

# Switch...case, тернарный (ternary) оператор



ПРЕПОДАВАТЕЛЬ



# Артем Гордийчук

**Full-stack software engineer**

- Более 8 лет опыта работы
- Java, Spring, Hibernate, AWS, Oracle, PostgreSQL
- Проекты связанные с банковской, финансовой деятельностью, e-commerce

[artemsgor@gmail.com](mailto:artemsgor@gmail.com)

[www.linkedin.com/in/artem-g-48071a61](https://www.linkedin.com/in/artem-g-48071a61)



# 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. Вопросы по основному блоку
7. Практическая работа
8. Оставшиеся вопросы



TEL-RAN  
by Starta Institute

1

# ПОВТОРЕНИЕ ИЗУЧЕННОГО

# Повторение

- Принятие решений в Java
- if
- if-else
- nested if
- if-else-if ladder
- Instruction – return



2

# ВОПРОСЫ ПО ПОВТОРЕНИЮ

# Введение

- switch
  - switch в Java
  - Синтаксис: switch-case
  - Важные правила для операторов switch
  - Блок-схема оператора switch-Case
  - Вложенный switch
  - switch VS if else
- Тернарный (ternary) оператор
  - Определение
  - Примеры







TEL-RAN  
by Starta Institute

3

# ОСНОВНОЙ БЛОК

# switch в Java

Оператор switch является оператором многостороннего перехода.

Оператор switch выполняет один оператор из нескольких условий.

Это похоже if-else-if.

Выражение может быть примитивными типами данных byte, short, char и int.

Проверяет равенство переменных по нескольким значениям.

Синтаксис:

```
switch(expression) {  
  
    // case statements values must be of  
    same type of expression  
  
    case 1 :  
  
        // Statements  
  
        break; // break is optional  
  
    case 2 :  
  
        // Statements  
  
        break; // break is optional  
  
    default :  
  
        // Statements  
  
}
```

# switch в Java

Эволюция:

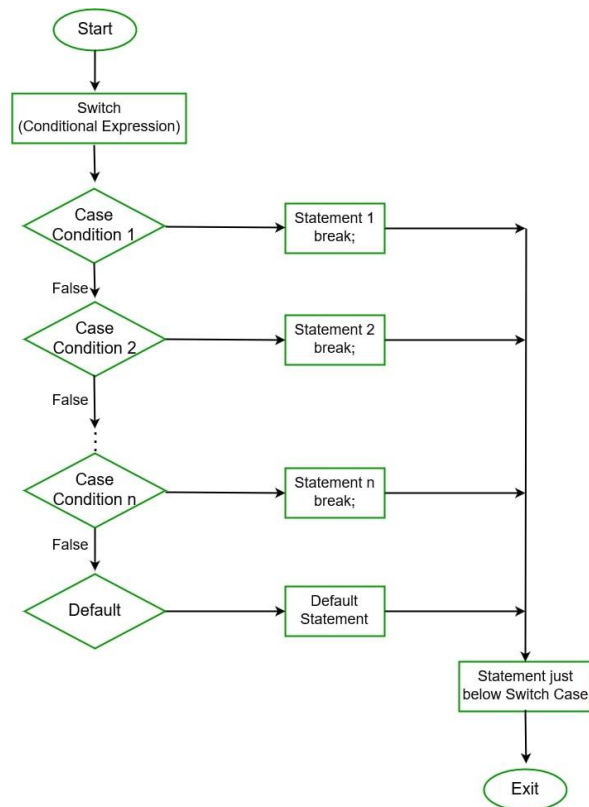
- <= JDK-6 → byte, short, char и int.  
Поддержка enum и wrappers (Character, Byte, Short, Integer)
- >= JDK-7 → String
- >= JDK -12 → switch выражения, в отличие от switch инструкций
- >= JDK – 17 → Сопоставление с образцом в switch



# Важные правила для операторов switch

- Может быть любое количество случаев, но повторяющиеся значения случаев не допускаются.
- Значение для case должно иметь тот же тип данных, что и переменная в switch.
- Значение для case должно быть постоянным. Переменные не допускаются.
- Оператор **break** используется внутри switch для завершения последовательности операторов.
- Каждый оператор case может иметь оператор break, который является необязательным. Когда управление достигает оператора break , оно переходит к элементу управления после выражения switch. Если оператор break не найден, выполняется следующий случай.
- Оператор default является необязательным и может появляться в любом месте внутри блока переключателя. В случае, если он не в конце, то после оператора default необходимо оставить оператор break, чтобы пропустить выполнение следующего оператора case

# Блок-схема оператора switch-Case



# Вложенный switch

Мы можем использовать switch как часть последовательности операторов внешнего switch.

Это называется вложенным switch.

Поскольку оператор switch определяет свой собственный блок, между константами case во внутреннем switch и во внешнем switch не возникает конфликтов.

Синтаксис:

```
switch (year) {
```

```
    case 1:
```

```
    ...
```

```
    break;
```

```
    case 2:
```

```
        // Вложенный Switch
```

```
        switch (Branch) {
```

```
            case "TelRan Berlin":
```

```
                case "TelRan Israel":
```

```
                ...
```

```
                break;
```

```
            case "TelRan USA":
```

```
                ...
```

```
                break;
```

```
            default:
```

```
                ...
```

```
        }
```

```
    }
```

# Switch VS if-else

switch	if-else
Проверяет выражения, основанные только на одном целом, перечисляемом значении или объекте	Может проверять выражения на основе диапазонов значений или условий
Оператор switch при компиляции, компилятор создает «таблицу переходов», которую он будет использовать для выбора пути выполнения в зависимости от значения выражения, потому что компилятор знает, что case-константы все одного типа	В то время как в случае if-выражений компилятор таких знаний не имеет и это может работать дольше
Операторы switch отлично подходят для фиксированных значений данных	Условные переходы if-else отлично подходят для переменных условий, которые приводят к логическому значению
Оператор switch может оказаться быстрее все элементы получают одинаковое время доступа	Для достижения последнего элемента требуется гораздо больше времени, поскольку оценивается каждое предыдущее условие
Переключатель выглядит намного чище	Спорный момент, if в некоторых случаях выглядит опрятнее



TEL-RAN  
by Starta Institute

# 4

## ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ



## ЗАДАНИЕ

1. Создайте switch-блок с днями недели
2. Создайте переменные `int day`; `String dayString`;
3. В каждом кейсе инициализируйте переменную `dayString` правильным значением.
4. Запустите программу, передав в switch-условие день 2.
5. В case №4 удалите `break`;
6. Запустите программу
7. Проанализируйте вывод

# Экспресс-опрос

- **Вопрос 1.**

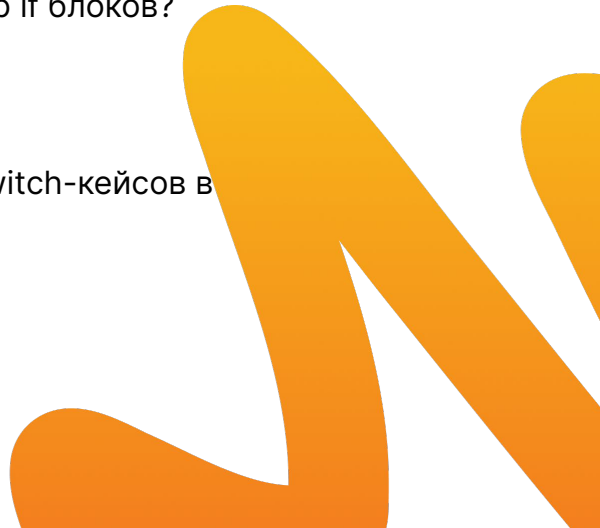
Что общего и чем отличаются инструкции `return` и `break`?

- **Вопрос 2.**

Объясните почему `switch` блок может работать эффективнее чем набор `if` блоков?

- **Вопрос 3.**

Как вы думаете какие сложности могут вызвать множество ( $>1000$ ) `switch`-кейсов в коде?



# Ternary operator

Тернарный оператор Java — единственный условный оператор, который принимает три операнда.

Это однострочная замена инструкции if-then-else.

Мы можем использовать тернарный оператор вместо условий if-else.

Условный оператор занимает меньше места и помогает писать операторы if-else кратчайшим возможным способом.

Синтаксис:

```
variable = Expression1 ? Expression2 :  
Expression3
```

```
if (Expression1) {  
    variable = Expression2;  
} else {  
    variable = Expression3;  
}
```



5

# ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Самостоятельно:

1. Создайте switch-блок, который мог бы определить, является ли переданный день выходным или рабочим днем.
2. Создайте переменные `int day`; `String dayString`;
3. Используйте объединение для нескольких случаев без операторов `break`
4. Реализуйте тот же процесс используя `if-else` блоки
5. Реализуйте тот же процесс используя тернарное выражение
6. Сравните решения

6

# ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ



TEL-RAN  
by Starta Institute

7

# ПРАКТИЧЕСКАЯ РАБОТА

# Практическое задание

Создать программу, выводящую на экран случайно сгенерированное трёхзначное натуральное число и его наибольшую цифру.

Например: 398

Выход: 9





# Реализация задания

```
public static void main(String[] args) {  
    Random rnd = new Random();  
    int a = rnd.nextInt( bound: 899) + 100;  
    int b = a % 10;  
    int c = (a / 10) % 10;  
    int d = (a / 100) % 10;  
    if (b >= c & b > d || b > c & b >= d) {  
        System.out.println("В числе " + a + " наибольшая цифра " + b);  
    } else {  
        if (c > b & c >= d) {  
            System.out.println("В числе " + a + " наибольшая цифра " + c);  
        } else {  
            System.out.println("В числе " + a + " наибольшая цифра " + d);  
        }  
    }  
}
```



TEL-RAN  
by Starta Institute

# 8

## ОСТАВШИЕСЯ ВОПРОСЫ

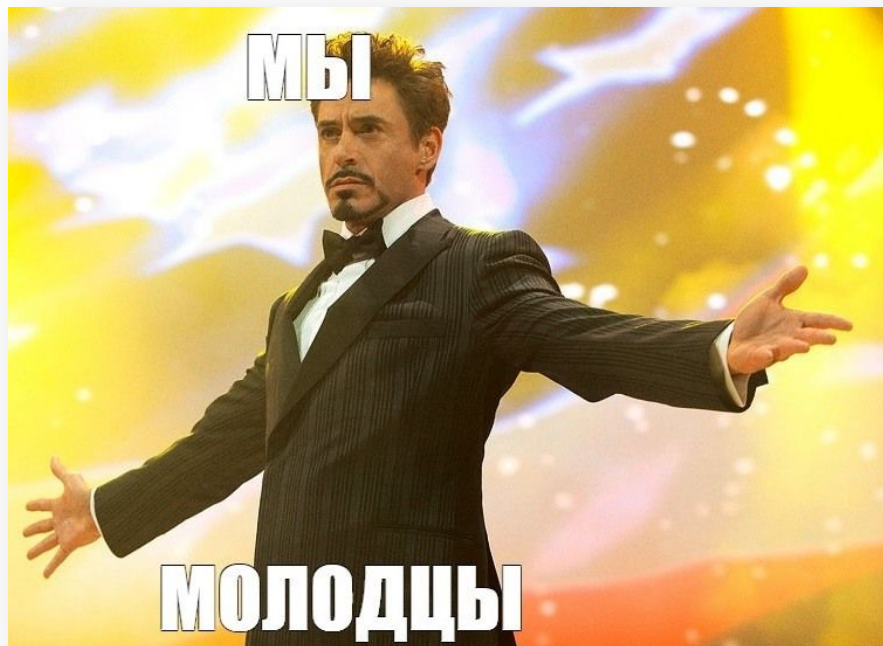
# Домашнее задание

1. Создайте две переменные `*isEdekaOpen*` и `*isReweOpen*`, значения которых зависят от того, открыты магазины или нет.
  - a. Реализует логический метод `*canBuy*`, возвращающий `boolean`
  - b. Значение этой переменной должно быть `true`, если хотя бы один магазин открыт, иначе `false`.
  - c. Отобразите строку «Я могу купить еду, это .....» и значение.
2. Реализуйте программу, которая попросит пользователя ввести год и напечатать этот год `isLeap` (високосный) или нет.
3. Реализуйте программу, которая попросит пользователя ввести три целых числа (используйте сканер) и напечатает максимум из трех чисел.

# Полезные ссылки

- [The switch Statement \(The Java™ Tutorials > Learning the Java Language > Language Basics\) \(oracle.com\)](#)
- [Switch statement - Wikipedia](#)

# ЗАКЛЮЧЕНИЕ



# Дополнительная практика

Для введённого пользователем с клавиатуры натурального числа посчитайте сумму всех его цифр (заранее не известно сколько цифр будет в числе).

Например:

Ввод = 12345

Вывод =  $1+2+3+4+5 = 15$