данных и управляющие конструкции

Занятие 3  
27.03.21

Александр Буянов



# Типы данных

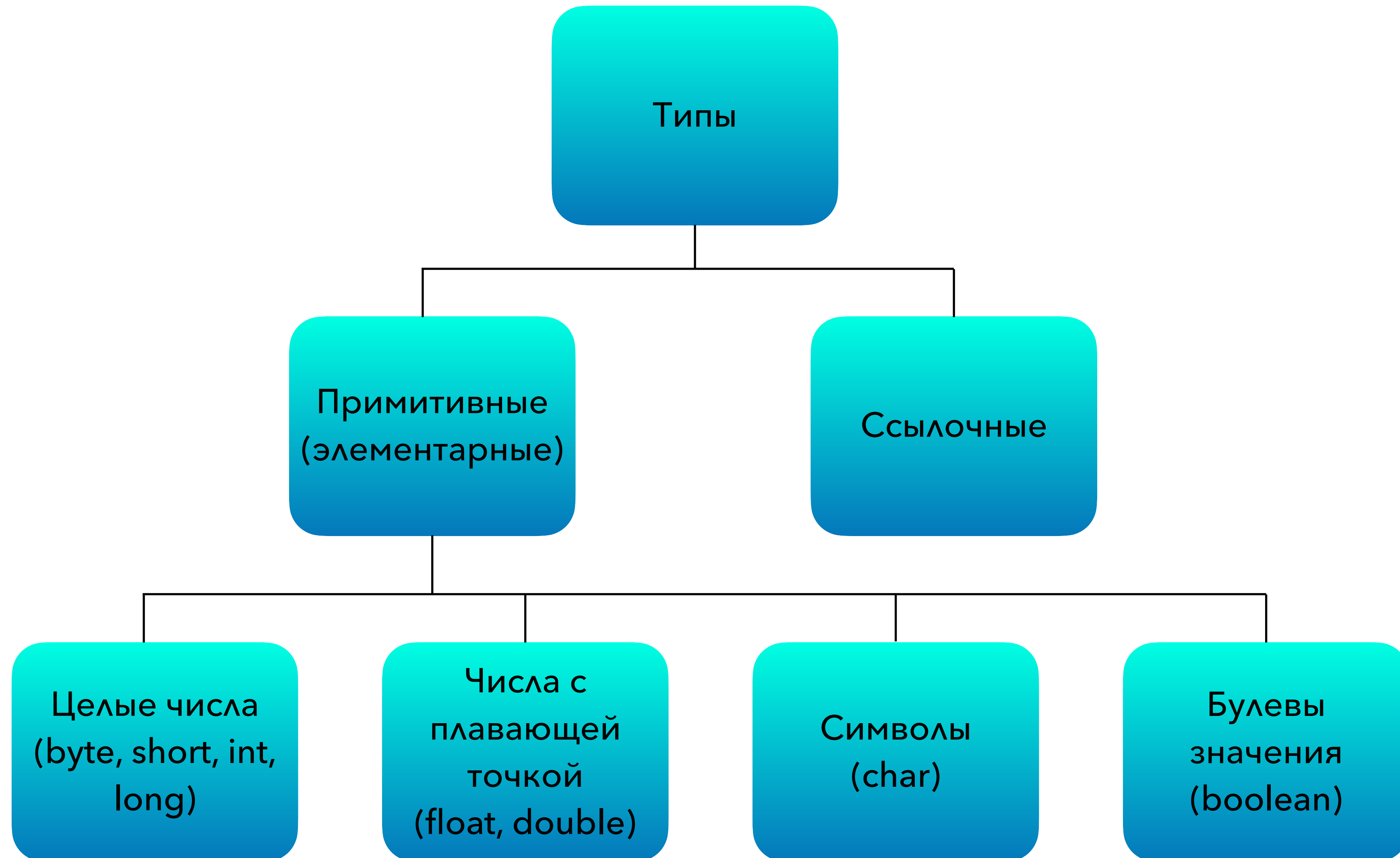
**Тип данных** - множество значений и операций над этими значениями.

Каждая переменная обладает типом, каждое выражение имеет тип и каждый тип строго определен.

Все присваивания переменных проверяются на строгое соответствие типов.

Ключевое слово var

# Типы данных



# Целочисленные типы Java

Имя	Ширина в битах	Диапазон допустимых значений
byte	8	от - 128 до 127 (от $-2^7$ до $2^7 - 1$ )
short	16	от - 32768 до 32767 (от $-2^{15}$ до $2^{15} - 1$ )
int	32	от -2147483648 до 2147483647 (от $-2^{31}$ до $2^{31} - 1$ )
long	64	от -9223372036854775808 до 9223372036854775807 (от $-2^{63}$ до $2^{63} - 1$ )

# Классы-обертки примитивных типов

Primitive Data Types	Wrapper Classes
int	Integer
short	Short
long	Long
byte	Byte
float	Float
double	Double
char	Character
boolean	Boolean

[Подробнее об обертках](#)

# Числа с плавающей точкой

Тип	Размер (бит)	Диапазон
<b>float</b>	32	от $1.4e-045$ до $3.4e+038$
<b>double</b>	64	от $4.9e-324$ до $1.8e+308$

[Подробнее о вычислениях в числах с плавающей точкой](#)



# Литералы

Обычная запись: -1.234

Экспоненциальная запись: -123.4e-2 ( $-123.4 \cdot 10^{-2}$ )

Шестнадцатеричная запись: 0xFFFFpFF ( $FFFF \cdot 2^{FF}$ )

С суффиксом типа:

38f

3e19d

123.4e-2f

444.444d

# Логический тип

Тип	Размер (бит)	Диапазон
<b>boolean</b>	32	<b>false</b> (ложь) или <b>true</b> (истина)



# Работа с логическими типами

Оператор	Описание
&	Логическое AND (И)
&&	Сокращённое AND
	Логическое OR (ИЛИ)
	Сокращённое OR
^	Логическое XOR (исключающее OR (ИЛИ))
!	Логическое унарное NOT (НЕ)
&=	AND с присваиванием
=	OR с присваиванием
^=	XOR с присваиванием
==	Равно
!=	Не равно
?:	Тернарный (троичный) условный оператор

# Символьный тип

Тип	Описание	Запись
char	символ UTF-16. Для записи используем одинарные кавычки	'А'; " – <b>ТАК НЕЛЬЗЯ</b> 'Всем привет!' – <b>ТАК НЕЛЬЗЯ</b>
String	Строка. Для записи используем двойные кавычки	"Всем привет!"; ""; "А"

# СИМВОЛЬНЫЙ ТИП

Тип	Размер (бит)	Диапазон
<b>char</b>	16	от 0 до 65536 (от 0 до $2^{16}$ )

Литералы:

- Символ в одинарных кавычках: 'a'
- Шестнадцатеричный код смвола: '\u78bc'
- Спецпоследовательности: '\n', '\t', '\'', '\\'

Таблица символов Юникод

# Арифметические операции

## Операция

+

-

\*

/

%

## Описание

Сложение (а также унарный плюс)

Вычитание (а также унарный минус)

Умножение

Деление

Деление по модулю

# Арифметические операции

- Результат деления целых чисел - целое число
- Результат деления на ноль - исключительная ситуация, получаем исключение `ArithmeticException`
- Переполнение числа не является исключительной ситуацией, но происходит ошибка вычисления



# Все примитивные типы данных

Ключевое слово	Тип	Диапазон допустимых значений (включительно)	Пример
byte	8-битное целочисленное значение	от -128 до 127	123
short	16-битное целочисленное значение	от -32768 до 32767	12345
int	32-битное целочисленное значение	от -2147483648 до 2147483647	1234567890
long	64-битное целочисленное значение	от -9223372036854775808 до 9223372036854775807	1234567890
float	32-битное значение с плавающей точкой	приблизительно $\pm 3.40282347E+38F$ (6-7 значащих десятичных цифр)	123.45f
double	64-битное значение с плавающей точкой	приблизительно $\pm 1.7976931348623157E+308F$ (15 значащих десятичных цифр)	123.456
char	16-битное значение Unicode	от 0 до 65535	'a'
boolean	истина или ложь	true или false	true

# Управляющие конструкции

- if
- switch
- while
- do while
- for
- break и continue



# ДЗ

## Задача 1

Сгенерировано случайное целочисленное число (любой диапазон).  
Необходимо написать программу, считывающую вводимое пользователем значение до тех пор, пока значение передаваемое в программу не совпадет со случайным значением. Если введенное число больше или меньше случайного, программа дает подсказку (больше/меньше).

**Комментарий:** для генерации случайного числа можно использовать класс Random;

## Задача 2

Написать программу, выполняющую рисование рамки вокруг текстовой строки. Программа должна принимать на вход размеры рамки (длина/ширина) и саму строку.

Например, при таком запуске программы:

```
6
20
Abracadabra
```

Тогда программа должна вывести

```
1 *****
2 *          *
3 *  Abracadabra  *
4 *              *
5 *              *
6 *****
```

Текстовая строка должна быть отцентрирована как по горизонтали, так и по вертикали. В случае, если длина строки не позволяет вписать строку в рамку заданного размера, программа должна вместо рамки выводить сообщение.

```
Ошибка
```

Для вычисления длины строки используйте:

```
String a = "Abracadabra";
a.length();
```