



STEPS TO PROGRAMMING

Урок 2-ри

Контрол на потока

Какво научихме миналият урок?

- Променливи
- Константи
- Примитивни и не-примитивни видове
- Памет
- Аритметични оператори


Инсталиране на софтуера

Standalone installation

Install IntelliJ IDEA manually to manage the location of every instance and configuration files. For example, if you have a policy that requires specific

Windows Windows (ZIP) macOS Linux

1. Download the installer ↗ .exe.


 There is a separate installer for ARM64 processors.

2. Run the installer and follow the wizard steps.

On the **Installation Options** step, you can configure the following:

On the **Installation Options** step, you can configure the following:

3. Run the installer and follow the wizard steps.

 There is a separate installer for ARM64 processors.

Минимални спецификации на компютъра:

<https://www.jetbrains.com/help/idea/installation-guide.html#requirements>

Инструкции за инсталация (Windows, Linux, macOS):

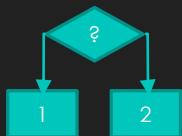
<https://www.jetbrains.com/help/idea/installation-guide.html#standalone>

Инсталирайте JetBrains IntelliJ IDEA Community Edition (не Ultimate Edition).

Контрол на потока?

- Какво е контрол на потока?
- От къде логически идва контрол на потока?
- Как използваме тези принципи в нашият живот?
- Как тези принципи се използват в програмриането?

Програма за деня



Втори урок (Контрол на потока)

- I. Булеви променливи
- II. Булеви оператори
- III. Контрол на потока с if-else statement
- IV. Контрол на потока с switch

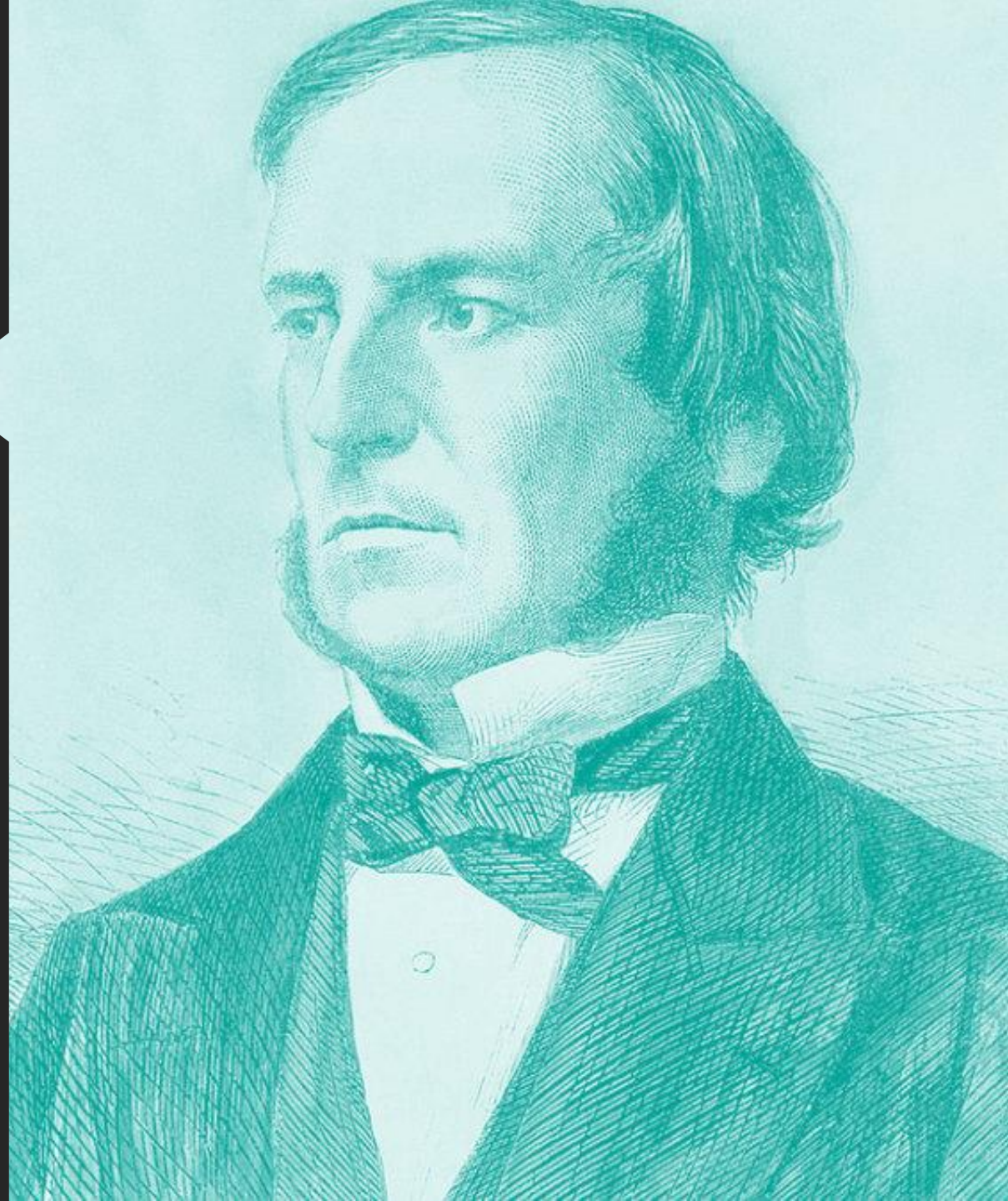
Булеви оператори

&& || ^ !

== != > < >= <=

Булева алгебра (Boolean algebra)

- George Boole създава нов 'разклон' в математиката който се казва 'Булева алгебра'.
- Различава се от елементарната алгебра по следните два основни начини:
 - Използват се логически оператори (AND, OR, XOR, NOT, equality) вместо типичните аритметични оператори (+, -, ×, ÷).
 - Операндите са 'стойности на истината' (true, false).



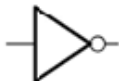
ОСНОВНИ ЛОГИЧЕСКИ ОПЕРАТОРИ

YES



| INPUT | | OUTPUT |
|-------|--|--------|
| A | | |
| 0 | | 0 |
| 1 | | 1 |

NOT



| INPUT | | OUTPUT |
|-------|--|--------|
| A | | |
| 0 | | 1 |
| 1 | | 0 |

AND



| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |

OR



| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

XOR



| INPUT | | OUTPUT |
|-------|---|--------|
| A | B | |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

Source

Пример (AND, NAND, OR)



```
boolean cheesecakeIsNice = true;  
boolean cheesecakeIsHealthy = false;  
  
cheesecakeIsNice && cheesecakeIsHealthy // false  
  
!(cheesecakeIsNice && cheesecakeIsHealthy) // true - YAY!  
  
cheesecakeIsNice || cheesecakeIsHealthy // true
```

Пример (OR vs XOR)

```
boolean haveCake = true;  
boolean canEatCake = true;
```

```
haveCake || canEatCake // true - YAY!
```

```
haveCake ^ canEatCake // false :(
```

| OR | | | XOR | | |
|---|---|--------|---|---|--------|
|  | | |  | | |
| INPUT | | OUTPUT | INPUT | | OUTPUT |
| A | B | | A | B | |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

Оператори за сравнение на равенствост (Equality operators)

| A | Оператор | B | Отговор |
|----|----------|---|---------|
| 5 | == | 5 | true |
| 10 | == | 3 | false |
| 5 | != | 5 | false |
| 10 | != | 3 | true |

Оператори за сравнение на стойност (Comparison operators)

| A | Оператор | B | Отговор |
|----|----------|----|---------|
| 5 | > | 5 | false |
| 10 | > | 3 | true |
| 3 | > | 10 | false |
| 5 | < | 5 | false |
| 10 | < | 3 | false |
| 3 | < | 10 | true |
| 5 | >= | 5 | true |
| 10 | >= | 3 | true |
| 3 | >= | 10 | false |
| 5 | <= | 5 | true |
| 10 | <= | 3 | false |
| 3 | <= | 10 | true |

if statement

```
boolean iSeeTreesOfGreen = true;  
boolean redRosesToo = true;  
boolean theyBloomForMeAndYou = true;
```

```
if (iSeeTreesOfGreen && redRosesToo && theyBloomForMeAndYou)  
{  
    System.out.println("And I think to myself");  
    System.out.println("What a wonderful world");  
    // Louis Armstrong - What a Wonderful World  
}
```

if-else statement

```
int age = 15;  
int movieAgeGuidelines = 12;  
  
if (age < movieAgeGuidelines) {  
    System.out.println("Sorry, you're too young to watch this movie.");  
} else {  
    System.out.println("Here are your tickets! Enjoy the movie!");  
}
```

if-else if statement

```
int guess = 15;
int number = 55;

System.out.println("I thought of a number - see if you can guess it :");
if (guess < number) {
    System.out.println("Sorry, your guess is too low. Try again");
} else if (guess > number) {
    System.out.println("Sorry, your guess is too high. Try again.");
} else if (guess == number) {
    System.out.println("You guessed correctly! Well done! *clapping*");
}
```



Демо: Булева логика

Switch statement

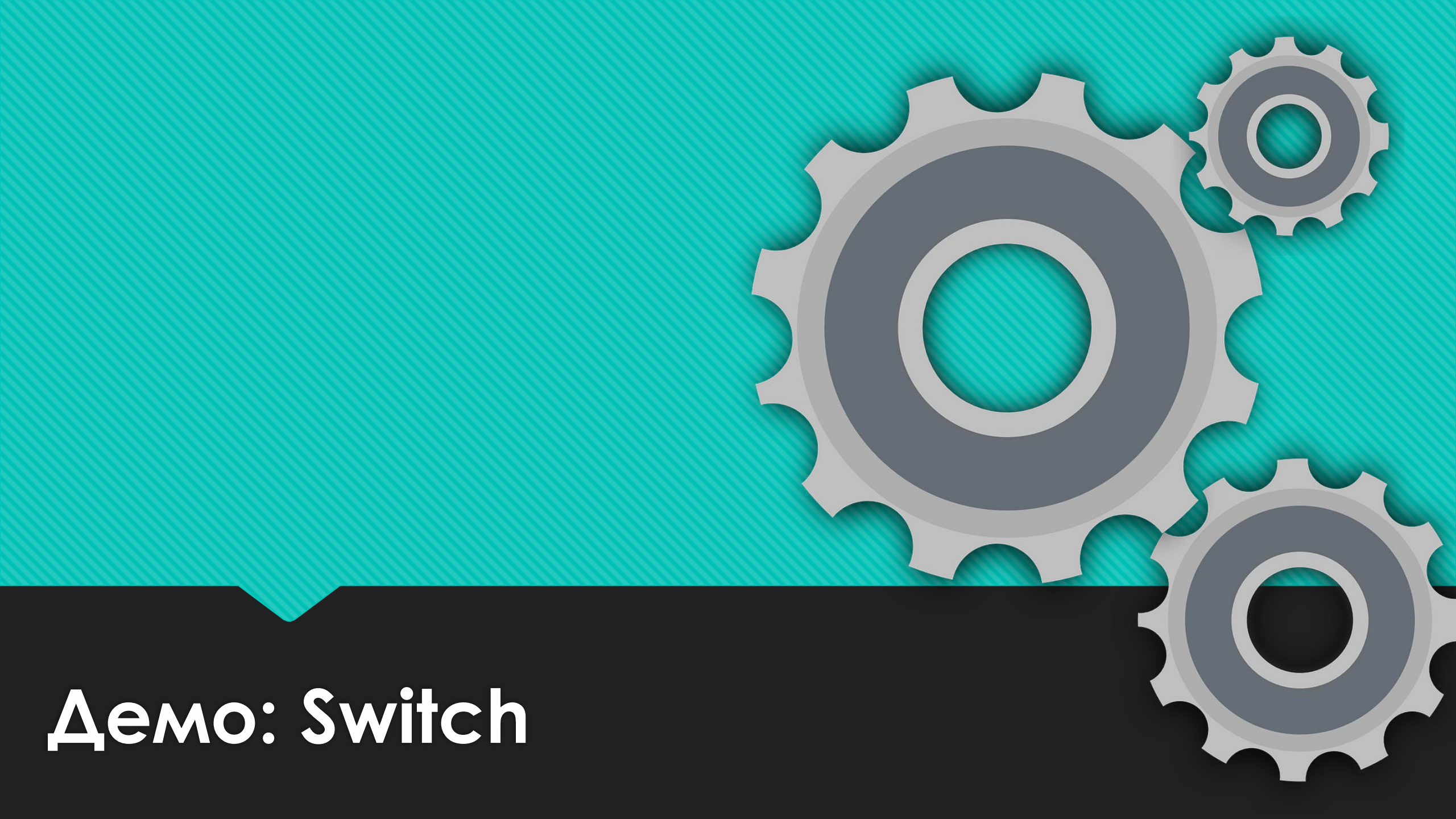
- Какво е switch statement?

```
switch(month) {  
    case 1:  
        System.out.println("We are in January");  
    case 2:  
        System.out.println("We are in February");  
    case 3:  
        System.out.println("We are in March");  
}
```

- Как се различава от If?
- Кога да използваме If и кога switch?

Switch statement

- Какви са лимитите на switch?
- Други алтернативи на if or switch
- Ternary оператор
- `bool ternary = 15 > 10 ? true : false`



Демо: Switch

Обобщение

- Булеви Оператори
- Какво е контрол на потока
- If-else
- Switch

Въпроси и Отговори

Тази презентация, ресурси и код:
<https://github.com/stanivanov92/steps-to-programming>