Faculty of Software Engineering and Computer Systems

Programming

Lecture #0
Introduction and more.

Instructor of faculty
Pismak Alexey Evgenievich
Gavrilov Anton Valerievich
Kronverksky Pr. 49, 1331 room
pismak@itmo.ru
avgavrilov@itmo.ru

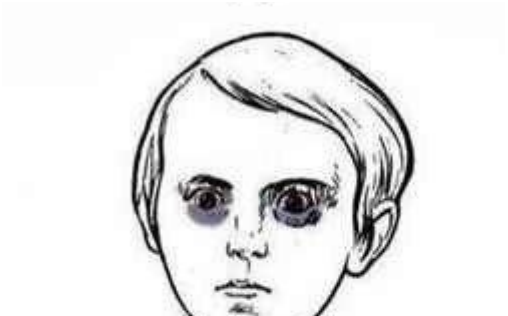
Saint-Petersburg

Class Introduction

- Who are your teacher(s)
  - Experience with Java
  - Experience with ITMO
  - Experience with IT
- Communication (tg chat)

Introduction. Course objectives

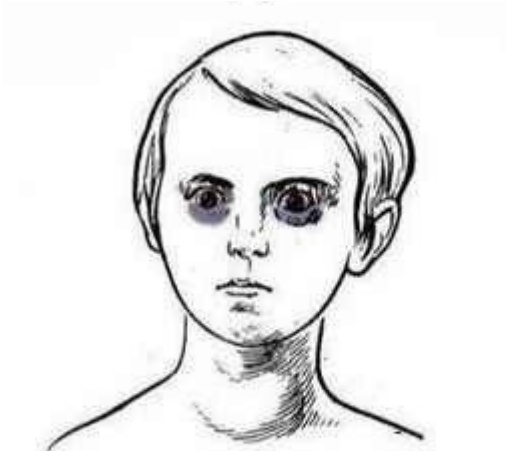
- Learn Java from basics to advanced technology
- In this course, you will acquire programming skills at a sufficient level to work in IT
- After this course, you will be able to stay awake for two nights in a row and feel ~~great~~



Introduction. Course objectives

After this course, you should be able to do the following:

- Create simple and complex Java-applications
- Learn to apply programming paradigms such as inheritance, encapsulation, and polymorphism. This skill is useful in programming in any language
- Learn to compile, build and run Java applications of any complexity
- …
- stay awake for two nights in a row and feel ~~great~~

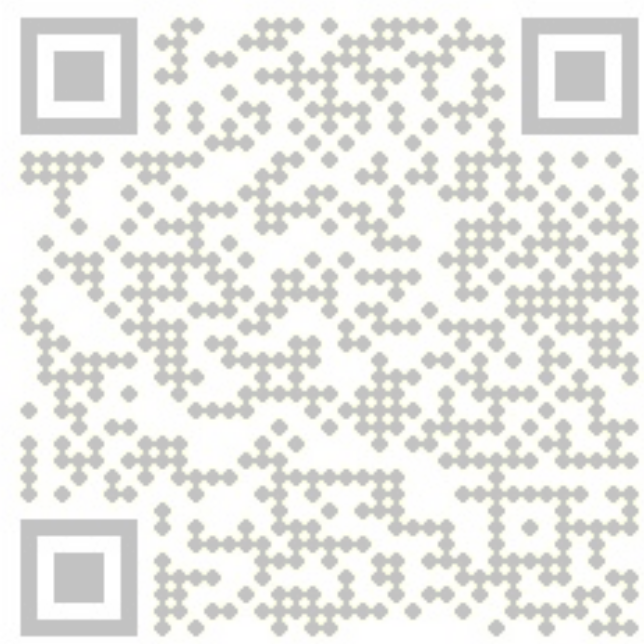


Introduction. Prerequisites

To successfully complete this course, you must know:

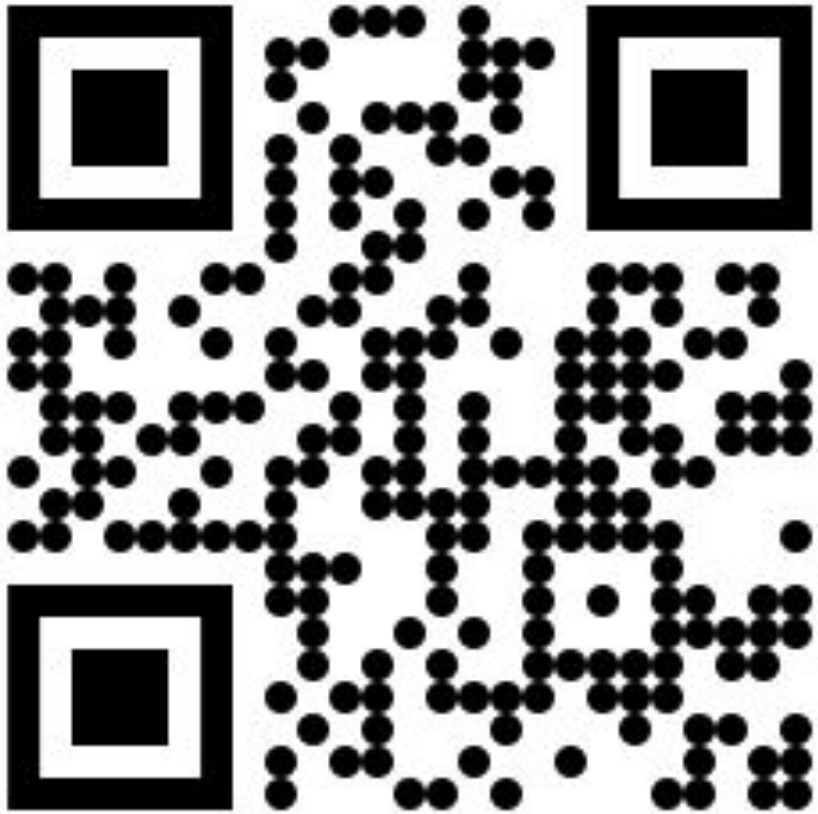
- basics of programming and informatics
- about two hours of time each day for this course
- how to stay awake **one** night and feel great

Introduction. Grades



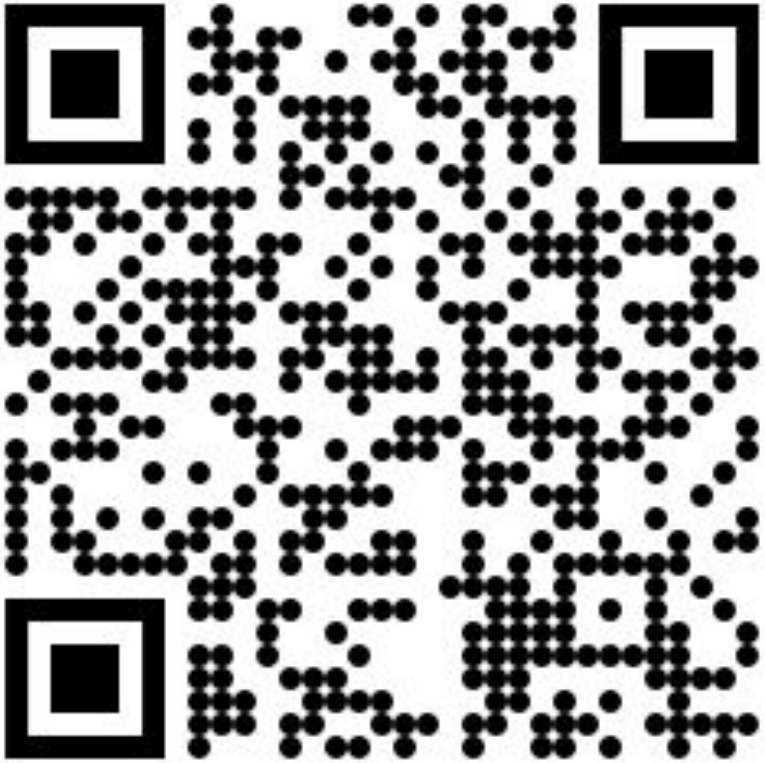
Introduction. Materials and news

se.ifmo.ru/courses/programming



tg channel

Introduction. Lectures & guides (partly)



* Если сможешь

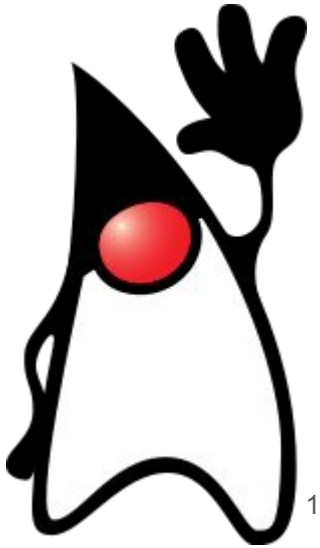
Introduction. Course parts

- How many labs
- How many tests
- Lectures
- Exams

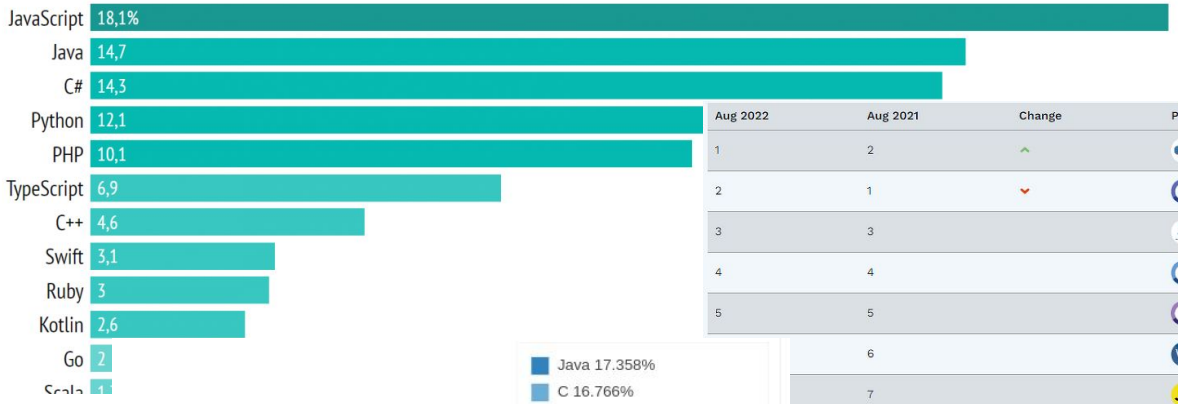
Introduction. Tools

- Text editor (SublimeText, Notepad++)
- JDK
- IDE (intelliJ IDEA, Eclipse, NetBeans)
- CMD/Terminal (Putty, WinSCP, shell, helios)

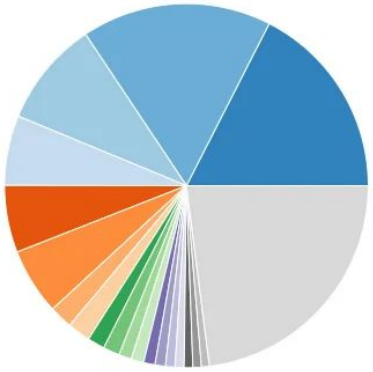
Why Java?

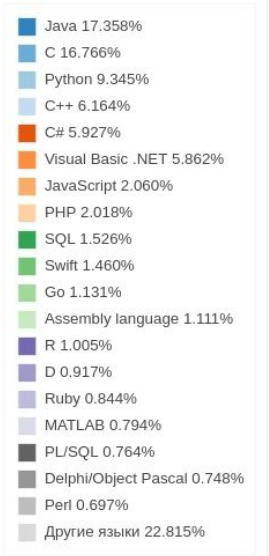


Introduction. Why Java?



Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	↑	Python	15.42%	+3.56%
2	1	↓	C	14.59%	+2.03%
3	3		Java	12.40%	+1.96%
4	4		C++	10.17%	+2.81%
5	5		C#	5.59%	+0.45%
	6		Visual Basic	4.99%	+0.33%
	7		JavaScript	2.33%	-0.61%
	9	↑	Assembly language	2.17%	+0.14%
	10	↑	SQL	1.70%	+0.23%
	8	↓	PHP	1.39%	-0.80%



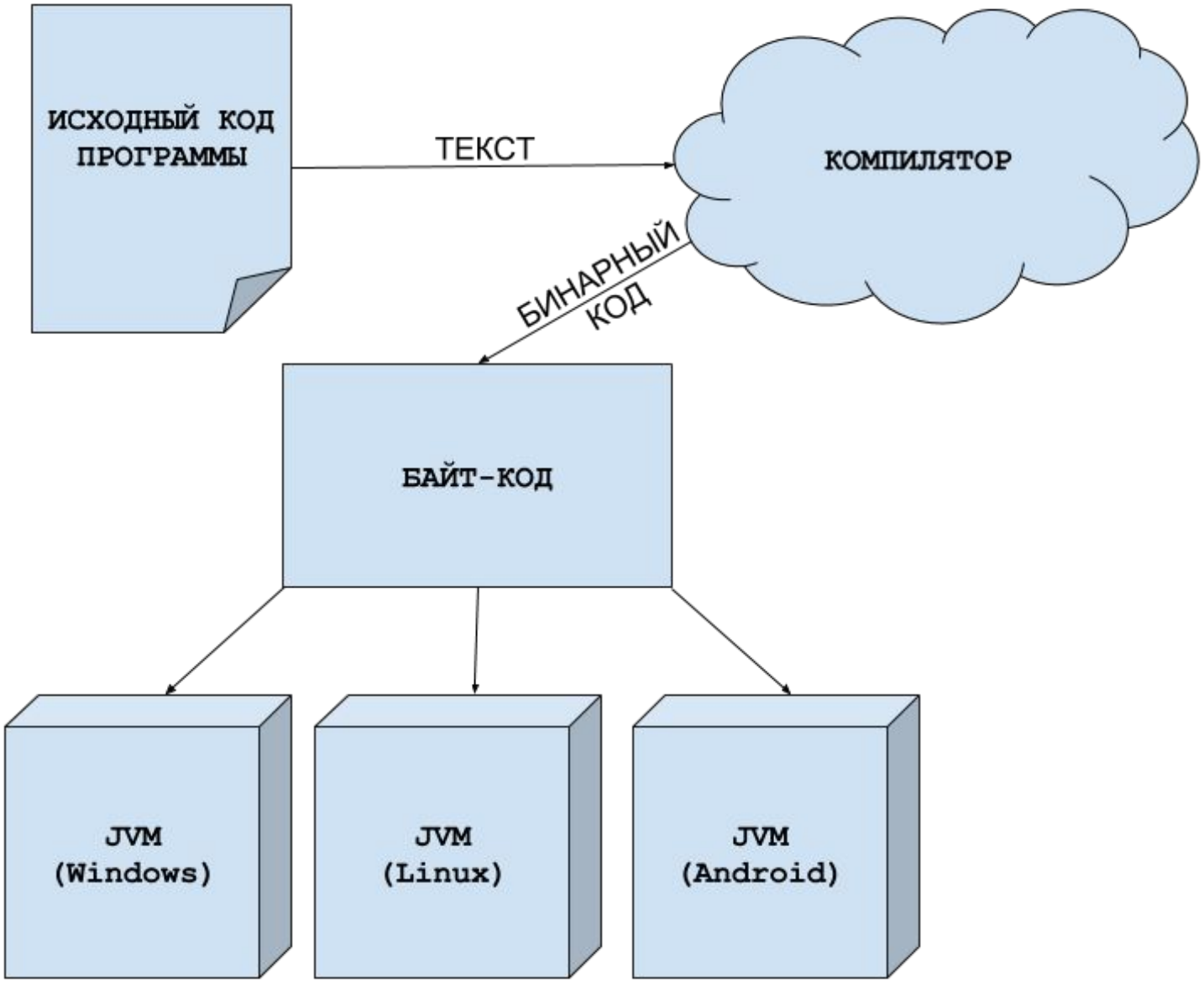


Rank	Language	Type	Score
1	Python		100.0
2	Java		95.3
3	C		94.6
4	C++		87.0
5	JavaScript		79.5
6	R		78.6
7	Arduino		73.2
8	Go		73.1
9	Swift		70.5
10	Matlab		68.4

Introduction. Why Java?

- high threshold for entry into IT (low competition)
- Java is a very large and complex product. Knowing Java is easy to master other areas
- cross-platform

Introduction. Cross-platform



Introduction. Cross-platform

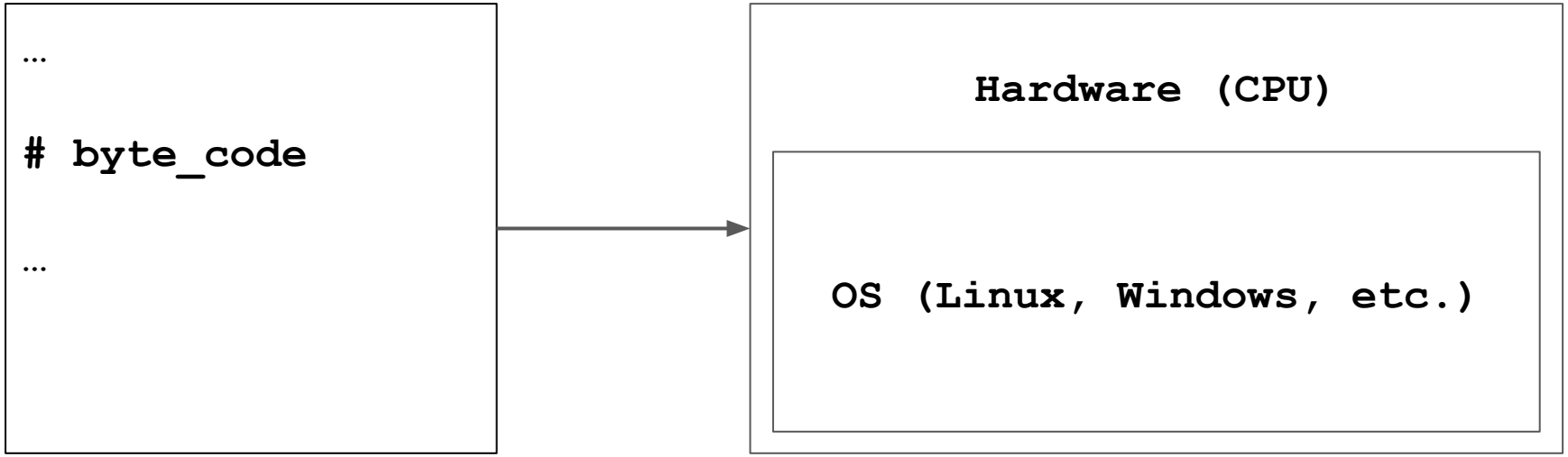
```
public class Object
    public Object() {}

    …
}
```



```
Compiled from "Object.java"
public class java.lang.Object
{
  public java.lang.Object();
    Code:
       0: return
  public boolean
equals(java.lang.Object);
    Code:
       0: aload_0
       1: aload_1
       2: if_acmpne     9
       5: iconst_1
       6: goto          10
       9: iconst_0
      10: ireturn
  …
```

Introduction. Cross-platform



Java

Java. Comments

Однострочные комментарии

```
// комментарии
```

Многострочные комментарии

```
/*
комментарии
*/
```

Специальные комментарии (javadoc)

```
/**
*
* @author  James Gosling
*/
```

Java. First application

Main.java

```
1.  public class Main {
2.
3.      public static void main(String[] args) {
4.
5.          System.out.println("Внимание!");
6.
7.      }
8.
9.  }
```

Java. Simple math-application

```
 1.  public class SimpleProgram {
 2.
 3.      public static void main(String[] args) {
 4.
 5.          var x = 5.0D;
 6.
 7.          var result = Math.pow(x, 2.5);
 8.
 9.          System.out.println(result);
10.
11.      }
12.
13.  }
```

Java. Program is written. What's next?

JDK
- compiler
- debugger
- byte-code tools
- JVM
- standard library

Java. Program is written. What's next?

1. Compilation

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

2. Packaging to jar

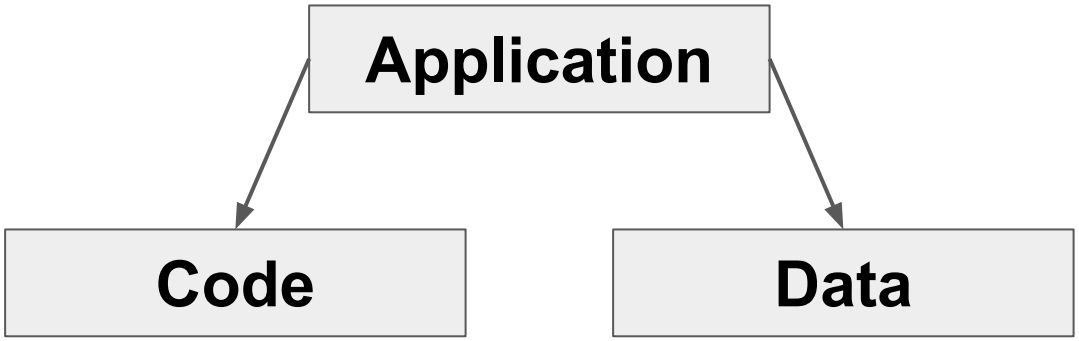
```
jar   -cfm   app.jar   MANIFEST.mf   *.class
```

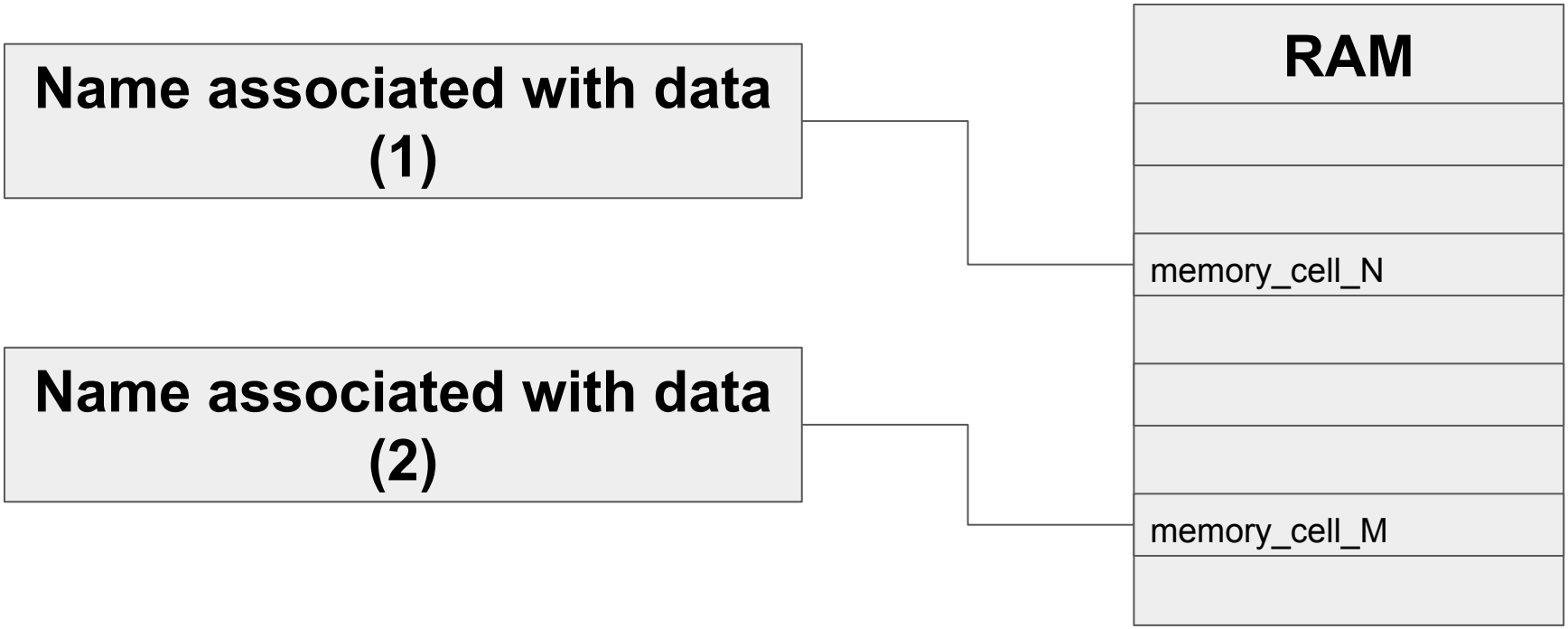
3. Run

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

4. Debugging

Java. Variables and constants



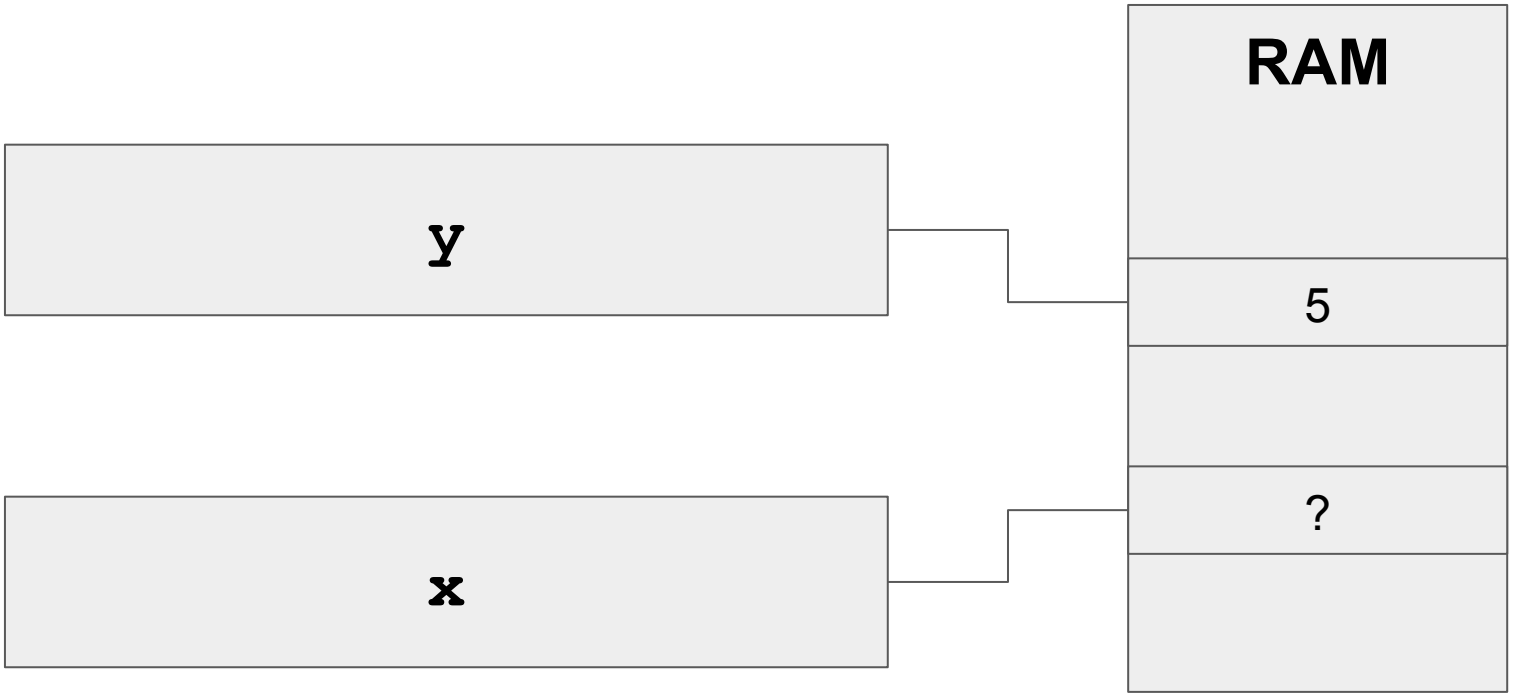


Java. Variables and constants

```
int x;

int y = 5;


data_type variable_name [ =  default_value ] ;
```



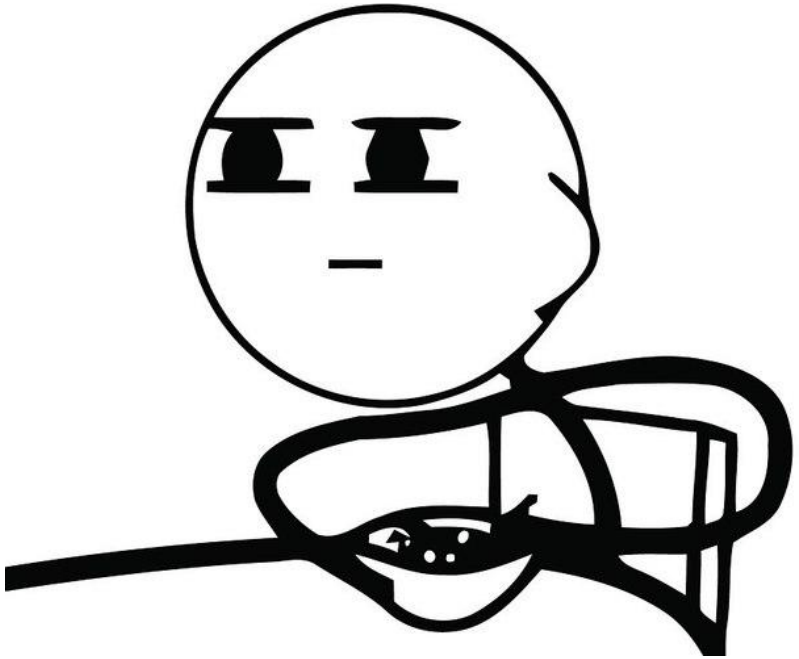
Java. Variables and constants

```
final double E = 5.0;

final double R;
// …
R = 5.0;
```

what is it for?

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

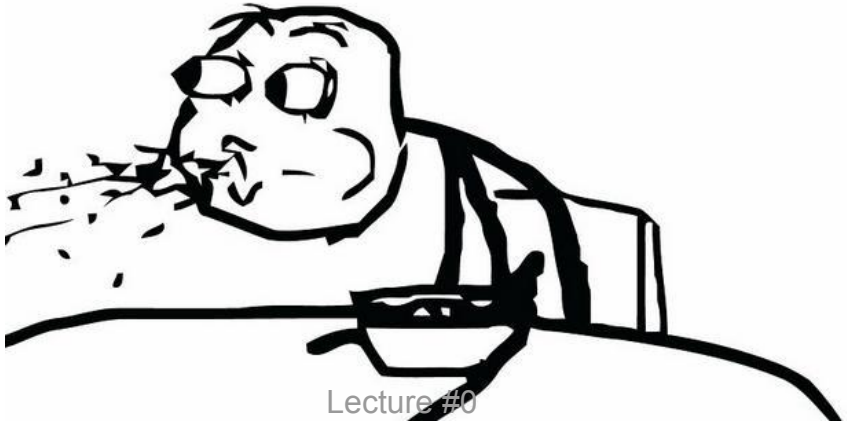


Java. Variables and constants

what is it for?

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

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



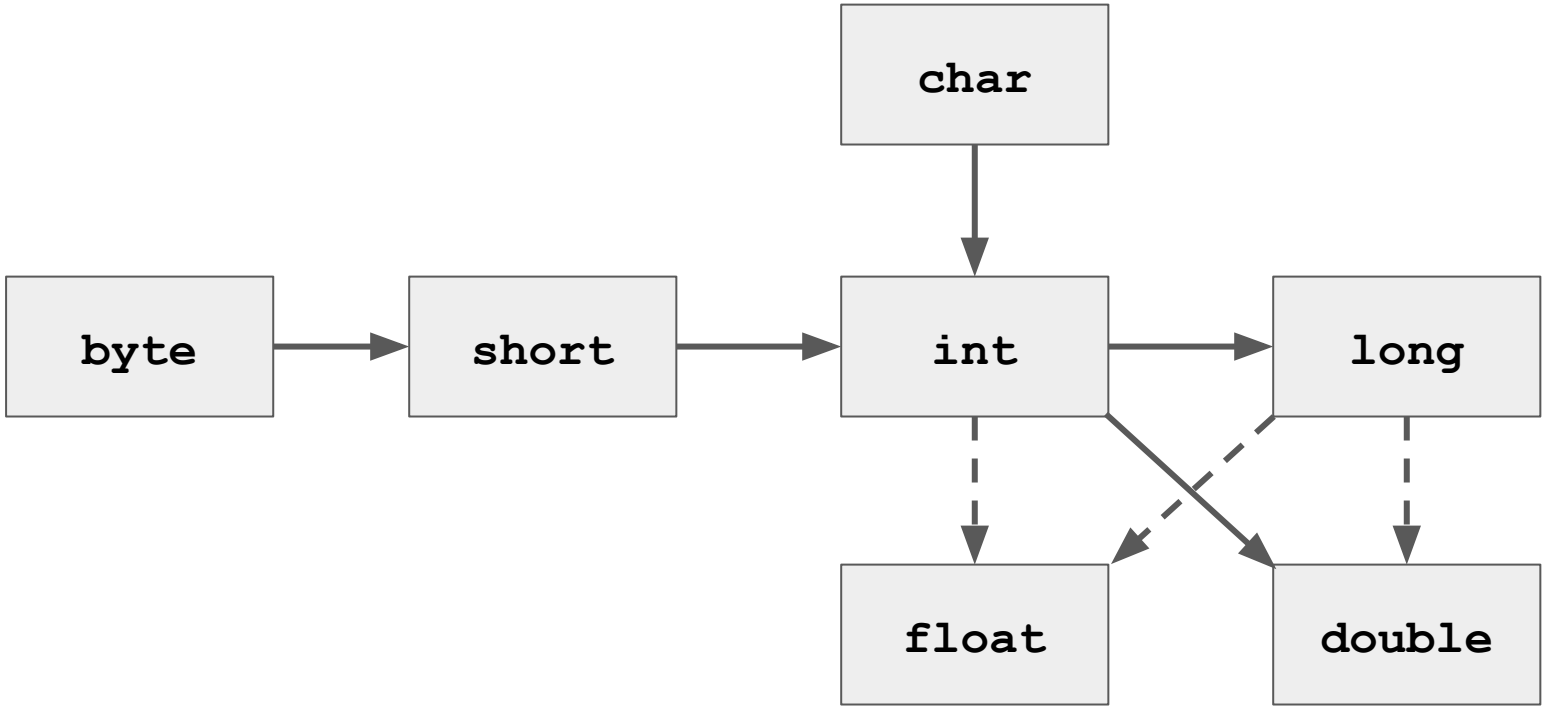
Java. Data types

Целочисленные	Вещественные	Логические	Объекты
byte	double	boolean	
short	float		
char			
int			
long			

Java. Data types

Data type	Size (bit)	Values
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 or 32	true or false

Java. Convert data types



Java. Casting data types


```
double e = 2.7;

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


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

Java. Operators

Unary	Binary
-	- +
--	
++	/ *
~	
!	| ||
[]	
	& &&
	== !=

```
int x = -y;          // унарный "минус"

int r = x - y;       // бинарный "минус"
```

Java. Operators

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

Java. Math functions

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

`https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Math.html`

Java. Math constants

- `Math.PI` `// 3.141592653589793238462643383279`
- `Math.E` `// 2.718281828459045235360287471352`

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

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

Java. Just in time test

```
 1.  public class PracticValiables {
 2.
 3.     public static void main(String[] args) {
 4.
 5.        byte x = 127;
 6.
 7.        x++;
 8.
 9.        System.out.println(x);
10.
11.     }
12.  }
```

128 **1** **-1** **127**

Java. Strings

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

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

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

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html

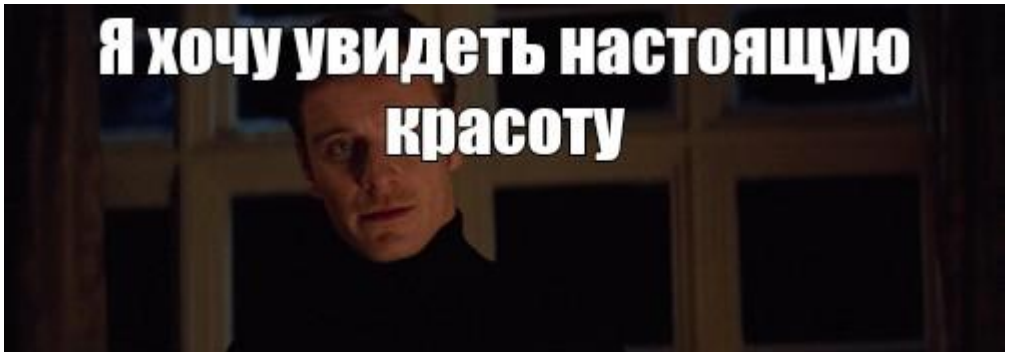
Java. Formatting output

```
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
```



Java. Formatting output

```
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**

Java. Summary about you next steps

- You will do the lab work at home
  - Write code
  - Copy it to server (helios.cs.ifmo.ru)
  - Compile and run it
- Your teacher may give you additional practice tasks
- Examination of lab work
  - Be ready to hear the question, made up from unfamiliar words
  - Try to find answer and repeat discuss with teacher