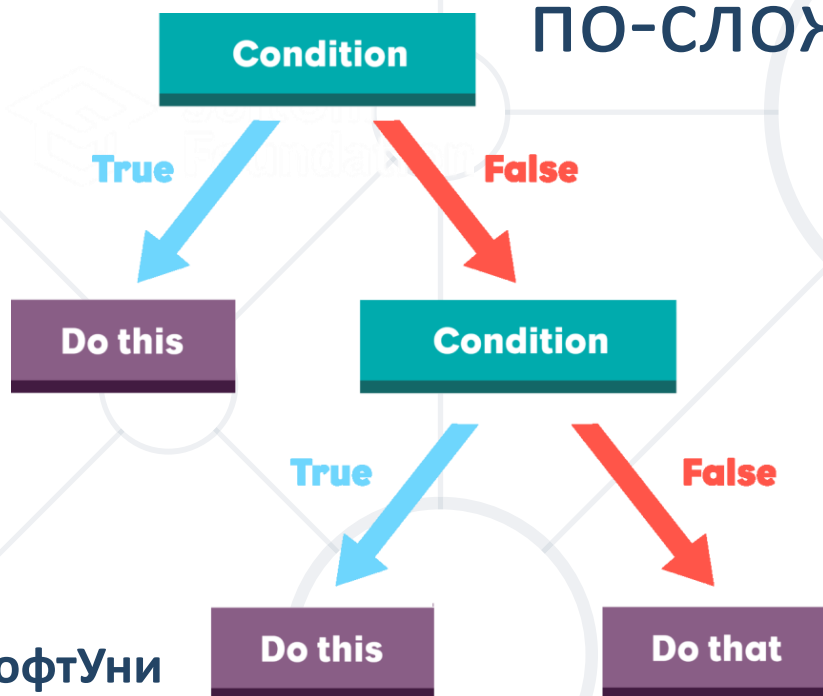


По-сложни проверки

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?

Отговор заради CamelCase

savedMoney

pascal case

SavedMoney

спестениПари

spesteniPari4ki

2. Каква стойност ще присвои променливата "isGreater":

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

Отговор

True

False

8

15

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

```
if ("caseSensitive" == "CaseSensitive")  
{  
    Console.WriteLine("Svetlin");  
}  
else  
{  
    Console.WriteLine("Petar");  
}  
// прави разлика между големи и малки букви
```

Svetlin

Error

No output

Отговор

Petar

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

```
Console.WriteLine(123456 % 100 == 56); //(резултат е от 123456/100 =56)
```

Отговора, защото
имаме логически
оператор

True

False

56

Error

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

```
string role = "Administrator";  
if (role != "Administrator")  
{  
    Console.WriteLine("No permission");  
}  
else  
{  
    Console.WriteLine("Welcome");  
} //пита се дали са различни, не те са еднакви и връща false
```

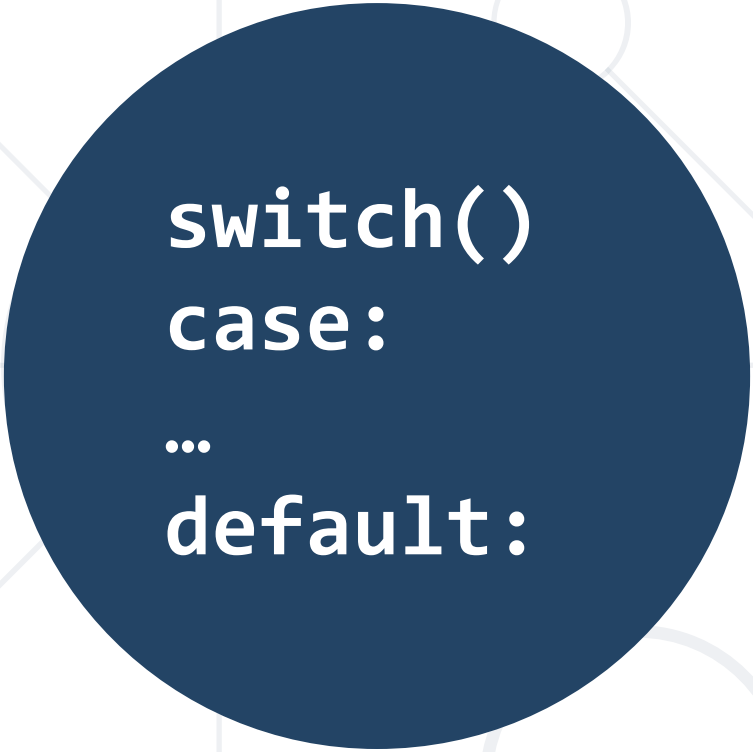
Отговора

"Welcome"

No permission

No output

Compile
time error

A background network diagram consisting of a grid of light gray lines intersecting at various points. At these intersections, there are several light gray circles of different sizes. A large, solid dark blue circle is positioned in the upper-middle part of the image, containing white text. Below this circle, the title 'Условна конструкция Switch-case' is written in a bold, dark blue font.

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

Условна конструкция Switch-case

Условна конструкция Switch-case

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

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

Условието в
switch case
е стойност

Изброяване на
условия (стойности)
за проверката

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

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

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

Код, който ще се изпълни,
ако няма съвпадение с
нико 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

5 → Friday

2 → Tuesday

```
int day = int.Parse(Console.ReadLine());
switch (day)
{
    case 1:
        Console.WriteLine("Monday"); break;
    case 2:
        Console.WriteLine("Tuesday"); break;
    // TODO: check the other days
    case 7:
        Console.WriteLine("Sunday"); break;
    default:
        Console.WriteLine("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:
 - Чете 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:



Почивен или работен ден - решение

```
switch (day)
{
    case "Monday":
    case "Tuesday":
        // TODO
        Console.WriteLine("Working day");
        break;
    case "Saturday":
    case "Sunday":
        Console.WriteLine("Weekend");
        break;
    default:
        Console.WriteLine("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
-------	---	-------	--------	---	-----------	------	---	---------


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



```
{ if else }
```

Вложени условни конструкции

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

```
if (condition1)
{
    Console.WriteLine("condition1 valid");
    if (condition2)
        Console.WriteLine("condition2 valid");
    else
        Console.WriteLine("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



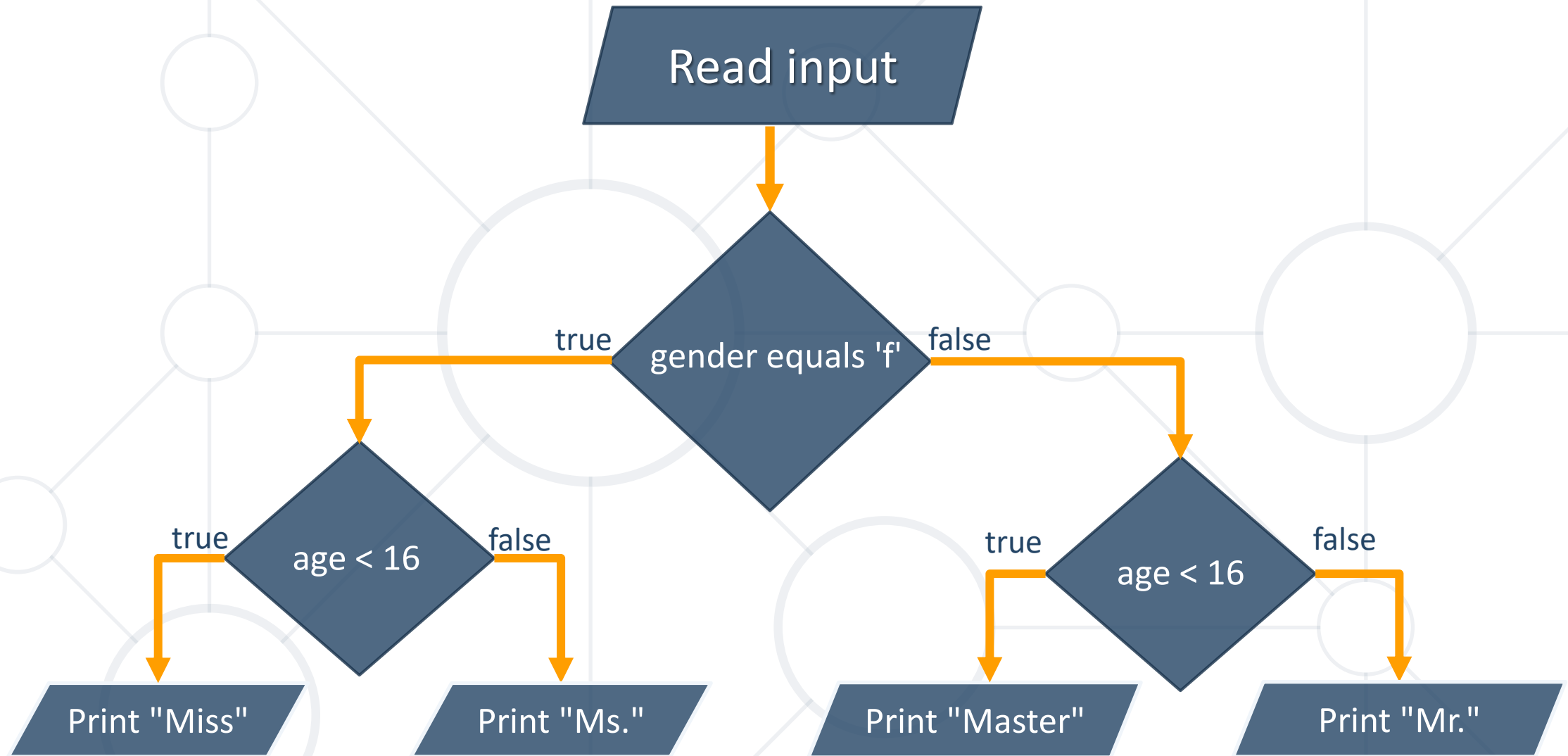
Miss

16
m



Mr.





Обръщение според възраст и пол - решение

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

- Напишете програма, която чете от потребителя:

- Име на продукт
- Град
- Количество

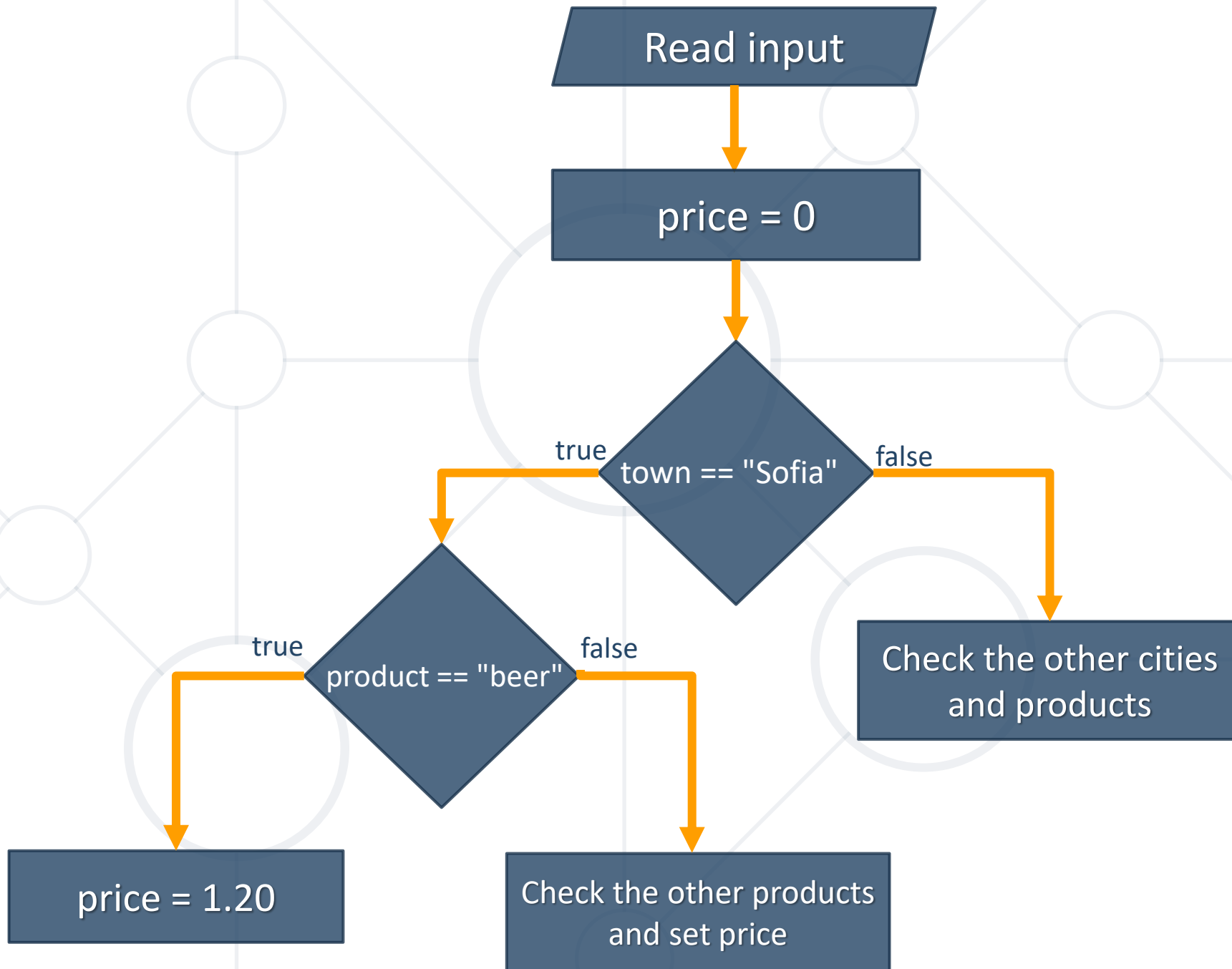
- Пресмята цената му спрямо таблицата:



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

- Примерен вход и изход:





```
string productName = Console.ReadLine();
string town = Console.ReadLine();
int quantity = int.Parse(Console.ReadLine());
double price = 0;
if (town == "Sofia")
{
    if (productName == "coffee") price = quantity * 0.50;
    // TODO: finish the checks for all the products...
}
else if (town == "Plovdiv")
else if (town == "Varna")
// TODO: check other two towns...
```

A background network diagram consisting of a grid of light gray lines intersecting at various points. At these intersections, there are several circles of different sizes, some solid light gray and some hollow. The overall pattern suggests a complex network or data structure.

&&

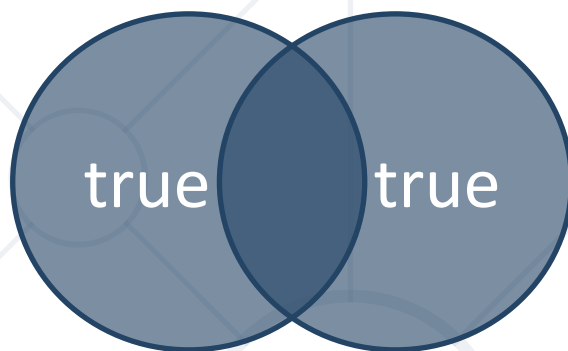
||

!

Логически оператори

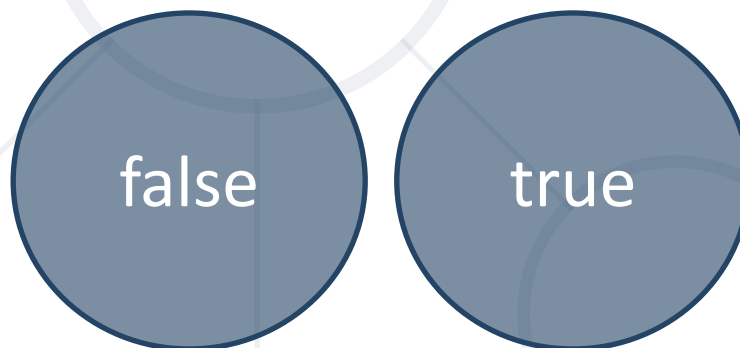
- Оператори, които комбинират или изключват условия
- Връщат булев резултат (true или false)

"&&" - И



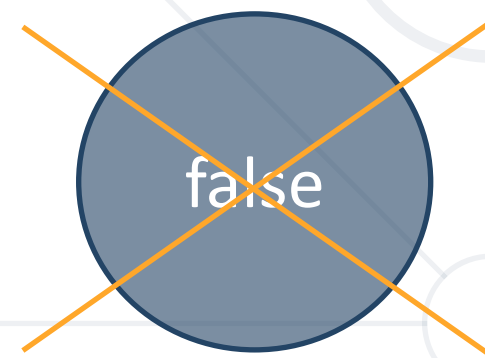
Вярност на двете условия

"||" - 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

&&

```
int a = int.Parse(Console.ReadLine());  
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

```
int number = int.Parse(Console.ReadLine());

if (number >= -100 && number <= 100 && number != 0)
{
    Console.WriteLine("Yes");
}
else
{
    Console.WriteLine("No");
}
```


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

```
string word = Console.ReadLine();  
if (word == "Example" || word == "Demo") ...
```



- Вложенные проверки

```
if (word == "Example"){  
}  
else if (word == "Demo"){  
}
```

- Логический оператор **||**

```
if (word == "Example" ||  
    word == "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

```
string day = Console.ReadLine();
if(day == "Monday" || day == "Tuesday" || day == "Friday")
{
    Console.WriteLine(12);
}
else if (day == "Wednesday" || day == "Thursday")
{
    Console.WriteLine(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:

```
int number = int.Parse(Console.ReadLine());  
bool isValid = (number > 10) && (number % 2 == 0);  
if (!isValid)  
{  
    Console.WriteLine("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 отпечата нищо
- Примерен вход и изход:

75 → invalid

150 → (няма изход)

Невалидно число - решение

```
int number = int.Parse(Console.ReadLine());  
bool isValid = number >= 100 && number <= 200 || number == 0;  
if (!isValid)  
{  
    Console.WriteLine("invalid");  
}
```

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

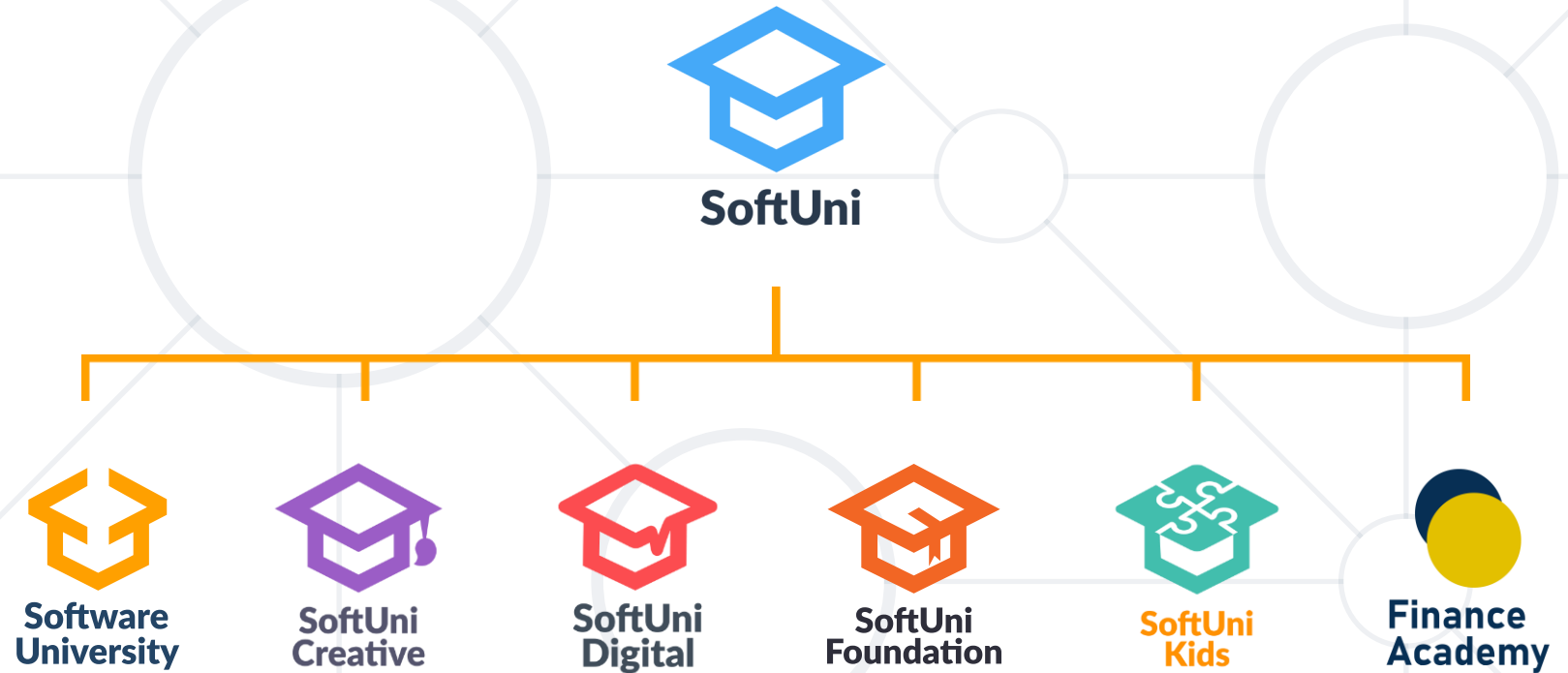
```
int a = 50;
int b = 200;
int c = 300;
if ((a >= 100 && b <= 200) || (c + b >= 300 && c <= 400))
{
    Console.WriteLine("Yes"); // Yes
}

if (a >= 100 && (b <= 200 || c + b >= 300) && c <= 400)
{
    Console.WriteLine("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

