



Faculty of Software Engineering and Computer Systems

Programming

**Lecture #0. Introduction
and more.**

Instructor of faculty
Pismak Alexey Evgenievich
Kronverksky Pr. 49, 374 room
pismak@itmo.ru

Saint-Petersburg

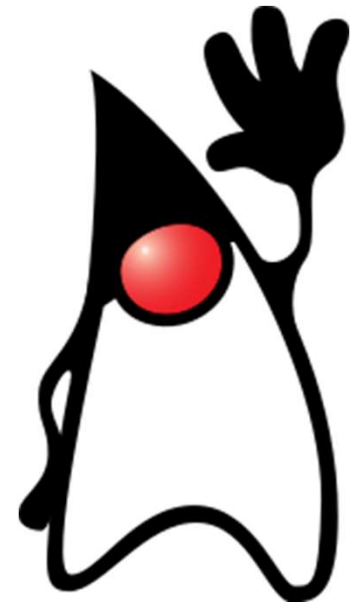
Рекомендуемый материал

- Гаврилов А.В., Клименков С.В., Цопа Е.А.
“Программирование на Java. Конспект лекций”
- Кей Хорстманн, Гари Корнелл “Java. Библиотека профессионала. Том 1. Основы”
- Кей Хорстманн “Java SE 8. Вводный курс” (“Java 8 для нетерпеливых”)
- Слайды лекций с комментариями “Programming. Learning Java” (<https://se.ifmo.ru/>)

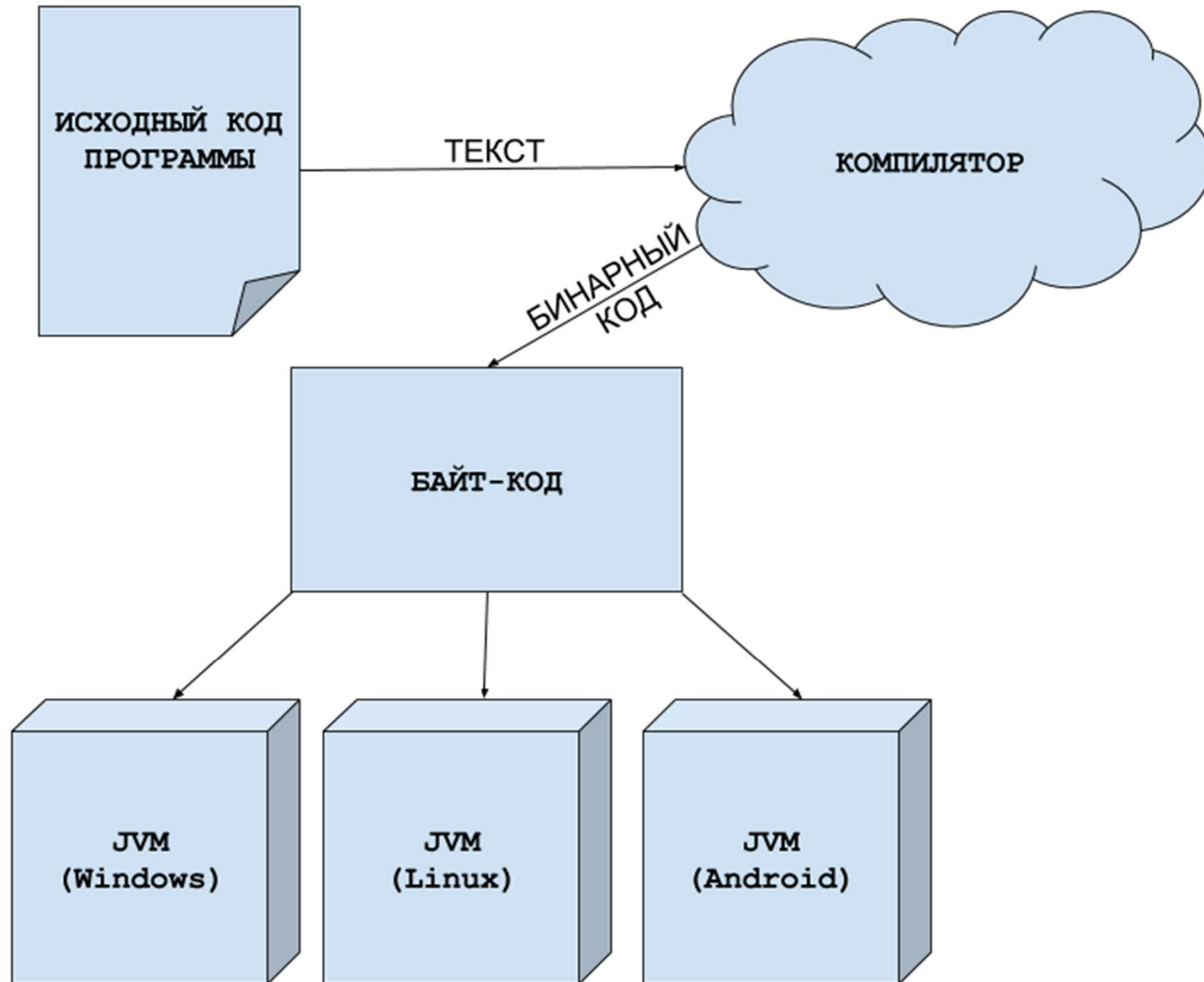
Инструменты для выполнения лабораторных работ

- Текстовые редакторы (SublimeText, Notepad++)
- JDK
- IDE (intelliJ IDEA, Eclipse, NetBeans)
- CMD (Putty, WinSCP, shell, helios)

Почему Java?



Жизненный цикл Java-программы

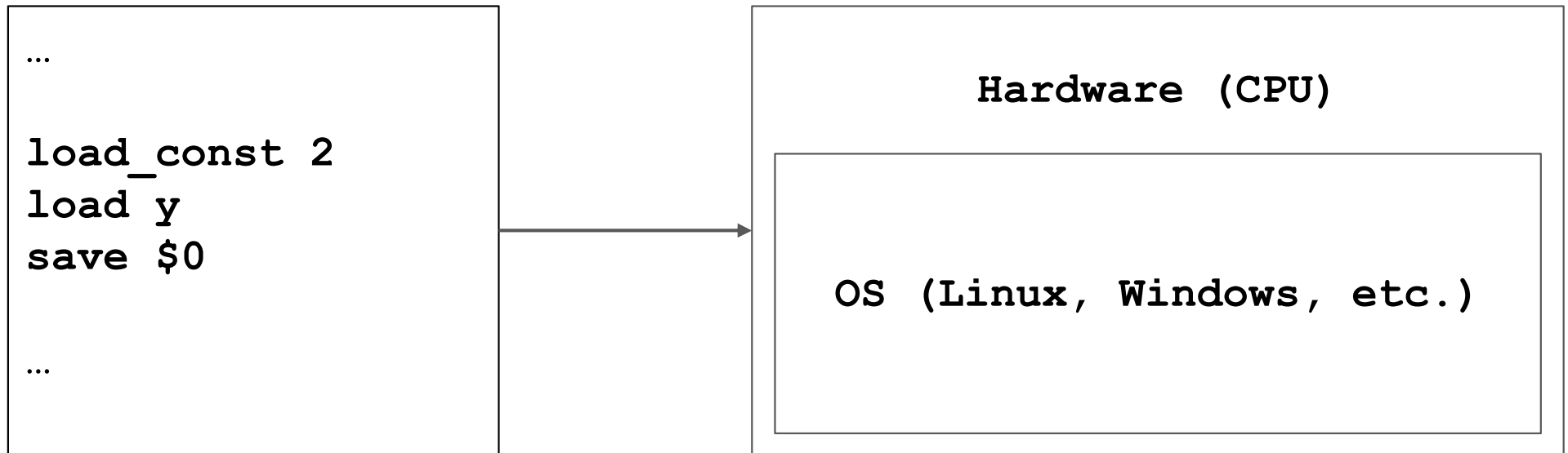


Компилятор

```
public class Alp
  public static
    x = y * 2;
  }
}
```

```
...
load_const 2
load y
save $0
...
```

JVM (Java Virtual Machine)



НИЧОСИ

*** КРОССПЛАТФОРМЕННОСТЬ**



Комментарии в программах Java

Однострочные комментарии

// комментарии

Многострочные комментарии

/
комментарии
/


Специальные комментарии (javadoc)

*/**
*
* @author James Gosling
/

Первые строки на языке Java

Main.java

```
1. public class Main {  
2.     public static void main(String[] args) {  
3.         System.out.println("Внимание!");  
4.     }  
5. }
```



Простая арифметическая программа

```
1. public class SimpleProgram {  
2.     public static void main(String[] args) {  
3.  
4.         double x = 5.0D;  
5.  
6.         double result = Math.pow(x, 2.5) ;  
7.  
8.         System.out.println(result) ;  
9.     }  
10. }
```

Что дальше делать с исходниками?

→ Что есть JDK?

- компилятор
- отладчик
- byte-code инструментарий
- виртуальная машина java
- стандартная библиотека java

→ Что есть JRE?

- виртуальная машина java
- стандартная библиотека java

Что дальше делать с исходниками?

1. Компиляция

```
javac    Main.java
```

```
javac    -d    target    Main.java
```

```
javac    --help
```

2. Упаковка в jar

```
jar      -cfm    app.jar    MANIFEST.mf    *.class
```

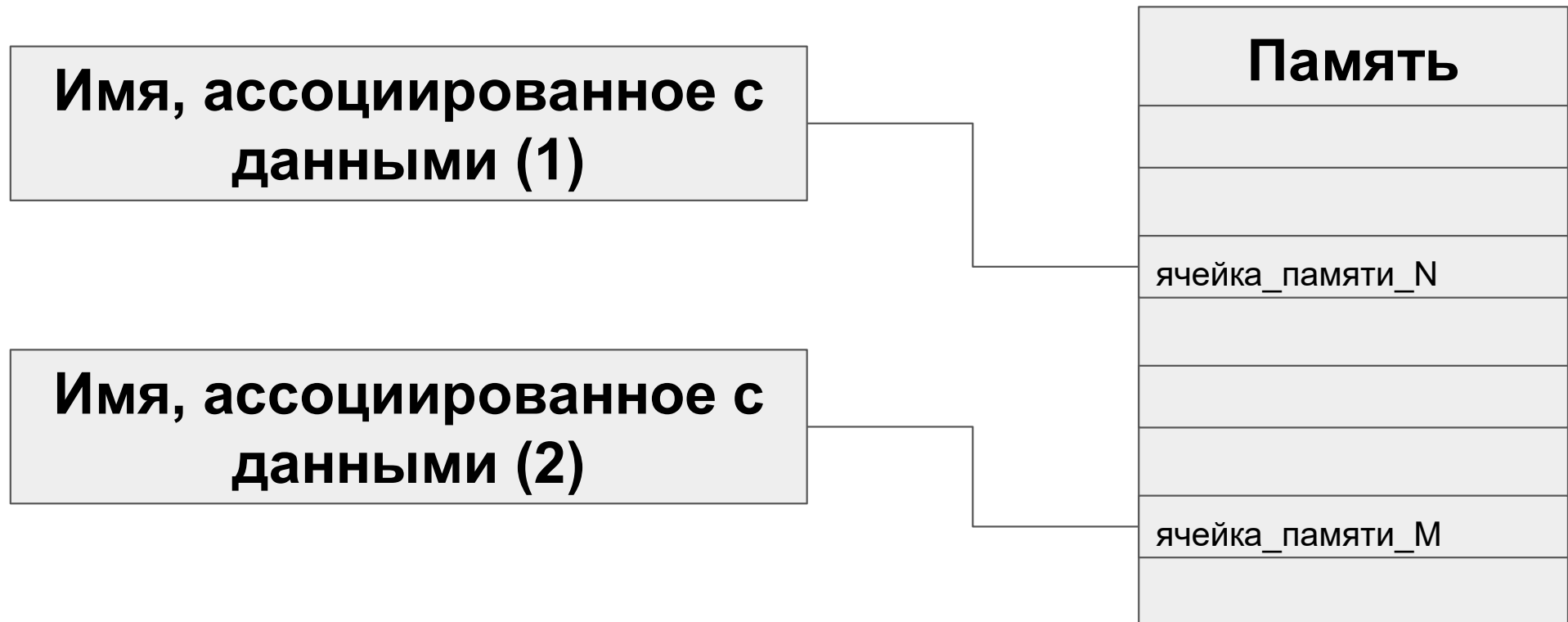
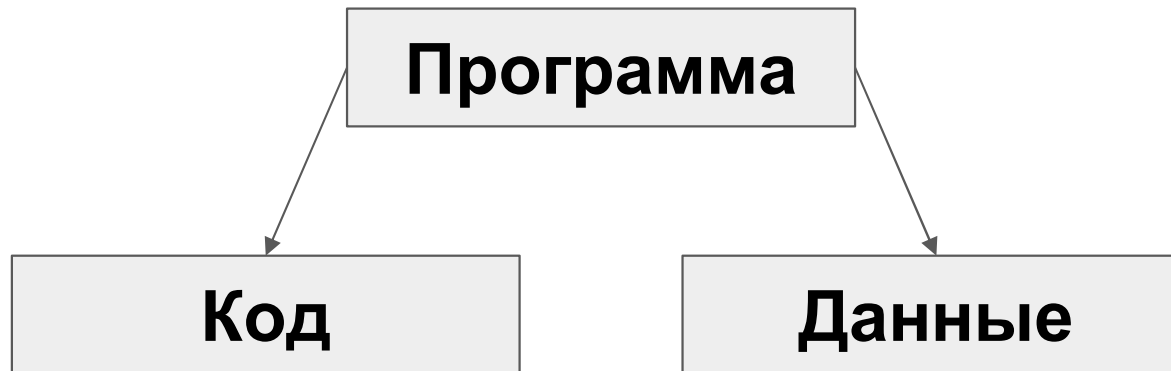
3. Запуск

```
java     -jar    app.jar
```

```
java     Main
```

4. Отладка (debugging)

Представление данных

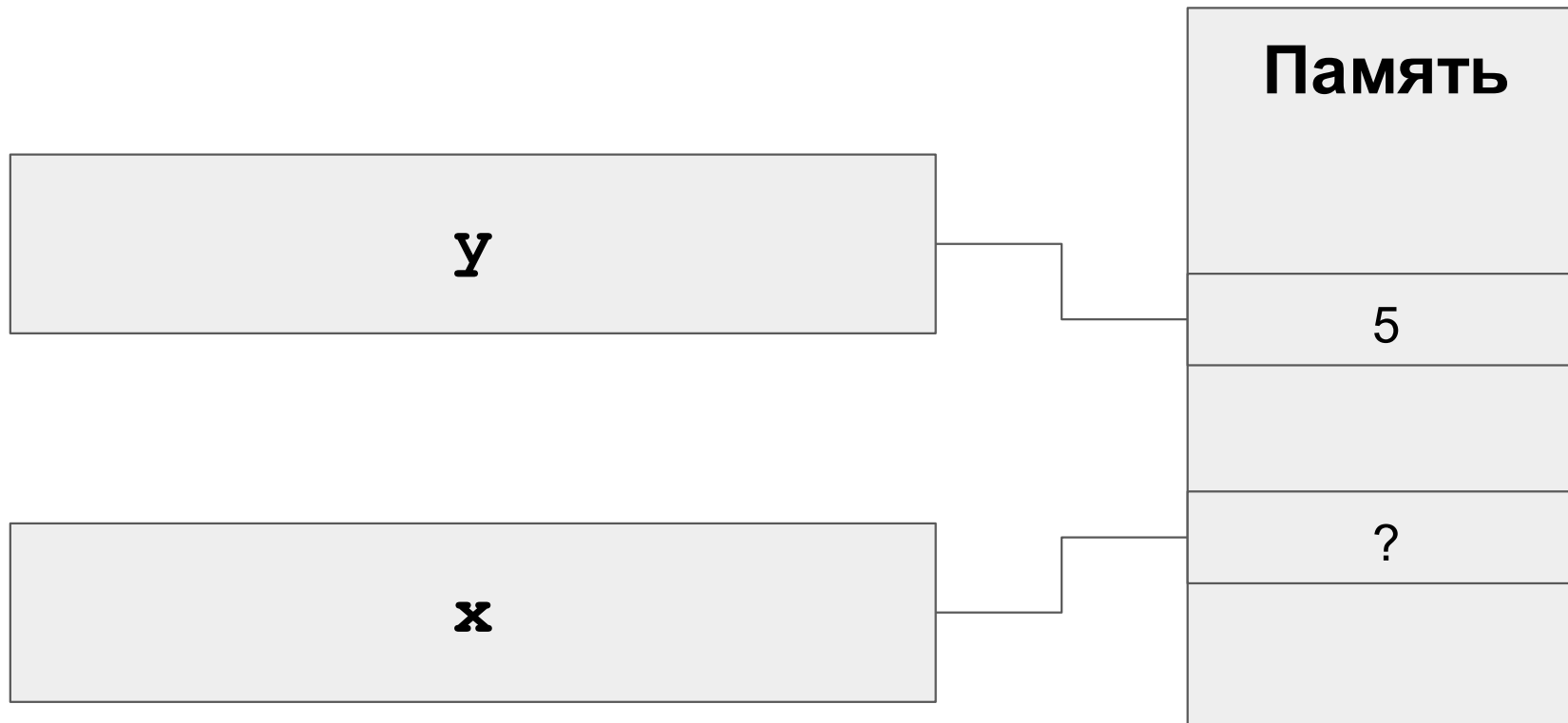


Переменные

```
int x;
```

```
int y = 5;
```

```
тип_данных имя_переменной [ = значение_по_умолчанию ] ;
```



Константы

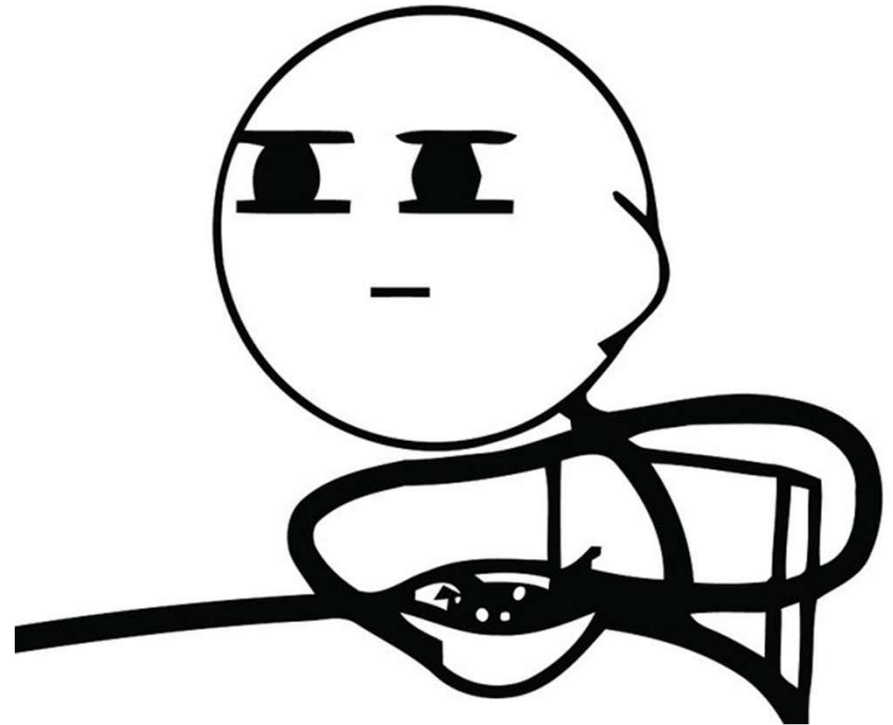
Объявление констант

```
final double E = 5.0;
```

```
final double R;  
// ...  
R = 5.0;
```

Назначение констант

```
int total = 8 * 5 * 4 * 12;
```



Константы

Назначение констант

```
int total = 8 * 5 * 4 * 12; // ПЛОХО
```

```
final int HOURS = 8, DAYS = 5, WEEKS = 4, MONTHS = 12;  
// ...  
int total = HOURS * DAYS * WEEKS * MONTHS; // ХОРОШО
```



Типы данных. Классификация.

Целочисленные

Вещественные

Логические

Объекты

`byte`

`double`

`boolean`

`short`

`float`

`char`

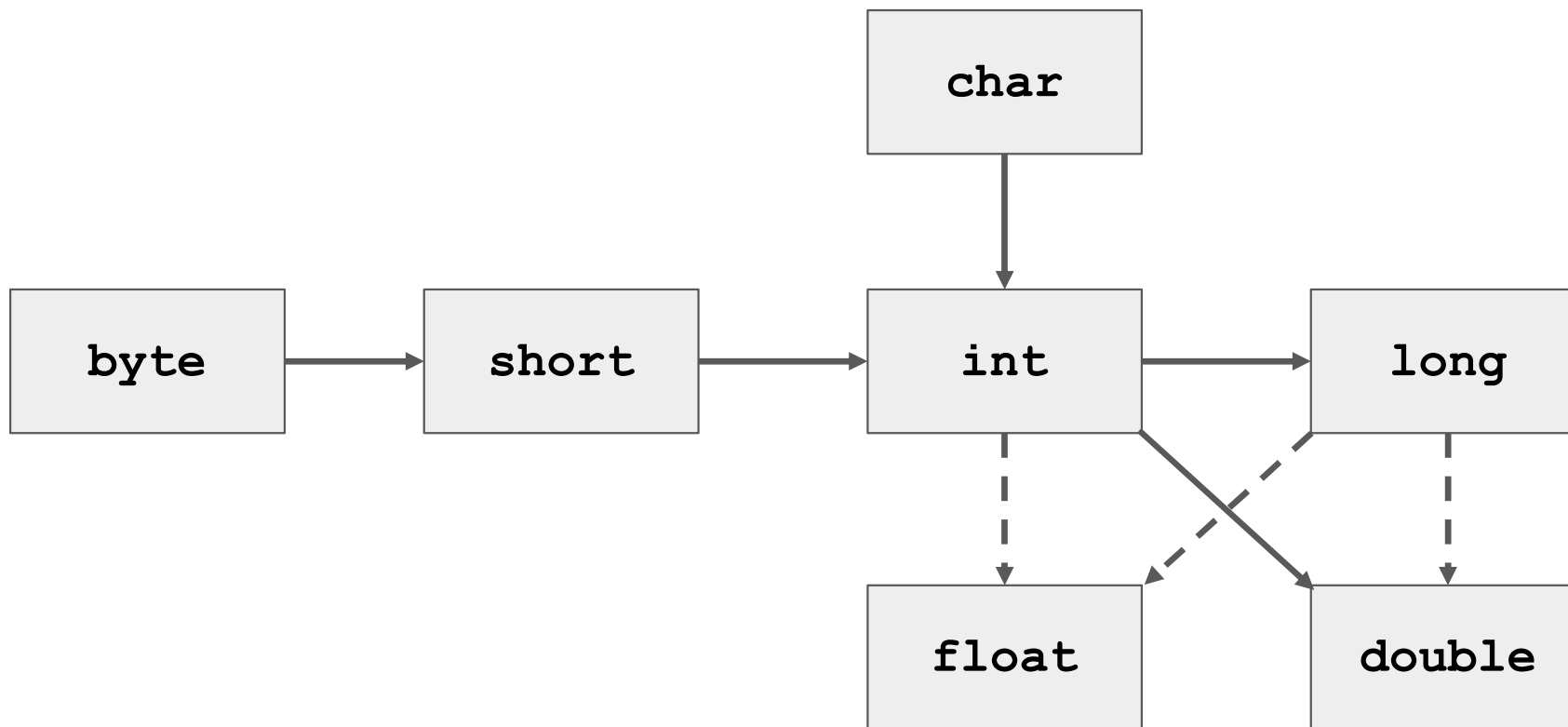
`int`

`long`

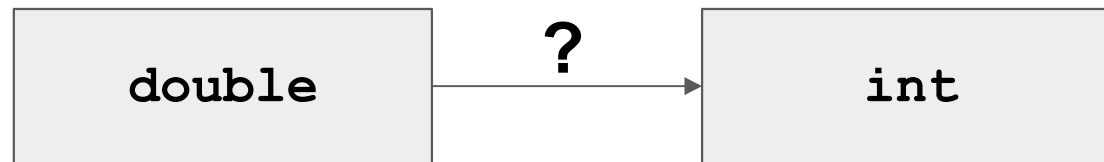
Типы данных. Подробности.

Тип данных	Размер (бит)	Значения
byte	8	от -128 до 127
short	16	от -32768 до 32767
char	16	от 0 до 65535
int	32	от -2147483648 до 2147483647
long	64	от -9223372036854775808 до 9223372036854775807
float	32	от -1.4e-45f до 3.4e+38f
double	64	от -4.9e-324 до 1.7e+308
boolean	1 ИЛИ 32	true ИЛИ false

Преобразование типов



Прив~~е~~дение типов



```
double e = 2.7;
```

```
int notE = (int) e;    // notE = 2
```

```
int roundE = (int) Math.round(e);    // roundE = 3
```

Операции

Унарные

-
--
++
~
!
[]

Бинарные

- +

/ *

| ||

& &&

== !=

```
int x = -y;
```

```
// унарный "минус"
```

```
int r = x - y;
```

```
// бинарный "минус"
```

Приоритеты операций

№	Оператор
1	[] . ()
2	! ~ ++ -- + - (приведение) new
3	* / %
4	+ - (бинарные)
5	>> << >>>
6	< <= > >= instanceof
7	== !=
8	&
9	^
10	
11	&&
12	
13	?:
14	= += -= *= /= %= = ^= <<= >>= >>>=

Математические функции

Функция	Конструкция
Абсолютное значение x	<code>Math.abs (x);</code>
Косинус x	<code>Math.cos (x);</code>
Синус x	<code>Math.sin (x);</code>
Экспонента	<code>Math.exp (x);</code>
Квадратный корень x	<code>Math.sqrt (x);</code>
Корень суммы квадратов (гипотенуза)	<code>Math.hypot (x, y);</code>
Натуральный логарифм	<code>Math.log (x);</code>
Округление до ближайшего	<code>Math.round (x);</code>
Возведение 'x' в степень 'y'	<code>Math.pow(x, y);</code>

<https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

Математические константы

- `Math.PI` `// 3.14159265358979323846`
- `Math.E` `// 2.7182818284590452354`

```
double res1 = 3 * Math.PI / 2;
```

```
double res2 = Math.log(5) * Math.E;
```

Строки

```
String name = "Alis";
```

```
String lastName = new String("Parker");
```

```
String a = "Java\u2122";           // Java™  
String b = "";                   // пустая строка
```

<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

Работа со строками

Конкатенация	<code>str.concat("!")</code>
Длина строки	<code>str.length()</code>
Деление строки по критерию	<code>str.split(", ")</code>
Получение символа по индексу	<code>str.charAt(5)</code>
Проверка наличия подстроки	<code>str.contains("мир")</code>
Проверка пустой строки	<code>str.isEmpty()</code>
Замена символов строки	<code>str.replace("Привет", "Здоров")</code>
Усечение пробельных символов	<code>str.trim()</code>
Вычленение подстроки	<code>str.substring(0, 5)</code>
Получение массива символов	<code>str.toCharArray()</code>

```
String str = "Привет, мир";
```

Работа со строками



```
double x = Double.parseDouble(str);
```

```
if (str.matches("-?\\d+(\\.\\d+)?")) { // регулярные выражения
    Double.parseDouble(str);
}
```

```
for(char c : str.toCharArray()) {
    // ручная проверка каждого символа
}
```

```
try {
    double d = Double.parseDouble(str);
    System.out.println("Input " + d);
} catch (Exception e) {
    System.out.println();
}
```

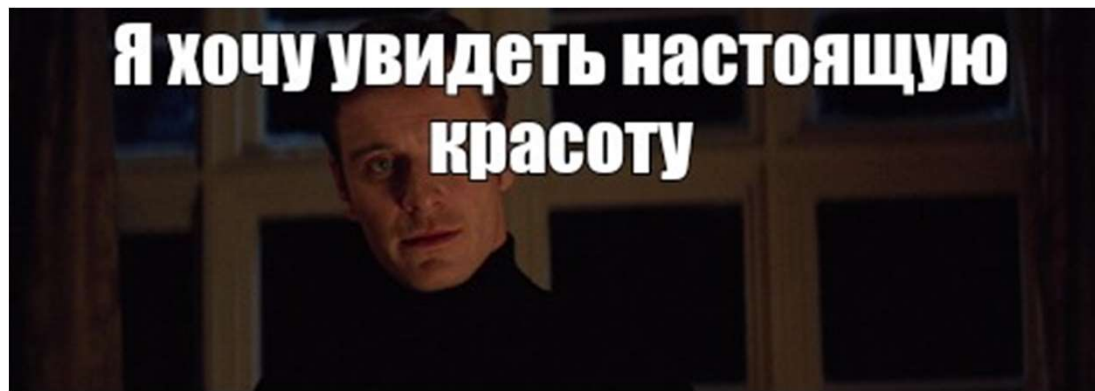
Форматированный вывод

```
double x = 5.683;
```

```
double y = Math.pow(x, 2.5);
```

```
System.out.printf("%.2f ^ 2.5 = %.3f", x, y);
```

Output: 5,68 ^ 2.5 = 76,890



Форматированный вывод

```
printf(String format, Object... args)
```

```
printf("%.4f", Object... args)
```

What is "format"?

```
%[argument_index$][flags][width][.precision]conversion
```

Эльфийско-русский словарь

% **[argument_index\$]** **[flags]** **[width]** **[.precision]** **conversion**

argument_index\$ — целое десятичное число, указывающее позицию аргумента в списке аргументов

flags — специальные символы для форматирования. Например, флаг "+" означает, что числовое значение должно включать знак +, флаг "-" означает выравнивание результата по левому краю, флаг «,» устанавливает разделитель тысяч у целых чисел.

[width] — положительное целое десятичное число, которое определяет минимальное количество символов, которые будут выведены

[.precision] — неотрицательное целое десятичное число с точкой перед ним. Обычно используется для ограничения количества символов.

conversion — это символ, указывающий, как аргумент должен быть отформатирован. Например **d** для целых чисел, **s** для строк, **f** для чисел с плавающей точкой.

* copy-paste from Java documentation

The end

