

# Java Basic

lecture #5. Boolean. Logic expression

Mentor: Il'yas Miftakhov

## lecture #5. Boolean. Logic expression

- Основные понятия
- Реляционные операторы Java
- Логические операторы
  - Оператор логического И (&&)
  - Оператор «логическое ИЛИ» (||):
  - Оператор логического НЕ(!):

## Основные понятия - операторы

- **Операторы** составляют основной строительный блок любого языка программирования.
- **Java** предоставляет множество типов операторов, которые можно использовать в зависимости от необходимости выполнения различных вычислений и функций.
- **Операторы** классифицируются на основе предоставляемой ими функциональности.
  1. Арифметические операторы
  2. Унарные операторы
  3. Оператор присваивания
  4. Реляционные операторы
  5. Логические операторы
  6. Тернарный оператор
  7. Побитовые операторы
  8. Операторы сдвига

## Реляционные операторы

- Реляционные операторы Java — это набор бинарных операторов, используемых для проверки отношений между двумя операндами
- Возвращают логический результат после сравнения

Синтаксис:

`var1 <relation operator> var2`

Примеры:

1. `operator (==) -> var1 == var2`
2. `operator(!=) -> var1 != var2`
3. `operator(>) -> var1 > var2`
4. `operator(<) -> var1 < var2`
5. `operator(>=) -> var1 >= var2`
6. `operator(<=) -> var1 <= var2`

## Логические операторы

- Логические операторы используются для выполнения логических операций «И», «ИЛИ» и «НЕ»
- Используется для проверки условия или нескольких условий для принятия решения
- Второе условие не оценивается, если первое ложно

Синтаксис:

cond1 <logical operator> cond2

Примеры:

1. Оператор И (AND) ( && ) -> если (cond1 && cond2 ) -> если true выполнить, иначе не делать
2. Оператор ИЛИ (OR) ( || ) -> если (cond1 || cond2) -> если один из них true, выполнить, иначе не выполнять
3. Оператор НЕ (NOT) ( ! ) -> !(var1 < var2) -> false, если a меньше, чем var2

# Логические операторы

## 1. Логический оператор «И» (&&)

- `cond1 && cond2` возвращает `true`, когда оба `cond1` и `cond2` истинны (т.е. ненулевые).

## 2. Логический оператор «ИЛИ» (||)

- Если хотя бы один из двух дает истину, оператор возвращает истину.
- Чтобы результат был ложным, оба условия должны возвращать `false`.

## 3. Логический оператор НЕ (!)

- если условие ложно, операция возвращает истину, а когда условие истинно, операция возвращает ложь.

# SOLID

## S

Принцип единственной ответственности (single responsibility principle)

Для каждого класса должно быть определено единственное назначение. Все ресурсы, необходимые для его осуществления, должны быть инкапсулированы в этот класс и подчинены только этой задаче.

## O

Принцип открытости/закрытости (open-closed principle)

«программные сущности ... должны быть открыты для расширения, но закрыты для модификации».

## L

Принцип подстановки Лисков (Liskov substitution principle)

«функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа не зная об этом». См. также контрактное программирование.

## I

Принцип разделения интерфейса (interface segregation principle)

«много интерфейсов, специально предназначенных для клиентов, лучше, чем один интерфейс общего назначения»

## D

Принцип инверсии зависимостей (dependency inversion principle)

«Зависимость на Абстракциях. Нет зависимости на что-то конкретное»