

if-else-if.

Операторы в Java.

Логические операторы



ПРЕПОДАВАТЕЛЬ

**Фото
преподавателя**

**Имя
Фамилия**

Текущая должность

- Количество лет опыта
- Какой у Вас опыт - ключевые кейсы
- Самые яркие проекты
- Дополнительная информация по вашему усмотрению

[Корпоративный e-mail](#)

[Социальные сети \(по желанию\)](#)



ВАЖНО:

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

ПЛАН ЗАНЯТИЯ

1. Повторение
2. Вопросы по повторению
3. Основной блок
4. Задание для закрепления
5. Задание для закрепления
6. Задание для закрепления
7. Задание для закрепления
8. Задание для закрепления
9. Вопросы по основному блоку
10. Практическая работа
11. Оставшиеся вопросы

1

ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

- Объявление метода – 6 компонентов
- Типы методов в Java
- Как назвать метод?
- Random class
- Конструктор
- this





TEL-RAN
by Starta Institute

2

ВОПРОСЫ ПО ПОВТОРЕНИЮ

Введение

- if-else
- Основные понятия
- Логические операторы
 - Оператор логического И (&&)
 - Оператор «логическое ИЛИ» (||)
 - Оператор логического НЕ(!)



3

ОСНОВНОЙ БЛОК

Принятие решений в Java

- Принятие решений в программировании похоже на принятие решений в реальной жизни.
- В программировании также встречаются ситуации, когда мы хотим, чтобы определенный блок кода выполнялся при выполнении некоторого условия.
- Язык программирования использует операторы управления для управления потоком выполнения программы на основе определенных условий.
- Они используются для того, чтобы поток выполнения продвигался и разветвлялся в зависимости от изменений состояния программы.



Операторы выбора

if

if-else

if-else-if

nested-if

instruction – return



Оператор if

Оператор if является самым простым оператором принятия решения.

Он используется для принятия решения о том, будет ли выполняться определенный оператор или блок операторов, т. е. если определенное условие истинно, то блок операторов выполняется, в противном случае нет.

Синтаксис:

If (условие) {

// Операторы для выполнения

}



TEL-RAN
by Starta Institute

4

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создайте и инициализируйте переменную `int = 10`
2. В блоке `if` сравните созданную переменную со значением 15
3. После блока добавьте инструкцию `System.out.println("I am Not in if");`
4. Проанализируйте вывод

Оператор if-else

Оператор if говорит нам, что если условие истинно, он выполнит блок операторов, а если условие ложно, то нет.

Но что, если мы хотим сделать что-то еще, если условие ложно.

Мы можем использовать оператор else с оператором if для выполнения блока кода, когда условие ложно.

Синтаксис:

```
if (условие) {
```

```
    // Выполняет этот блок, если условие  
    истинно
```

```
} else {
```

```
    // Выполняет этот блок, если условие  
    ложно
```

```
}
```



5

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создайте и инициализируйте переменную `int = 10`
2. В блоке `if` сравните созданную переменную со значением 15
3. Внутри блока добавьте инструкцию `System.out.println("i is smaller than 15");`
4. В блоке `else` добавьте инструкцию `System.out.println("i is greater than 15");`
5. Проанализируйте вывод

Оператор nested-if

Вложенный if – это оператор если, который является целью другого если или еще.

Вложенные операторы if означают оператор if внутри оператора if.

Java позволяет нам вкладывать операторы if в операторы if. т. е. мы можем поместить оператор if внутри другого оператора if.

Синтаксис:

```
if (условие1) {
```

```
    // Выполняется, когда условие1  
    истинно
```

```
    if (условие2) {
```

```
        // Выполняется, когда условие2  
        истинно
```

```
    }
```

```
}
```





TEL-RAN
by Starta Institute

6

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создайте и инициализируйте переменную `int = 10`
2. В блоке `if` сравните созданную переменную со значением 10 (`==`)
3. Внутри блока добавьте блок `if` и сравните переменную со значением 15 , внутри блока добавить инструкцию `System.out.println("i is smaller than 15");`
4. Добавьте еще один блок `if-else` и сравните переменную со значением 12
5. В блоке `if` добавить инструкцию `System.out.println("i is smaller than 12 too")`
6. В блоке `else` добавьте инструкцию `System.out.println("i is greater than 12");`
7. Проанализируйте вывод

Оператор if-else-if ladder

Здесь выбираем один из нескольких вариантов.

Операторы if выполняются сверху вниз.

Как только одно из условий, управляющих if, становится истинным, выполняется оператор, связанный с этим if, и остальная часть лестницы игнорируется.

Если ни одно из условий не выполняется, будет выполнен последний оператор else.

Синтаксис:

```
if (condition) {  
    statement;  
}  
else if (condition) {  
    statement;  
}  
else {  
    statement;  
}
```



7

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создайте и инициализируйте переменную `int = 20`
2. В блоке `if` сравните созданную переменную со значением 10 (`==`)
 - а. В блоке `if` добавить инструкцию `System.out.println("i is 10")`
3. Добавьте блок `else if` и сравните переменную со значением 15 (`==`) , внутри блока добавить инструкцию `System.out.println("i is 15");`
4. Добавьте еще один блок `else if` и сравните переменную со значением 20 (`==`)
5. Добавьте блок `else` добавьте инструкцию `System.out.println("i is not present");`
6. Проанализируйте вывод

инструкция – return

Java поддерживает оператор (инструкцию) перехода: return.

Этот оператор передает управление другой части программы.

return

Используется для явного возврата из метода - управление программой передается обратно вызывающему методу.



TEL-RAN
by Starta Institute

8

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создайте и инициализируйте переменную `boolean t = true`;
2. Добавьте инструкцию `System.out.println("Before the return instruction");`
3. В блоке `if` определите оператор `return`
4. Добавьте инструкцию после блока `if` `System.out.println("This won't execute");`
5. Запустите код
6. Проанализируйте вывод используя режим дебаггера
7. Переопределите переменную `t = false`
8. Проанализируйте вывод используя режим дебаггера

Экспресс-опрос

- **Вопрос 1.**

Объясните, как вы понимаете определение “вложенное ветвление”

- **Вопрос 2.**

При применении инструкции return, куда вернется исполнение программы?



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.



Основные понятия - операторы

Операторы классифицируются на основе предоставляемой ими функциональности.

- Арифметические операторы
- Унарные операторы
- Оператор присваивания
- Реляционные операторы
- **Логические операторы**
- Тернарный оператор
- Побитовые операторы
- Операторы сдвига



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:

- Оператор И (AND) (`&&`) → если (`cond1 && cond2`) → если true выполнить, иначе не делать
- Оператор ИЛИ (OR) (`||`) → если (`cond1 || cond2`) → если один из них true, выполнить, иначе не выполнять
- Оператор НЕ (NOT) (`!`) → `!(var1 < var2)` → false, если а меньше, чем var2

Логические операторы

- Логический оператор «И» (&&)

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

- Логический оператор «ИЛИ» (||)

Если хотя бы один из двух дает истину, оператор возвращает истину.

Чтобы результат был ложным, оба условия должны возвращать false.

- Логический оператор НЕ (!)

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



9

ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ



TEL-RAN
by Starta Institute

1

ПРАКТИЧЕСКАЯ РАБОТА

0

Практическое задание 1

1. Создайте переменные `int a, b, c, d;`
2. Используя блок `if` (выражение) `{}`, составьте логические выражения
 - a. истина , если “a” меньше “b” И “b” равно “c”;
 - b. истина , если “a” больше “b” ИЛИ “c” равно “d”;
 - c. истина , если “a” НЕ равно “b”;



Реализация задания 1

```
// initializing variables
int a = 1, b = 2, c = 2, d = 0;

// using logical AND to verify two constraints
if ((a < b) && (b == c)) {
    d = a + b + c;
    System.out.println("The sum is: " + d);
}
else
    System.out.println("False conditions");
```

Экспресс-опрос

- **Вопрос 1.**

Сколько условий будет проверено в выражении: “условие 1” || “условие 2”

- **Вопрос 2.**

Сколько логических операторов может содержать выражение?



Практическое задание 2

1. Проанализировать if блоки

```
if (x > 30 && x < 50 && x != 5) {  
    System.out.println ("number from 30 to 50, and not equal to 5");  
}
```

```
if (x < 10 || x > 40) {  
    System.out.println ("number is or less than 10 or greater than 40");  
}
```

```
if (y == 5.5 || y == 10 || y == 8.3) {  
    System.out.println ("number is or 5.5 or 10 or 8.3");  
}
```

```
if (y == 5.5 && y == 10) {  
    System.out.println ("Is there something wrong?");  
}
```



Экспресс-опрос

- **Вопрос 1.**

Может ли быть так, что в операторе if-else не выполнится ни if, ни else?

- **Вопрос 2.**

Чем отличается условие $x < 10$ от $x \leq 10$, то есть "строго меньше" от "меньше либо равно" ?





TEL-RAN
by Starta Institute

1

ОСТАВШИЕСЯ ВОПРОСЫ

1

Домашнее задание

№1

Создать программу, выводящую на экран ближайшее к 10 из двух чисел, записанных в переменные m и n .

Числа могут быть, как целочисленные, так и дробные.

Например :

ввод : $m=10.5$, $n=10.45$

вывод: Число 10.45 ближе к 10.



Домашнее задание

№2

Есть устройство, на табло которого показывается количество секунд, оставшихся до конца рабочего дня. Когда рабочий день начинается ровно в 9 часов утра — табло отображает «28800» (т.е. остаётся 8 часов), когда 14:30 — на табло «9000» (т.е. остаётся два с половиной часа), а когда наступает 17 часов — на табло отображается «0» (т.е. рабочий день закончился).

Некоторые сотрудники не умеют оценивать остаток рабочего дня в секундах.

Требуется написать программу, которая вместо секунд будет выводить на табло понятные фразы с информацией о том, сколько полных часов осталось до конца рабочего дня.

Например: «осталось 7 часов», «осталось 4 часа», «остался 1 час», «осталось менее часа».

Объяснение: в переменную n должно записываться случайное (на время тестирования программы) целое число из диапазона от 0 до 28800, далее оно должно выводиться на экран (для тех, кто понимает в секундах) и на следующей строке (для тех кто не понимает) должна выводиться фраза о количестве полных часов, содержащихся в n секундах.



Полезные ссылки

- [The if-then and if-then-else Statements \(The Java™ Tutorials > Learning the Java Language > Language Basics\) \(oracle.com\)](#)
- [Conditional \(computer programming\) - Wikipedia](#)

ЗАКЛЮЧЕНИЕ



Дополнительная практика

1. Создайте две переменные `*isWeekend*` и `*isRain*`.
2. Создайте переменную `*canWalk*`, значение которой должно быть истинным, если это выходной день (`isWeekend = true`) и не идет дождь (`isRain = false`).
3. Создайте метод, который принимает два параметра и возвращает значение `true/false`
4. Результат сохраните в `canWalk`
5. Распечатайте в консоль.