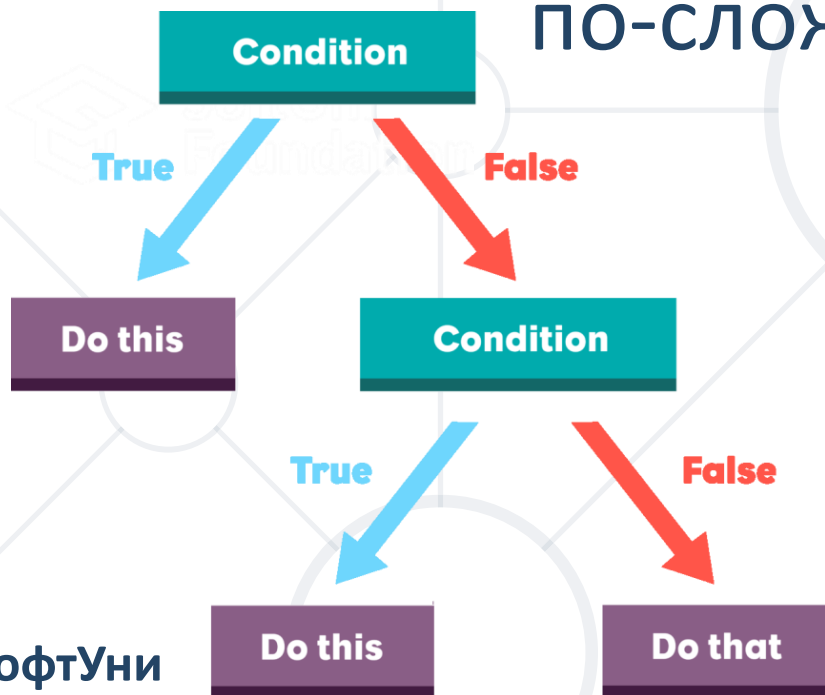


По-сложни проверки

Вложени If конструкции и
по-сложни логически условия



СофтУни
Преподавателски екип



SoftUni



<https://softuni.bg>

Софтуерен университет

1. Преговор
2. Условна конструкция **switch - case**
 - Множество случаи в switch – case
3. Вложени условни конструкции
4. Логически оператори
 - Логически оператори "&&", "||", "!"
 - Приоритет на условия





Преговор

1. Коя променлива е наименувана правилно?

`savedMoney`

`SavedMoney`

`спестениПари`

`spesteniPari4ki`

2. Каква стойност ще присвои променливата "isGreater":

```
boolean isGreater = (5 + 3) > (3 + 4);
```

True

False

8

15

3. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

```
if ("caseSensitive".equals("CaseSensitive")) {  
    System.out.println("Correct!");  
} else {  
    System.out.println("Not correct!");  
}
```

Error

No output

Correct!

Not
correct!

4. Какъв ще е резултатът от изпълнението на следната програма:

```
System.out.println(123456 % 100 == 56);
```

True

False

56

Error

5. Какво ще се отпечата на конзолата, ако изпълним следната логическа проверка:

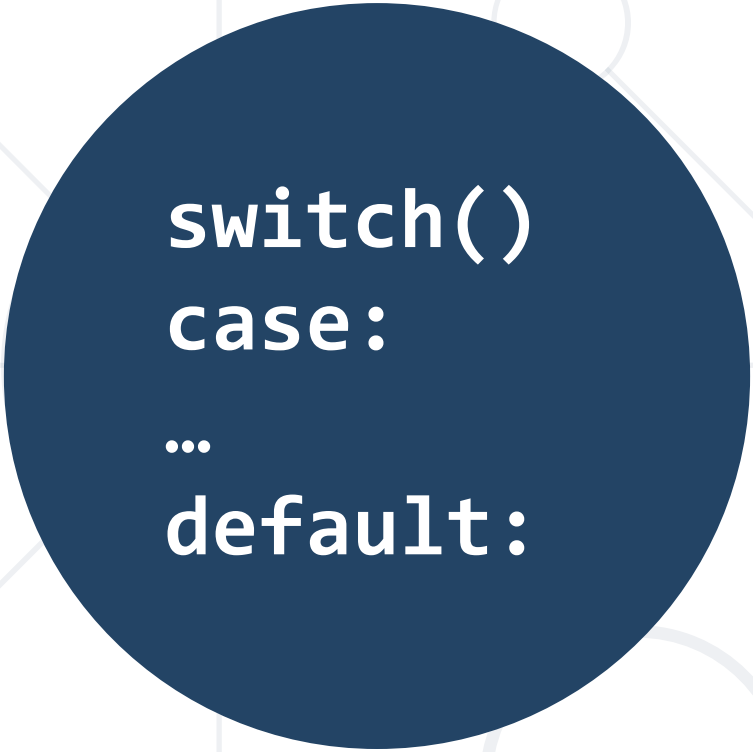
```
String role = "Administrator";  
if (role.equals("Administrator")) {  
    System.out.println("Welcome");  
} else {  
    System.out.println("No permission");  
}
```

No permission

Welcome

Compile
time error

No output



```
switch()  
case:  
...  
default:
```

Условна конструкция Switch-case

Условна конструкция Switch-case

- Работи като поредица `if/else if/else if...`

```
switch (...)  
{
```

Променлива
(текст или цяло число)

Изброяване на
условия (стойности)
за проверката

```
case ... :  
    // code  
    break;
```

```
case ... :  
    // code  
    break;
```

```
default:  
    // code  
    break;
```

Код, който ще се изпълни,
ако няма съвпадение с
нито един случай

```
}
```

- Напишете програма, която:
 - Чете цяло число, въведено от потребителя
 - Отпечатва на конзолата деня от седмицата с текст (на английски) според въведеното число [1...7]
 - Отпечатва на конзолата "**Error**", ако числото не е в диапазона
- Примерен вход и изход:

1 → Monday

4 → Thursday

Тестване на решението: <https://judge.softuni.bg/Contests/2391>

```
int day = Integer.parseInt(scanner.nextLine());
switch (day) {
    case 1:
        System.out.println("Monday"); break;
    case 2:
        System.out.println("Tuesday"); break;
    //TODO: check the other days
    case 7:
        System.out.println("Sunday"); break;
    default:
        System.out.println("Error"); break;
}
```

- Чрез **switch-case**, можем да изпълняваме един и същ код за множество условия

```
switch (...)  
{  
    case ... :  
    case ... :  
    case ... :  
        // code  
        break;  
    default:  
        // code  
        break;  
}
```

Кодът ще се изпълни
ако някое от трите
условия в серията е
вярно

- Напишете програма, която:
 - Четен ден от седмицата (**текст**) - въведен от потребителя
 - Ако денят е работен - отпечатва на конзолата "**Working day**"
 - Ако денят е почивен - отпечатва на конзолата "**Weekend**"
 - Ако се въведе текст различен от ден от седмицата - отпечатва на конзолата "**Error**"
- Примерен вход и изход:

Monday



Working day

Sunday



Weekend

Почивен или работен ден - решение

```
switch (day)
{
    case "Monday":
    case "Tuesday":
        // TODO
        System.out.println("Working day");
        break;
    case "Saturday":
    case "Sunday":
        System.out.println("Weekend");
        break;
    default:
        System.out.println("Error");
        break;
}
```

- Напишете програма, която:
 - Проверява дали въведеният вход от потребителя е плод или зеленчук измежду изброените:
 - Плодове: banana, apple, kiwi, cherry, lemon, grapes
 - Зеленчуци: tomato, cucumber, pepper, carrot
 - Извежда "**vegetable**", "**fruit**" или "**unknown**"
- Примерен вход и изход:

lemon → fruit carrot → vegetable java → unknown

Плод или зеленчук – решение

```
switch (food)
{
    case "banana":
        // TODO for all the fruits
    case "grapes":
        System.out.println("fruit");
        break;
    case "tomato"
        // TODO for all the vegetables
    case "carrot":
        System.out.println("vegetable");
        break;
    default:
        System.out.println("unknown");
        break;
}
```



```
{ if else }
```

Вложени условни конструкции

- Само при изпълнение на първото условие се преминава към вложената проверка

```
if (condition1) {  
    System.out.println("condition1 valid");  
    if (condition2) {  
        System.out.println("condition2 valid");  
    } else {  
        System.out.println("condition2 not valid");  
    }  
}
```

Вложена if конструкция

- Напишете програма, която чете от потребителя:
 - Възраст и пол
 - Принтира обръщение според въведените данни, както е показано на схемата (в следващия слайд)
- Примерен вход и изход:

12
f



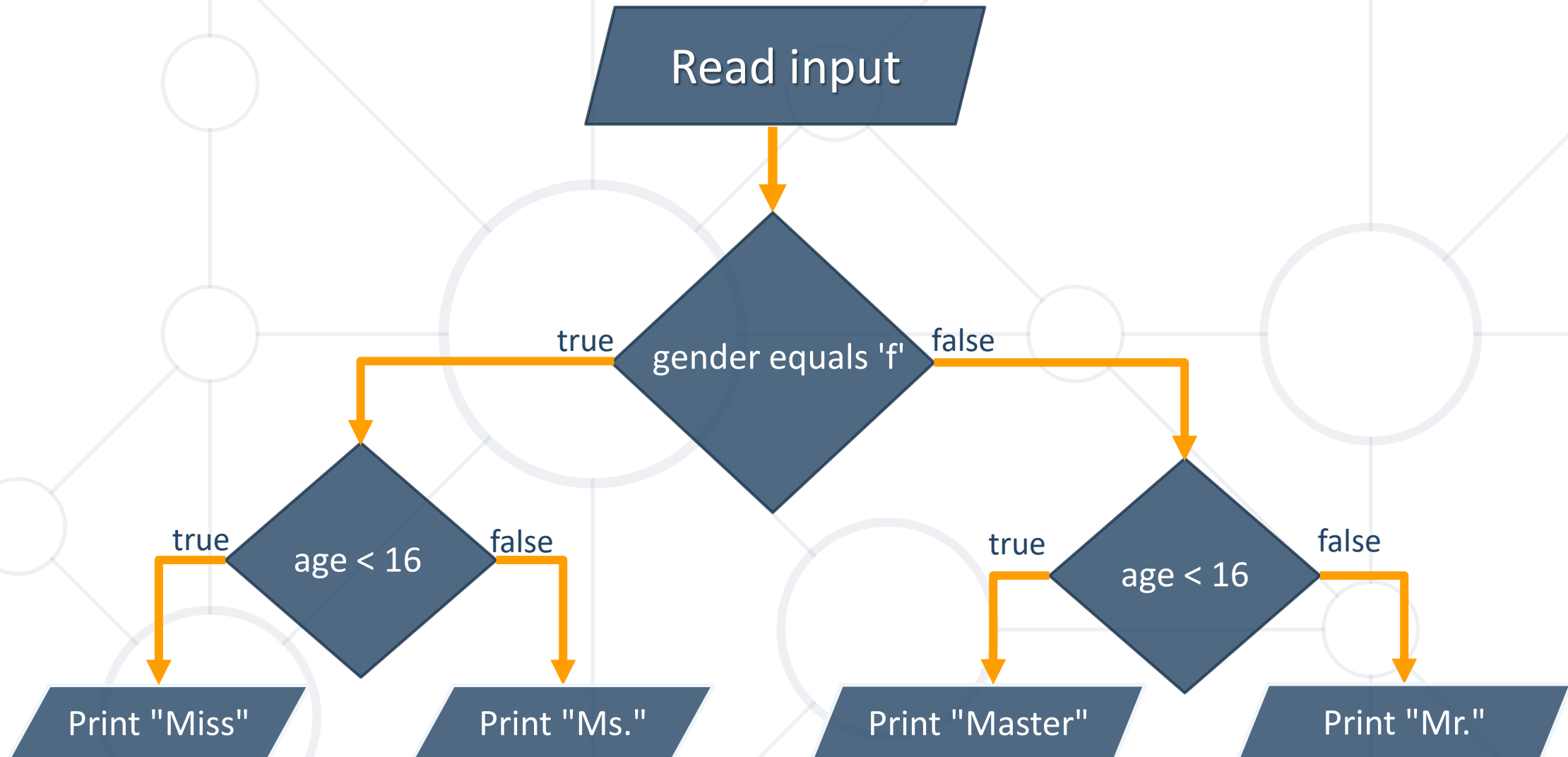
Miss

16
m



Mr.





Тестване на решението: <https://judge.softuni.bg/Contests/2391>

```
if(gender.equals("f")) {  
    if(age >= 16){  
        System.out.println("Ms.");  
    }  
    else {  
        System.out.println("Miss");  
    }  
}  
else {  
    //TODO: check others titles - "Mr.", "Master"  
}
```

Квартално магазинче – условие

- Напишете програма, която чете от потребителя:

- Име на продукт
- Град
- Количество

- Пресмята цената му спрямо таблицата:



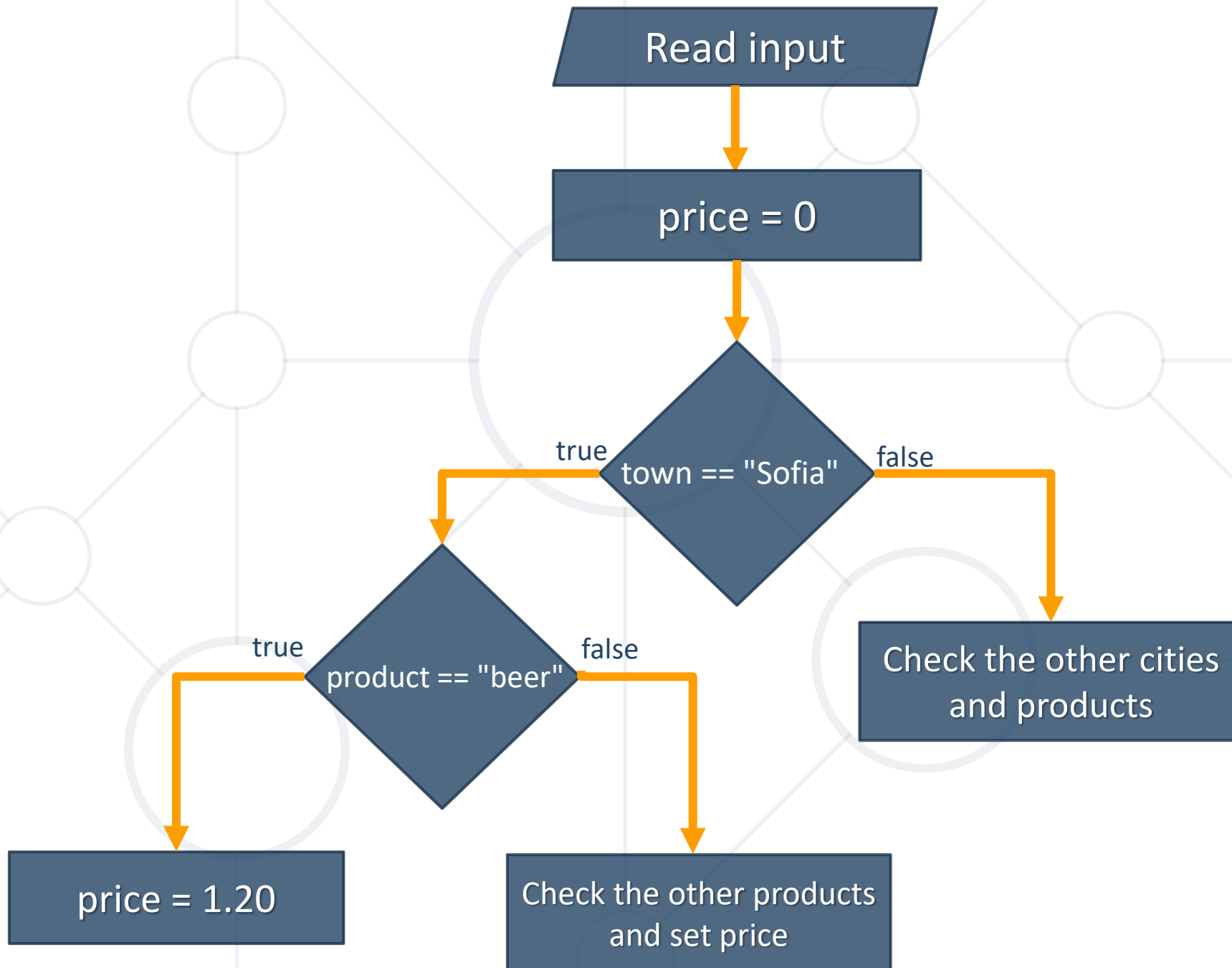
Град/продукт	coffee	water	beer	sweets	peanuts
Sofia	0.50	0.80	1.20	1.45	1.60
Plovdiv	0.40	0.70	1.15	1.30	1.50
Varna	0.45	0.70	1.10	1.35	1.55

Квартално магазинче – условие (2)


- Примерен вход и изход:



Тестване на решението: <https://judge.softuni.bg/Contests/2391>



```
String product = scanner.nextLine().toLowerCase();
String town = scanner.nextLine().toLowerCase();
double quantity = Double.parseDouble(scanner.nextLine());
if (town.equals("sofia")) {
    if (product.equals("coffee")) {
        System.out.println(0.50 * quantity);
    }
    // TODO: Check the other cases...
}
else if (town.equals("varna")) //TODO: Add Logic here...
else if (town.equals("plovdiv")) //TODO: Add Logic here...
```

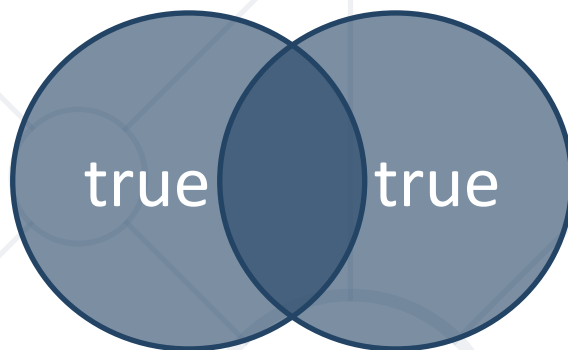
A background network diagram consisting of a grid of light gray lines intersecting at various points. Some of these intersections are marked with small, empty light gray circles. A larger, solid dark blue circle is centered in the upper half of the image, containing white text. Below this circle, the title 'Логически оператори' is written in a bold, dark blue font.

```
if ()  
else if()  
...  
else
```

Логически оператори

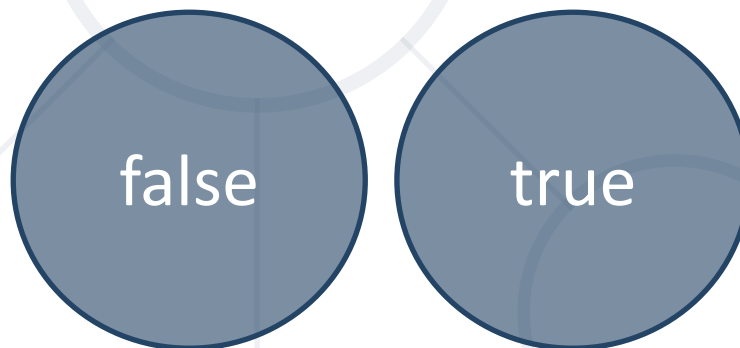
- Оператори, които комбинират или изключват условия
- Връщат булев резултат (true или false)

"&&" - И



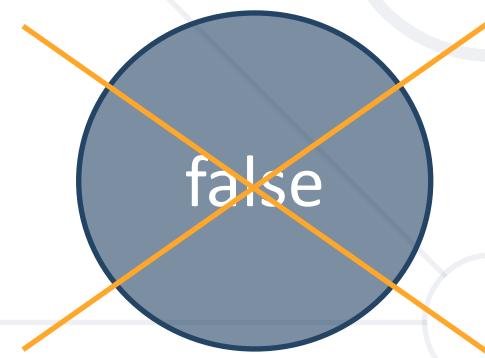
Вярност на двете условия

"||" - ИЛИ



Вярност на
едното или на другото
условие

"!" - ОТРИЦАНИЕ



Отрицание на условие

- Проверява изпълнението на няколко условия едновременно
- Пример: проверка дали число е едновременно:
 - по-голямо от 5 и по-малко от 10
 - четно

&&

```
Scanner scanner = new Scanner(System.in);  
int a = Integer.parseInt(scanner.nextLine());  
if (a > 5 && a < 10 && a % 2 == 0) ...
```

- Вложенные проверки

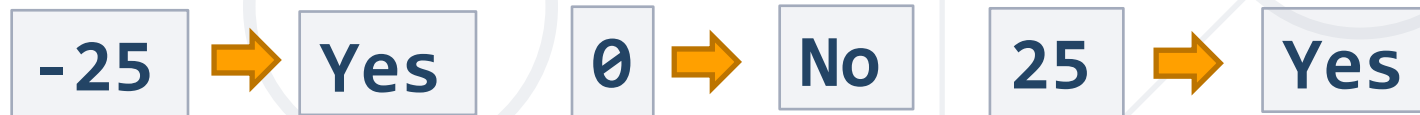
```
if (a > 5) {  
    if (a < 10) {  
        if (a % 2 == 0) {  
            }  
        }  
    }  
}
```

- Логический оператор **&&**

```
if (a > 5 && a < 10  
    && a % 2 == 0) {  
}
```



- Напишете програма, която:
 - Проверява дали въведеното число от потребителя е в интервала $[-100, 100]$ и е различно от 0
 - Извежда "Yes", ако е в интервала и различно от 0, или "No" ако е извън тях.
- Примерен вход и изход:



Число в интервала – решение

```
Scanner scanner = new Scanner(System.in);  
int number = Integer.parseInt(scanner.nextLine());  
  
if (number >= -100 && number <= 100 && number != 0) {  
    System.out.println("Yes");  
} else {  
    System.out.println("No");  
}
```

Тестване на решението: <https://judge.softuni.bg/Contests/2391>

- Проверява дали е изпълнено поне едно измежду няколко условия
- Пример: проверка дали въведената дума е:
 - "Example" или "Demo"

```
Scanner scanner = new Scanner(System.in);  
String input = scanner.nextLine();  
if (input.equals("Example") || input.equals("Demo")) ...
```

- Вложенные проверки

```
if(word.equals("Example")){  
}  
else if(word.equals("Demo")){  
}
```

- Логический оператор ||

```
if (word.equals("Example")  
||  
    word.equals("Demo")){  
}
```



- Напишете програма, която:
 - Чете ден от седмицата (**текст**) – въведен от потребителя
 - Отпечатва цената на билет за кино според деня от седмицата

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
12	12	14	14	12	16	16

- Примерен вход и изход:

Monday → 12

Sunday → 16

```
Scanner scanner = new Scanner(System.in);
String day = scanner.nextLine();
if (day.equals("Monday")
    || day.equals("Tuesday")
    || day.equals("Friday")){
    System.out.println(12);
} else if (day.equals("Wednesday")
    || day.equals("Thursday")) {
    System.out.println(14);
}
// TODO for Saturday and Sunday
```

- Проверява дали **не е изпълнено** дадено условие
- Пример:
 - Проверка дали число е по-голямо от 10 и е четно:

```
Scanner scanner = new Scanner(System.in);  
int number = Integer.parseInt(scanner.nextLine());  
boolean isValid = (number > 10) && (number % 2 == 0);  
if (!isValid) {  
    System.out.println("Invalid");  
}
```



- Напишете програма, която:
 - Чете цяло число - въведено от потребителя
 - Числото е валидно ако е в интервала [100...200] или е 0
 - Ако числото е невалидно да се отпечата на конзолата "invalid", в противен случай да не се отпечатва нищо
- Примерен вход и изход:

75 → invalid

150 → (няма изход)

```
Scanner scanner = new Scanner(System.in);  
int n = Integer.parseInt(scanner.nextLine());  
boolean isValid = n >= 100 && n <= 200 || n == 0;  
if (!isValid) {  
    System.out.println("invalid");  
}
```

- Чрез скоби **()** можем да приоритизираме условия

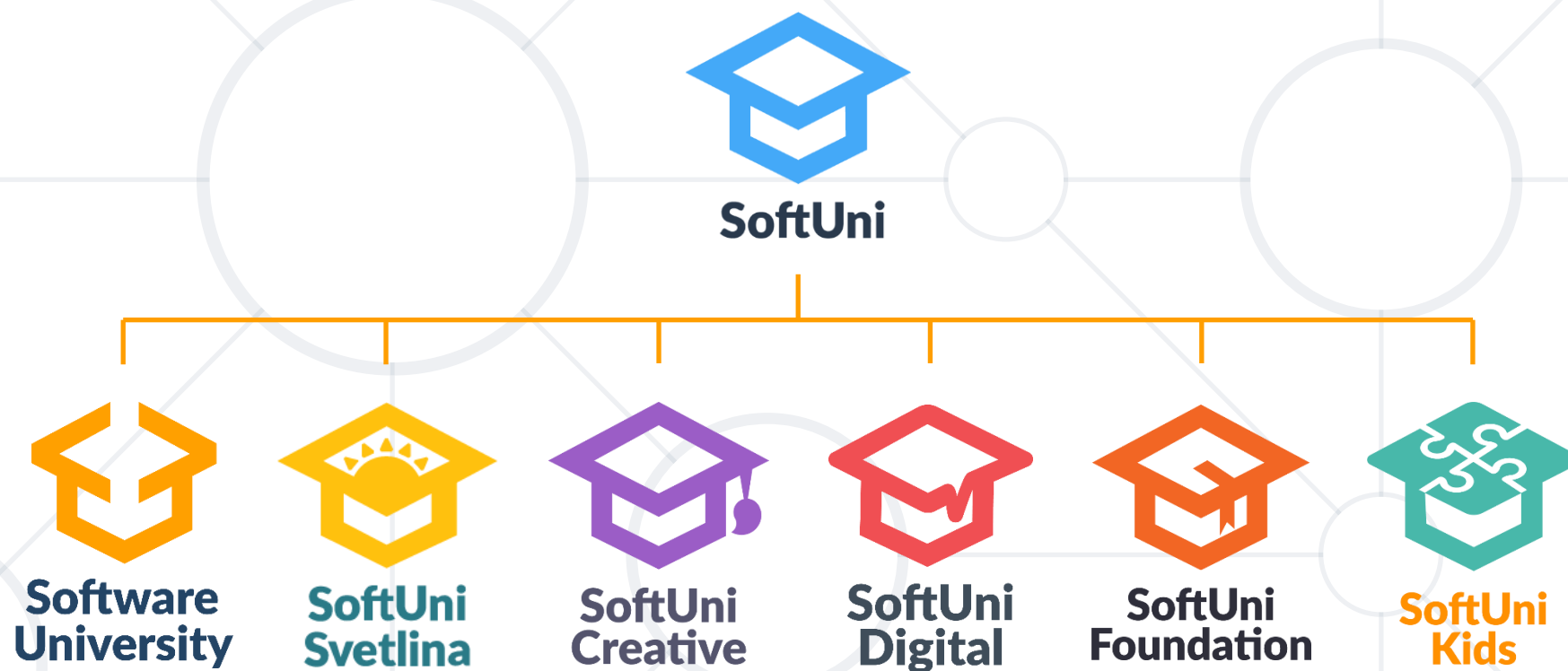
```
int a = 50;
int b = 200;
int c = 300;
if ((a >= 100 && b <= 200) || (c + b >= 300 && c <= 400))
{
    System.out.println ("Yes"); // Yes
}

if (a >= 100 && (b <= 200 || c + b >= 300) && c <= 400)
{
    System.out.println ("Yes"); // No output
}
```


- Условна конструкция - **switch-case**
- Вложени условни конструкции
- Логически оператори - **&&, ||, !**
 - Приоритет на условия – **()**



Въпроси?



- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – <https://softuni.org>
- © Софтуерен университет – <https://softuni.bg>



- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
 - softuni.bg
- Фондация "Софтуерен университет"
 - softuni.foundation
- Софтуерен университет @ Facebook
 - facebook.com/SoftwareUniversity
- Дискуссионни форуми на СофтУни
 - forum.softuni.bg



Software University

