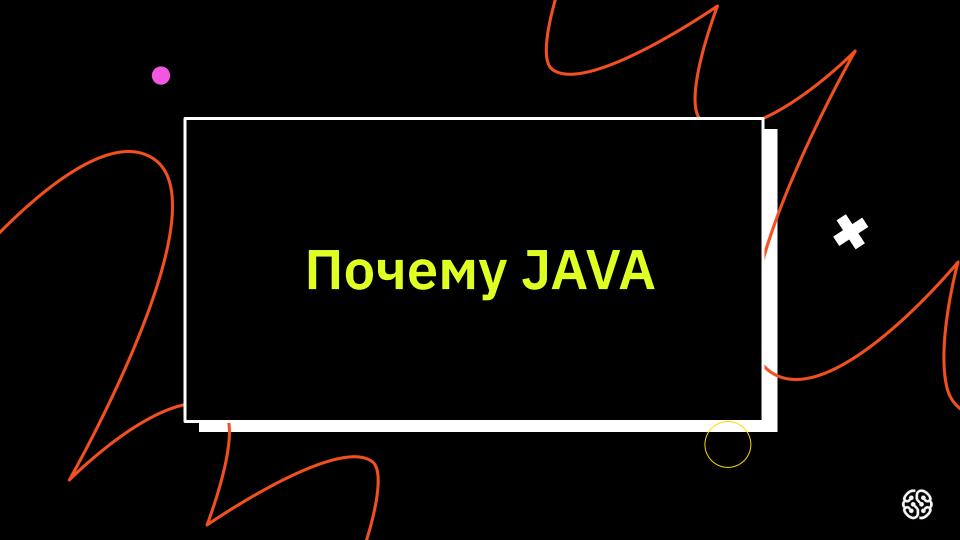




Простое и любопытное





**1.** Топ 3 ЯП в мире



- **1.** Топ 3 ЯП в мире
- 2. Библиотеки



- **1.** Топ 3 ЯП в мире
- 2. Библиотеки
- 3. Кроссплатформенность



- **1.** Топ 3 ЯП в мире
- 2. Библиотеки
- **3.** Кроссплатформенность
- 4. Тысячи вакансий разных направлений



- **1.** Топ 3 ЯП в мире
- **2.** Библиотеки
- **3.** Кроссплатформенность
- 4. Тысячи вакансий разных направлений
- **5.** Начало Android'a



- **1.** Топ 3 ЯП в мире
- **2.** Библиотеки
- 3. Кроссплатформенность
- 4. Тысячи вакансий разных направлений
- **5.** Hачало Android'a
- **6.** Безопасность

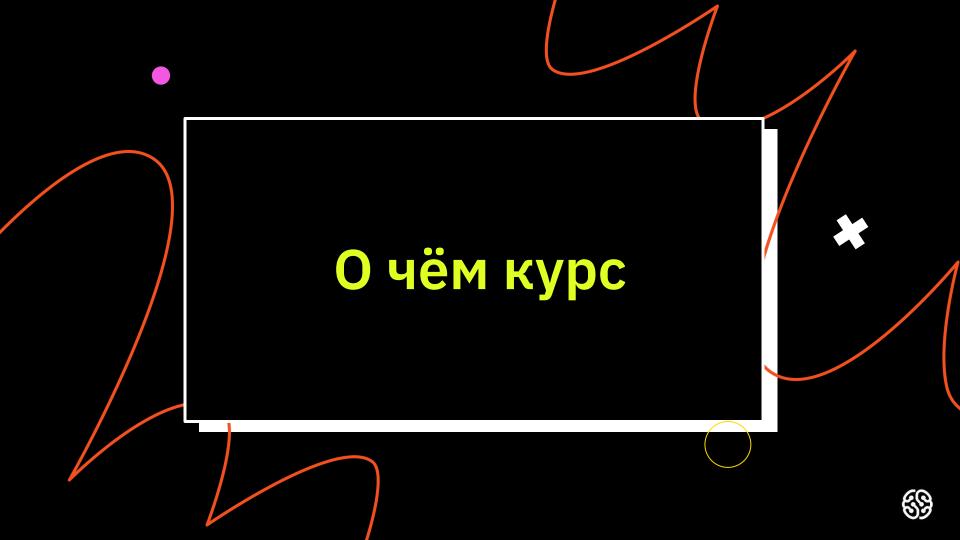


- **1.** Топ 3 ЯП в мире
- **2.** Библиотеки
- Кроссплатформенность
- 4. Тысячи вакансий разных направлений
- **5.** Hачало Android'a
- **6.** Безопасность
- **7.** ΟΟΠ



- **1.** Топ 3 ЯП в мире
- **2.** Библиотеки
- Кроссплатформенность
- 4. Тысячи вакансий разных направлений
- Начало Android'a
- Безопасность
- **7.** ΟΟΠ
- 8. Многопоточность, как она есть

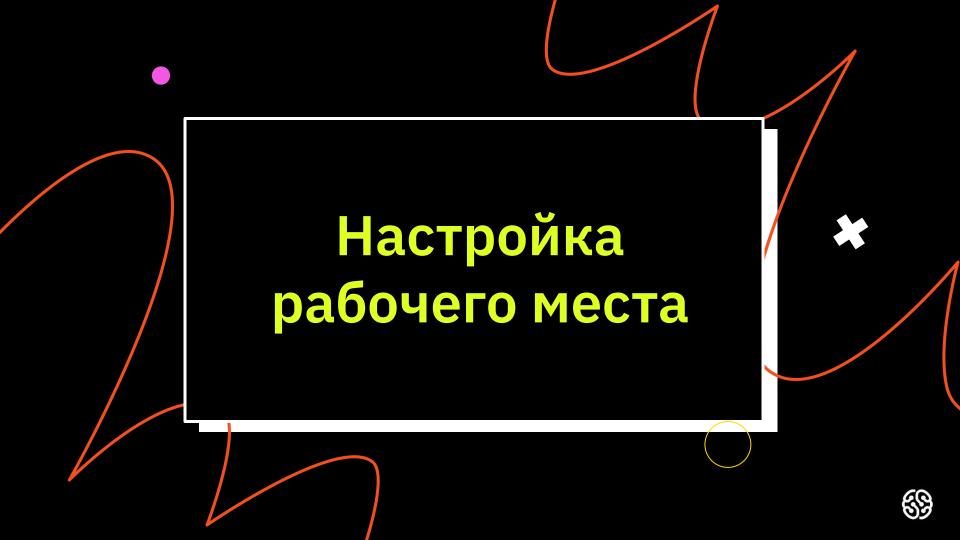




# Настройка рабочего места

- 1. Основы основ
  - Типы, базовые конструкции
- 2. Детальный разбор некоторых АРІ
  - Файлы и работа с потоками ввода\вывода
- 3. Погружение в Java Collection API
  - List'ы и не только





# Настройка рабочего места

#### Шаг 1:

Java JDK <a href="https://www.oracle.com/java/technologies/downloads/">https://www.oracle.com/java/technologies/downloads/</a>

#### Шаг 2:

Extension Pack VS Code

https://marketplace.visualstudio.com/items?itemName=vscjava.vscode-javapack

#### При желаниии:

https://www.jetbrains.com/ru-ru/idea/
println("Hello world");

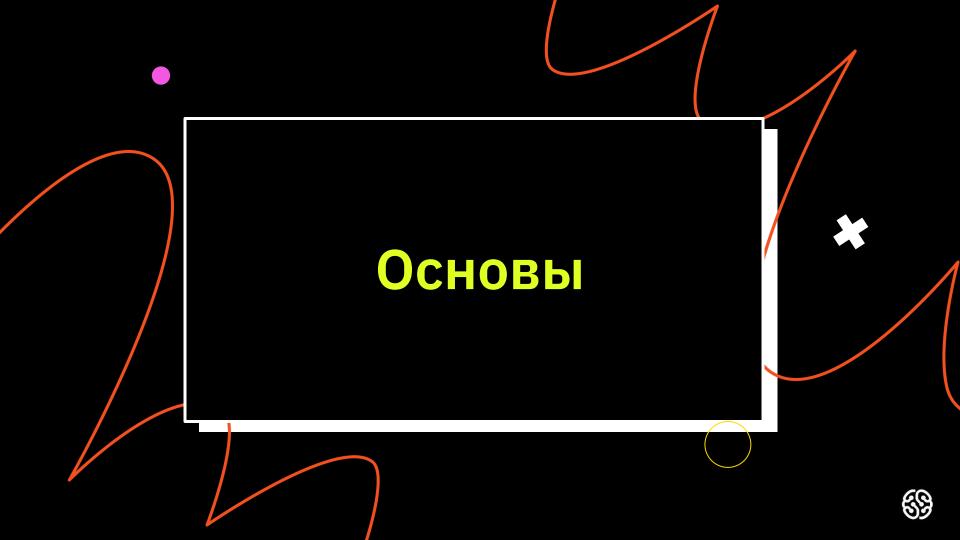


# Настройка рабочего места

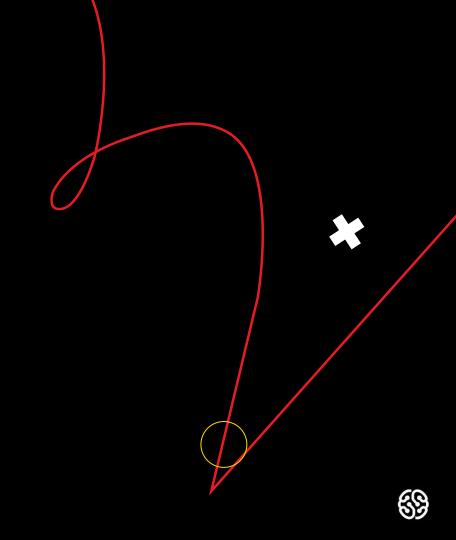
```
Program
public class Program {
   public static void main(String[] args) {
       System.out.println("Goodbye world");
```

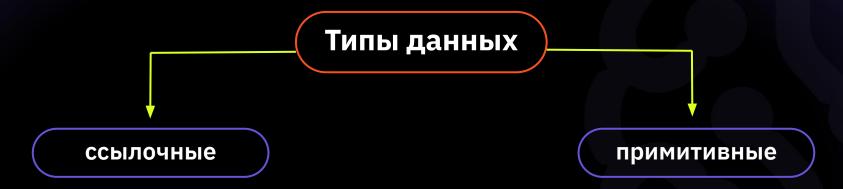
#### Структура простой программы





# Типы данных и переменные











Создание переменной

```
<тип> <идентификатор>;
<идентификатор> = <значение>;
```

да, да.. снова ; 😁



```
class Program
 public static void main(String[] args) {
     short age = 10;
     int salary = 123456;
                                     //10
     System.out.println(age);
     System.out.println(salary); //123456
```



```
class Program
 public static void main(String[] args) {
     float e = 2.7f;
     double pi = 3.1415;
     System.out.println(e);  // 2.7
     System.out.println(pi); // 3.1415
```



```
class Program
  public static void main(String[] args) {
      char ch = '1';
      System.out.println(Character.isDigit(ch)); // true
      ch = 'a';
      System.out.println(Character.isDigit(ch)); // false
```

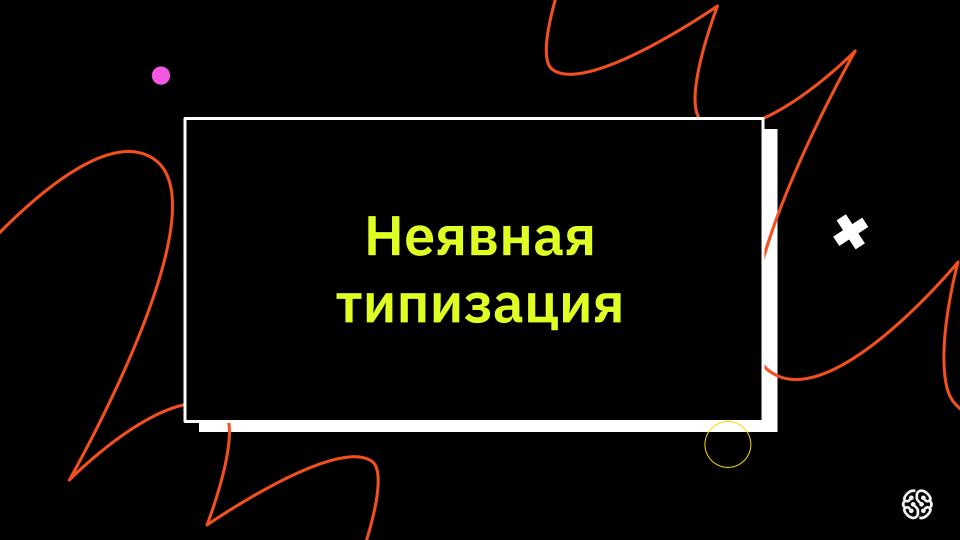


```
public class Program {
   public static void main(String[] args) {
      boolean flag1 = 123 <= 234;
      System.out.println(flag1); // true
      boolean flag2 = 123 >= 234 || flag1;
      System.out.println(flag2); // true
      boolean flag3 = flag1 ^ flag2;
      System.out.println(flag3); // false
   }
}
```



# Основы: типы данных \*Строки не так просты





#### Неявная типизация

```
public class Program {
    public static void main(String[] args) {
       var a = 123;
       System.out.println(a); // 123
       var d = 123.456;
       System.out.println(d); // 123.456
    }
}
```



#### Неявная типизация

```
public class Program {
   public static void main(String[] args) {
        var a = 123;
        System.out.println(a); // 123
        var d = 123.456;
        System.out.println(d); // 123.456
        System.out.println(getType(a)); // Integer
        System.out.println(getType(d)); // Double
        d = 1022;
       System.out.println(d); // 1022
        //d = "mistake";
        //error: incompatible types:
        //String cannot be converted to double
    static String getType(Object o){
        return o.getClass().getSimpleName();
```





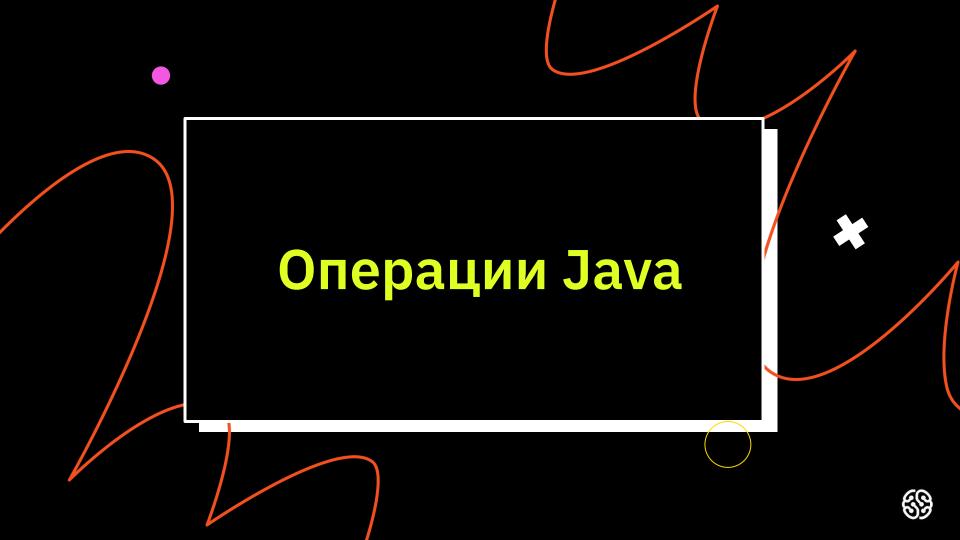
# Основы: классы-обертки

Примитив	Обертка
int	Integer
short	Short
long	Long
byte	Byte
float	Float
double	Double
char	Character
boolean	Boolean



```
class Program
{
   public static void main(String[] args) {
       System.out.println(Integer.MAX_VALUE); // 2147483647
       System.out.println(Integer.MIN_VALUE); // -2147483648
   }
}
```





#### Операции Java

- Присваивание: =
- Арифметические: \*, /, +, -, %, ++, --
- Операции сравнения: <, >, ==, !=, >=, <=</li>
- Логические операции: ||, &&, ^,!
- Побитовые операции <<, >>, &, |, ^





#### Массивы

#### Одномерные

```
public class Program {
   public static void main(String[] args) {
     int[] arr = new int[10];
     System.out.println(arr.length); // 10

     arr = new int[] { 1, 2, 3, 4, 5 };
     System.out.println(arr.length); // 5
}
```



#### Массивы

#### **Многомерные**

```
public class Program {
    public static void main(String[] args) {
        int[] arr[] = new int[3][5];
        for (int[] line : arr) {
            for (int item : line) {
                System.out.printf("%d ", item);
            System.out.println();
```

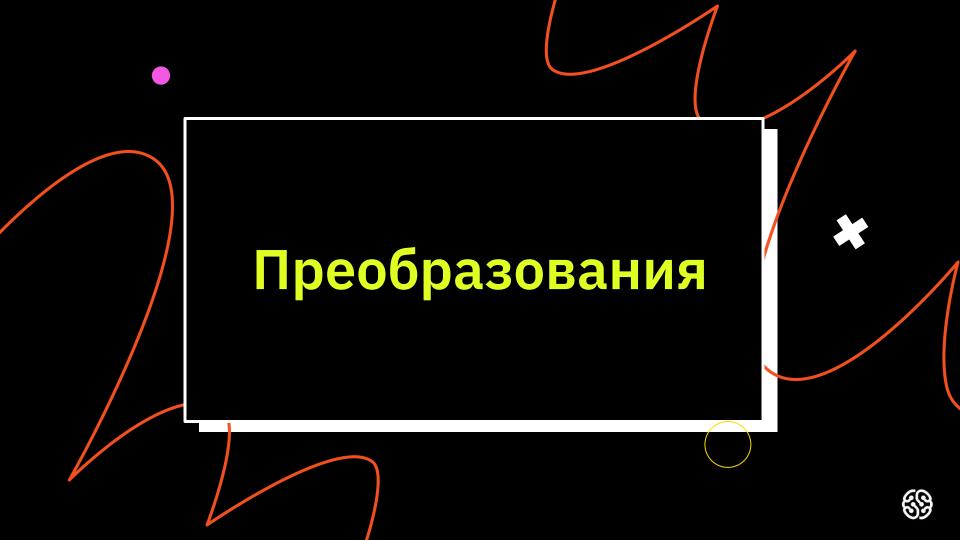


#### Массивы

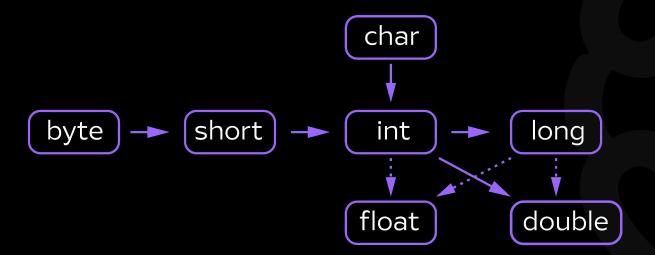
#### Многомерные

```
public class Program {
    public static void main(String[] args) {
        int[][] arr = new int[3][5];
        for (int i = 0; i < arr.length; i++) {</pre>
            for (int j = 0; j < arr[i].length; j++) {</pre>
                 System.out.printf("%d ", arr[i][j]);
            System.out.println();
```





# Преобразования





#### Можно

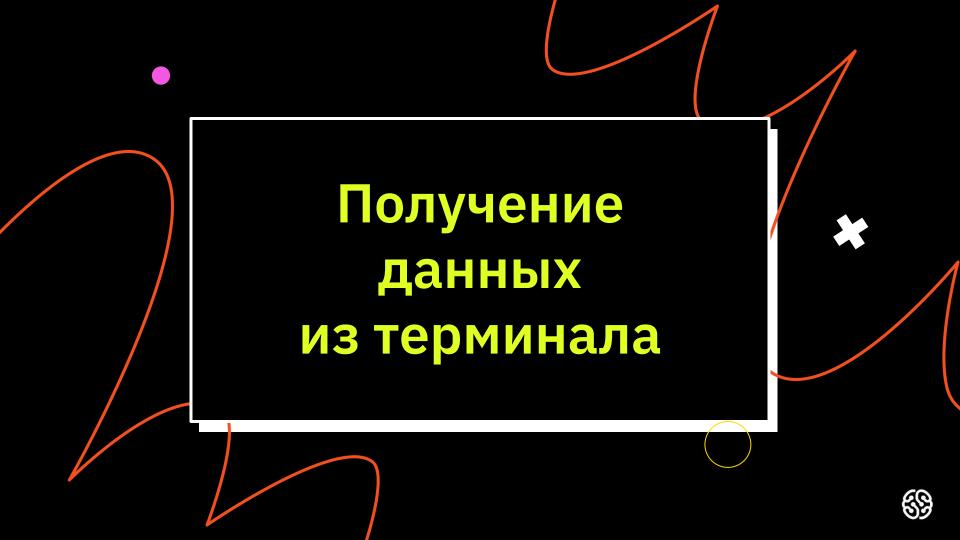
```
public class Program {
   public static void main(String[] args) {
       int i = 123; double d = i;
       System.out.println(i); // 123
       System.out.println(d); // 123.0
       d = 3.1415; i = (int)d;
       System.out.println(d); // 3.1415
       System.out.println(i); // 3
       d = 3.9415; i = (int)d;
       System.out.println(d); // 3.9415
       System.out.println(i); // 3
       byte b = Byte.parseByte("123");
       System.out.println(b); // 123
       b = Byte.parseByte("1234");
       System.out.println(b); // NumberFormatException: Value out of range
```



#### HO

```
class Program
{
  public static void main(String[] args) {
  int[] a = new int[10];
  double[] d = a; // ИЗУЧАЕМ ковариантность и контравариантность
  }
}
```





#### Получение данных из терминала

#### Строки

```
import java.util.Scanner;
public class Program {
  public static void main(String[] args) {
       Scanner iScanner = new Scanner(System.in);
       System.out.printf("name: ");
       String name = iScanner.nextLine();
       System.out.printf("Привет, %s!", name);
       iScanner.close();
```



#### Получение данных из терминала

#### Некоторые примитивы

```
import java.util.Scanner;
public class Program {
    public static void main(String[] args) {
        Scanner iScanner = new Scanner(System.in);
        System.out.printf("int a: ");
        int x = iScanner.nextInt();
        System.out.printf("double a: ");
        double y = iScanner.nextDouble();
        System.out.printf("%d + %f = %f", x, y, x + y);
        iScanner.close();
} }
```

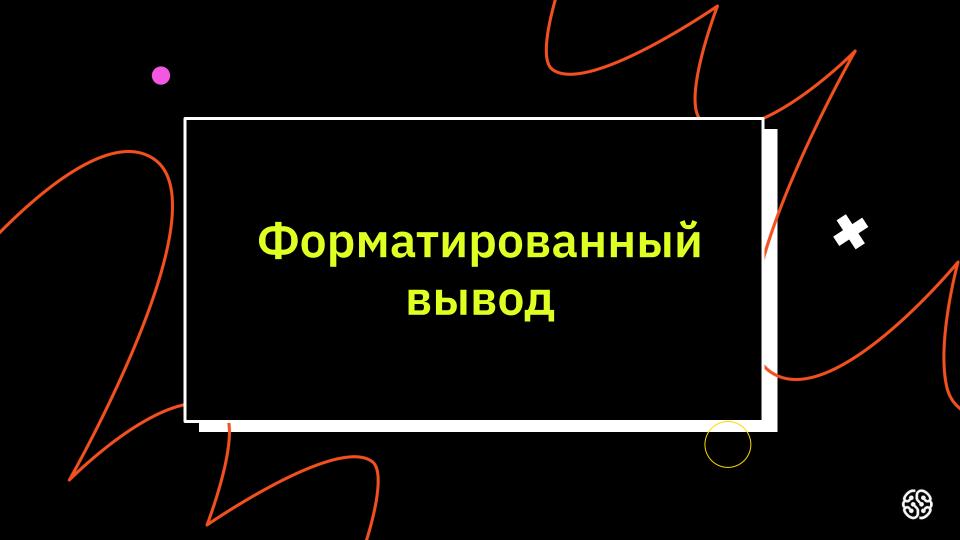


#### Получение данных из терминала

#### Проверка на соответствие получаемого типа

```
import java.util.Scanner;
public class Program {
   public static void main(String[] args) {
        Scanner iScanner = new Scanner(System.in);
        System.out.printf("int a: ");
        boolean flag = iScanner.hasNextInt();
        System.out.println(flag);
        int i = iScanner.nextInt();
        System.out.println(i);
        iScanner.close();
    }
}
```





# Форматированный вывод

```
public class Program {
   public static void main(String[] args) {
      int a = 1, b = 2;
      int c = a + b;
      String res = a + " + " + b + " = " + c;
      System.out.println(res);
   }
}
```



# Форматированный вывод

```
public class Program {
   public static void main(String[] args) {
     int a = 1, b = 2;
     int c = a + b;
     String res = String.format("%d + %d = %d \n", a, b, c);
     System.out.printf("%d + %d = %d \n", a, b, c);
     System.out.println(res);
}
```



#### Виды спецификаторов

%d: целочисленных значений

%х: для вывода шестнадцатеричных чисел

%f: для вывода чисел с плавающей точкой

%е: для вывода чисел в экспоненциальной форме,

например, 3.1415е+01

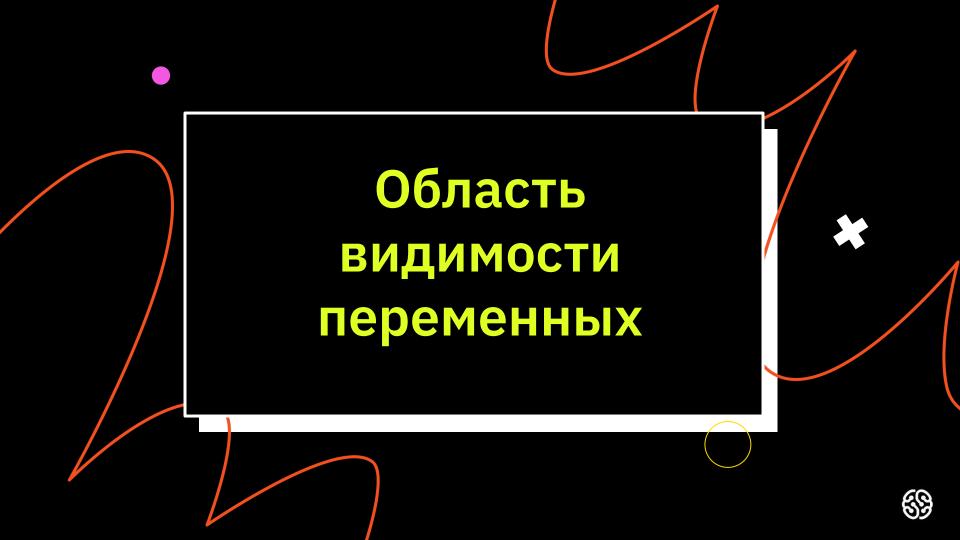
<u>%с: для вывода одиночного символа</u>

%s: для вывода строковых значений



#### Виды спецификаторов





#### Область видимости переменных

У переменных существует понятие «область видимости». Если переменную объявили внутри некоторого блока фигурных скобок { }, то снаружи этого блока переменная будет недоступна.

```
public class Program {
   public static void main(String[] args) {
        int i = 123;
        System.out.println(i);
   }
   System.out.println(i); // error: cannot find symbol
}
```





#### Функции и методы

Функции и методы — это технически одно и то же. Функции могут не принадлежать классам, а методы принадлежат. В java все функции являются методами.

Описание

Вызов

Возвращаемое значение

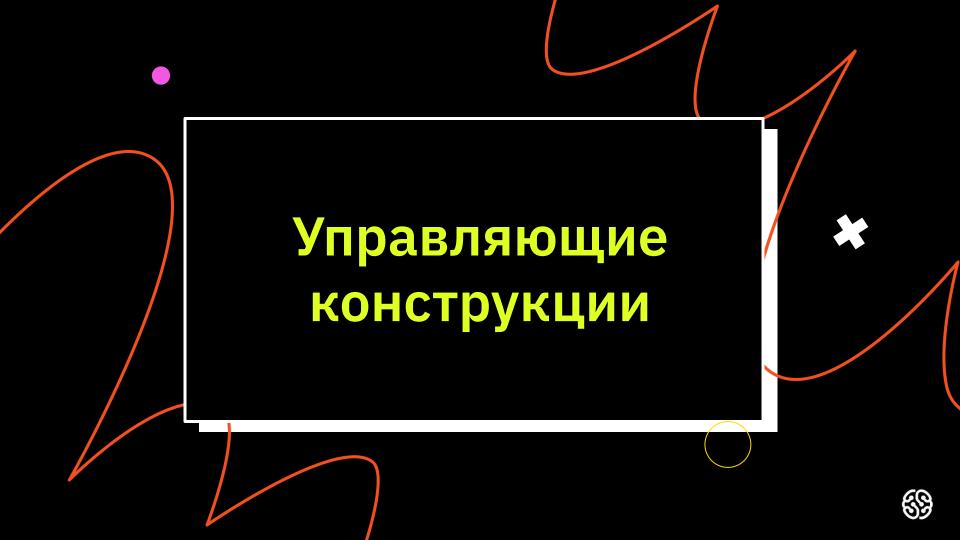
Рекурсия



#### Функции и методы

```
public class Program {
    static void sayHi() {
        System.out.println("hi!");
    static int sum(int a, int b) {
        return a+b;
    static double factor(int n) {
        if(n==1)return 1;
        return n * factor(n-1);
    public static void main(String[] args) {
        sayHi(); // hi!
        System.out.println(sum(1, 3)); // 4
        System.out.println(factor(5)); // 120.0
    }}
```





# Управляющие конструкции: условный оператор

```
public class Program {
    public static void main(String[] args) {
        int a = 1;
        int b = 2;
        int c;
        if (a > b) {
            c = a;
        } else {
            c = b;
        System.out.println(c);
```



# Управляющие конструкции: условный оператор

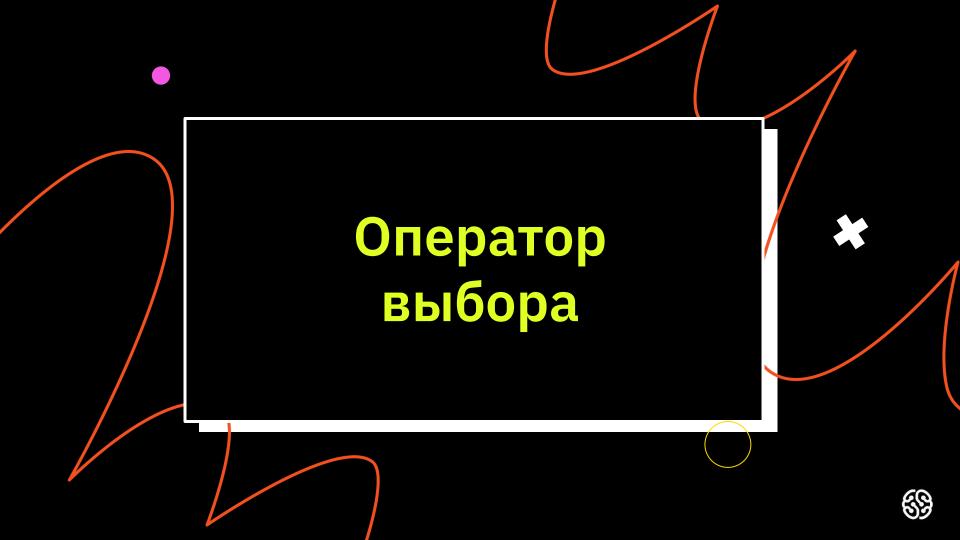
```
public class Program {
    public static void main(String[] args) {
        int a = 1;
        int b = 2;
        int c = 0;
        if (a > b) c = a;
        if (b > a) c = b;
        System.out.println(c);
```



# Управляющие конструкции: тернарный оператор

```
public class Program {
   public static void main(String[] args) {
     int a = 1;
     int b = 2;
     int min = a < b ? a : b;
     System.out.println(min);
   }
}</pre>
```

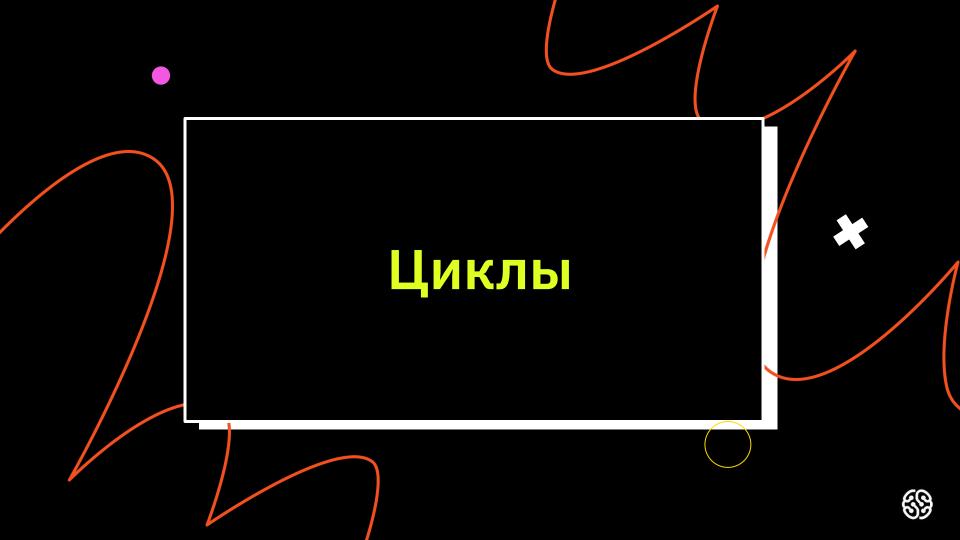




# Оператор выбора

```
import java.util.Scanner;
public class Program {
    public static void main(String[] args) {
        int mounth = value;
        String text = "";
        switch (mounth) {
            case 1:
                text = "Autumn";
                break;
            default:
                text = "mistake";
                break;
        System.out.println(text);
        iScanner.close();
```





#### Циклы

Цикл — это многократное выполнение одинаковой последовательности действий.

В java доступны следующие циклы:

- цикл while;
- цикл do while;
- цикл for; и его модификация for in



# Цикл while

```
public class Program {
    public static void main(String[] args) {
        int value = 321;
        int count = 0;
        while (value != 0) {
            value /= 10;
            count++;
        System.out.println(count);
```



#### Цикл do while

```
public class Program {
    public static void main(String[] args) {
        int value = 321;
        int count = 0;
        do {
            value /= 10;
            count++;
        } while (value != 0);
        System.out.println(count);
```



#### Циклы

#### continue, break

Операторы для управления циклами — continue и break.

Выполнение следующей итерации цикла — continue.

Прерывание текущей итерации цикла — break.

\* ближайшего к оператору



## Оператор цикла for

```
public class Program {
    public static void main(String[] args) {
        int s = 0;
        for (int i = 1; i <= 10; i++) {
            s += i;
        System.out.println(s);
```



#### Вложенные циклы

```
public class Program {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5; j++) {
                System.out.print("* ");
            System.out.println();
```

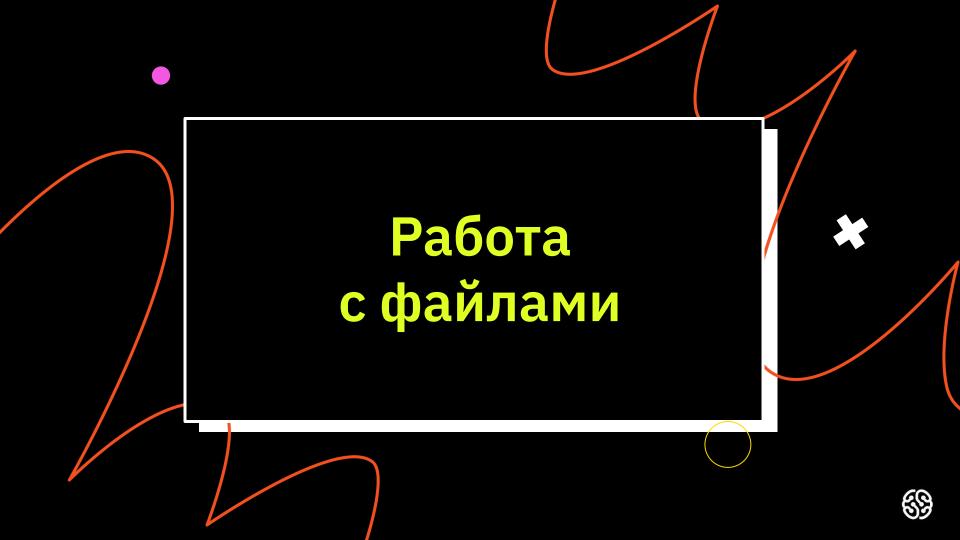


#### for:

#### Работает только для коллекций

```
public class Program {
    public static void main(String[] args) {
        int arr[] = new int[10];
        for (int item : arr) {
            System.out.printf("%d ", item);
        System.out.println();
```





# Работа с файлами

#### Создание и запись\ дозапись

```
import java.io.FileWriter;
import java.io.IOException;
public class Program {
    public static void main(String[] args) {
        try (FileWriter fw = new FileWriter("file.txt", false)) {
            fw.write("line 1");
            fw.append(' \ n');
            fw.append('2');
            fw.append(' \ n');
            fw.write("line 3");
            fw.flush();
        } catch (IOException ex) {
            System.out.println(ex.getMessage());
```



# Работа с файлами

#### Чтение, Вариант посимвольно

```
import java.io.*;
public class Program {
    public static void main(String[] args) throws Exception {
        FileReader fr = new FileReader("file.txt");
        int c;
        while ((c = fr.read()) != -1) {
            char ch = (char) c;
            if (ch == ' \n')
                System.out.print(ch);
            else {
                System.out.print(ch);
```



# Работа с файлами

#### Вариант построчно

```
import java.io.*;
public class Program {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new FileReader("file.txt"));
        String str;
        while ((str = br.readLine()) != null) {
            System.out.printf("== %s ==\n", str);
        br.close();
```



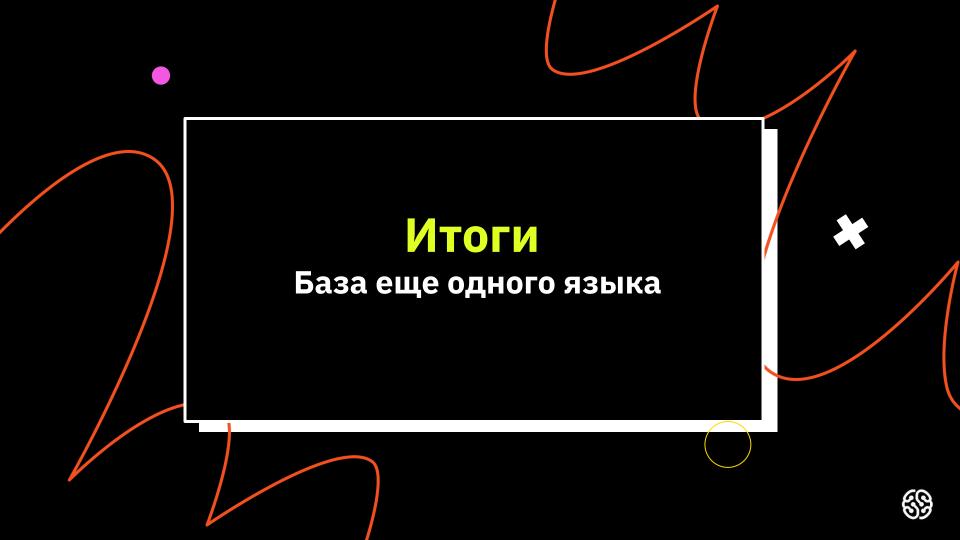
#### Задачи для демонстрации

#### Задачи для самоконтроля

- 1. Задана натуральная степень k. Сформировать случайным образом список коэффициентов (значения от 0 до 100) многочлена многочлен степени k.

  \*Пример: k=2 => 2\*x² + 4\*x + 5 = 0 или x² + 5 = 0 или 10\*x² = 0
- 2. Даны два файла, в каждом из которых находится запись
- многочлена. Сформировать файл содержащий сумму многочленов.





# Спасибо // за внимание /

