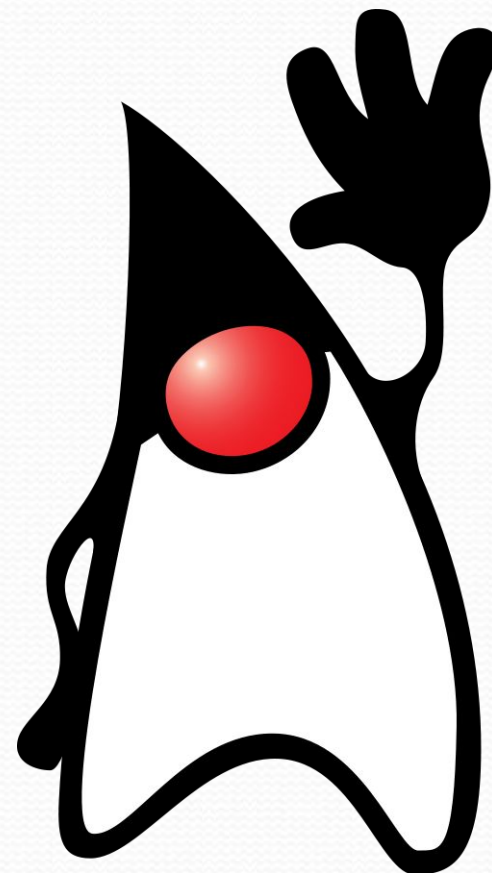


# Курс: «Программирование на Java»



Java™





- Рогаткин  
Роман  
Олегович
- [vk.com/teacher\\_1cb](https://vk.com/teacher_1cb)
- [teacher.1cb@yandex.ru](mailto:teacher.1cb@yandex.ru)
- +7 (901) 703-38-80 (Теле2)



# Базовые понятия в программировании

- 1. Оператор – действие или операция с помощью которой выполняется та или иная конструкция.  
Например в выражении « $2 + 2$ » - знак « $+$ » - является оператором.
- 2. «Операнд» - переменная, которая стоит с одной из сторон от оператора. Делится на:
  - Левый Операнд ( Стоит слева от оператора )
  - Правый Операнд ( Стоит справа от оператора )

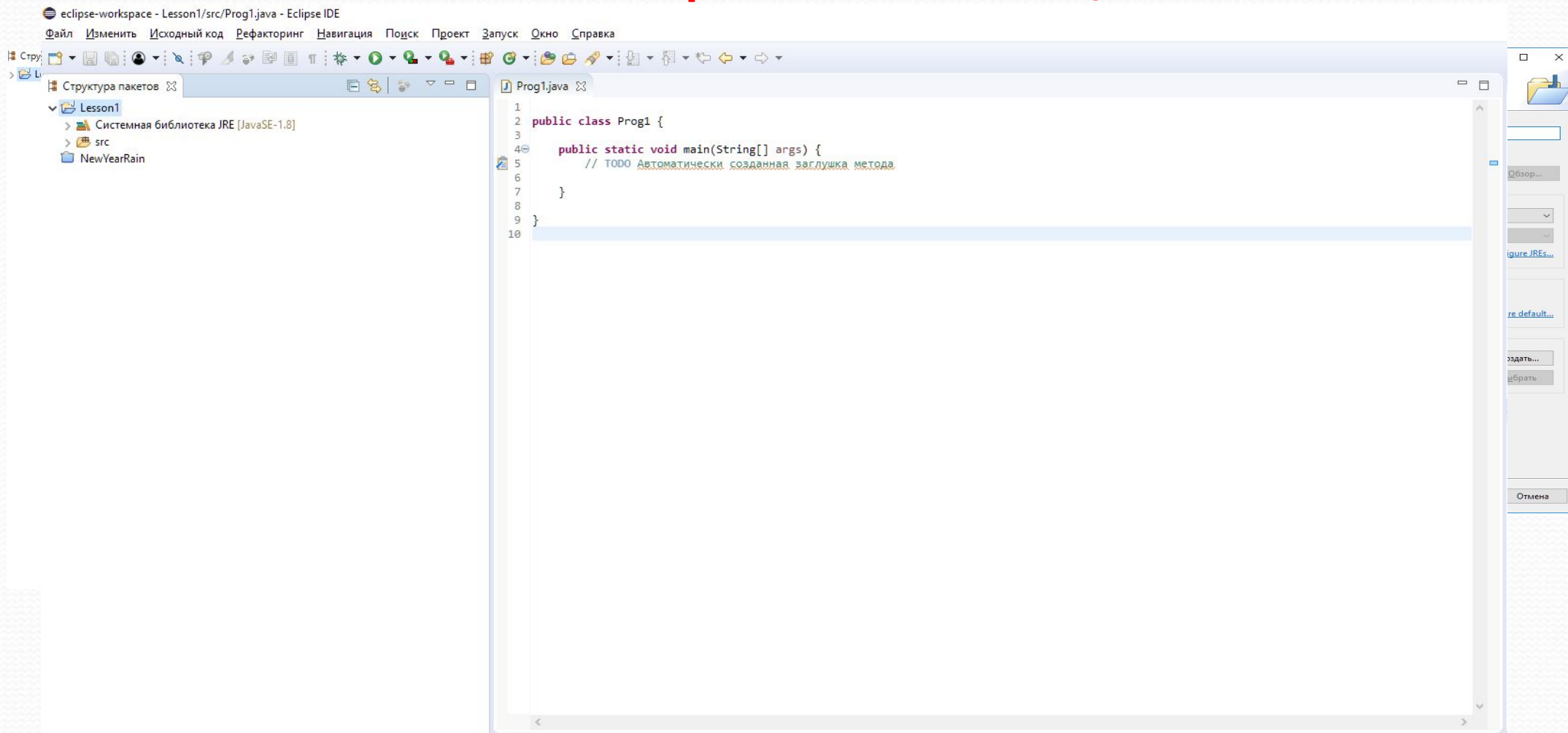
Например в выражении « $3 + 7$ »

Число « $3$ » - является левым операндом.

Число « $7$ » - является правым операндом.



# Создание проекта в Eclipse







```
SimpleMistakes.java
1 package mistakes;
2
3 public class SimpleMistakes {
4
5     public static void main(String[] args) {
6
7         int = 3;
8         int b = 7;
9
10        B = 8;
11        double a = 6
12
13        Int c = 1.5;
14
15        String str = h;
16
17        String str1 = "Hello";
18
19        str1 = str1 * 87;
20
21        System.out.println(str1);
22
23
24
25
26
27
```

Problems @ Javadoc Declaration Console

<terminated> SimpleMistakes [Java Application] C:\Program Files\Java\jdk-8.0.181\bin\javaw.exe (5 окт. 2018 22:01:59)

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

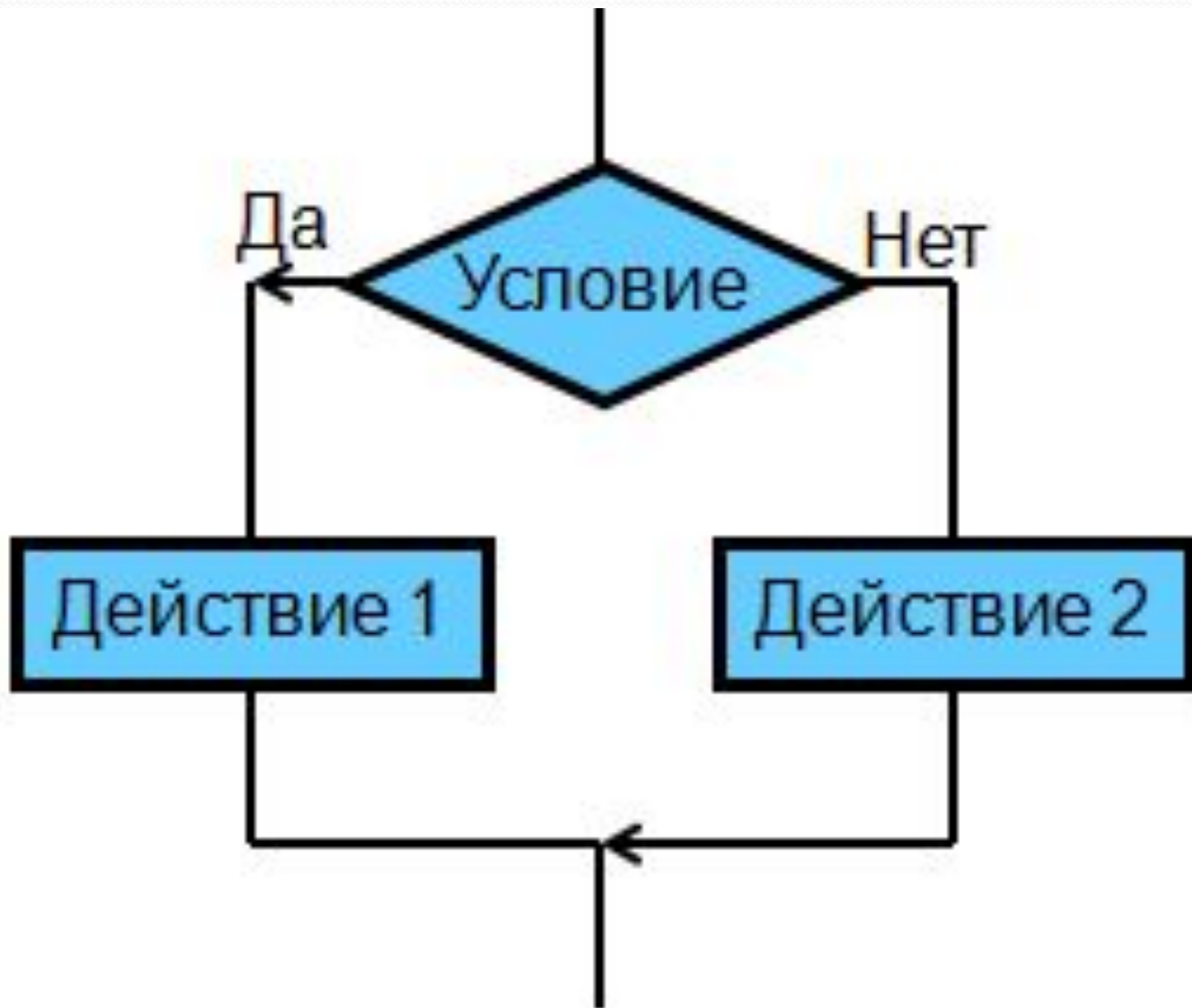
- 1 Syntax error, insert ". class" to complete Expression
- 2 The left-hand side of an assignment must be a variable
- 3 B cannot be resolved to a variable
- 4 Syntax error, insert ";" to complete LocalVariableDeclarationStatement
- 5 Int cannot be resolved to a type
- 6 h cannot be resolved to a variable
- 7 The operator \* is undefined for the argument type(s) String, int
- 8 Syntax error, insert "}" to complete MethodBody

at mistakes.SimpleMistakes.main(SimpleMistakes.java:7)

## Условные конструкции (ветвления)

- **Условная конструкция (ветвление)** — алгоритмическая конструкция, в которой, в зависимости от результата проверки условия («правда» или «ложь»), предусмотрен выбор одной из двух последовательностей действий (ветвей).
- Команда ветвления записывается так:
- Если (Условие) if(Условие)  
(Если условие правдиво) То<Действие1> { Действие 1}  
(Если условие ложно) Иначе<Действие2> else { Действие 2 }





# Операторы сравнения в условных конструкциях

- « > » - оператор «больше?». Проверяет является ли правый операнд больше левого операнда.
- « < » - оператор «меньше?». Проверяет является ли правый операнд больше меньше операнда.
- « == » - оператор «равно?». Проверяет равны ли правый операнд и левый операнд.
- « != » - оператор «не равно?». Проверяет не равны ли правый операнд и левый операнд.
- « >= » - оператор «больше либо равно?» Проверяет является ли правый операнд больше левого операнда или равен ему.
- « <= » - оператор «меньше либо равно?» Проверяет является ли левый операнд меньше правого операнда или равен ему.



# Пример условных конструкций в Java

```
if ( 100==100) {  
    System.out.println("Сто равно сотне");  
}  
else {  
    System.out.println("Сто не равно сотне");  
}
```

```
if ( 100!=100) {  
    System.out.println("Сто не равно сотне");  
}  
else {  
    System.out.println("Сто равно сотне");  
}
```

```
if ( 100<20) {  
    System.out.println("Сто меньше двадцати");  
}  
else {  
    System.out.println("Сто больше либо равно двадцати");  
}
```

```
if ( 100>20) {  
    System.out.println("Сто больше двадцати");  
}  
else {  
    System.out.println("Сто меньше либо равно двадцати");  
}
```

# Пример условных конструкций в Java

```
if ( 100<=20) {  
    System.out.println("Сто меньше либо равно двадцати");  
}  
else {  
    System.out.println("Сто больше двадцати");  
}
```

```
if ( 100>=20) {  
    System.out.println("Сто больше либо равно двадцати");  
}  
else {  
    System.out.println("Сто меньше двадцати");  
}
```



# Циклы

- Цикл — Конструкция, предназначенная для организации многократного исполнения набора действий.  
В программировании используются при необходимости повторять действие, пока определенное условие остается истинным.

# Циклы делятся на три типа

- 1. Цикл с предусловием (Условием входа в цикл) –  

```
while(Условие){  
    Действие  
}
```
- 2. Цикл с постусловием (Условием выхода из цикла) –  

```
do {  
    Действие  
} while (Условие)
```

 Прим. используется редко.
- 3. Цикл со счетчиком (Выполнятся пока счетчик не достигнет нужного значения).  

```
for( int i = 0; i < 5; i++){  
    Действие  
}
```

 (выполняется 5 раз)



# Примеры программ с циклами в Java

● `int a = 5;`

```
while(a < 10){  
    System.out.println(a);  
    a = a+2;  
}
```

● `int a = 100;`

```
do{  
    a = a+2;  
    System.out.println(a);  
}while(a < 5){
```

---

```
for(int i = 7; i<=11; i++){  
    System.out.println(i);  
}
```

# Примеры программ с циклами в Java

- ```
int a = 10;
while(a == 10){
    System.out.println(a);
    a = a+2;
}
```

- -----

- ```
int a = 10;
while(a != 10){
    System.out.println(a);
    a = a+2;
}
```

- ```
int a = 2;

do{
    a = a+2;
    System.out.println(a);
}while(a ==4){
```