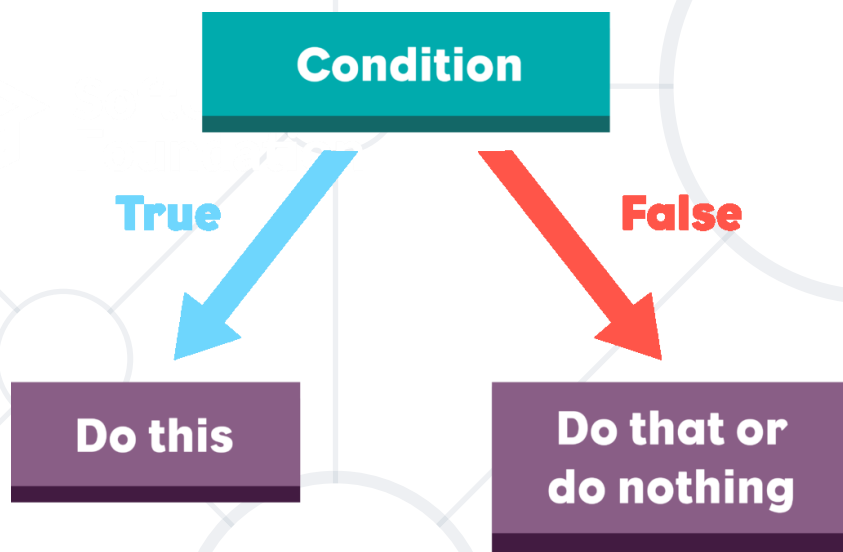


# Проверки

Логически изрази и проверки. Условна конструкция

if-else



СофтУни

Преподавателски екип



SoftUni



Software University

<https://softuni.bg>

1. Преговор
2. Логически изрази и проверки
  - Оператори за сравнение
3. Условни конструкции
4. Закръгляне и форматиране
5. Дебъгване
6. Серия от проверки
7. Живот на променлива





**Преговор**

1. Какъв е типът на променливата:

```
... number = "1000";
```

char

int

Отговор string

double

2. Какъв е типът на променливата:

```
... number = 1000;
```

Отговор  
**int**

**string**

**char**

**double**

## 3. Как се нарича долепването на два текста (низа)?

Събиране

Отговора

Конкатенация

Стрингосване

Съединяване

4. Какво ще се отпечата на конзолата, ако изпълним следната команда:

```
Console.WriteLine(10 % 3);
```

10

Отговор **1**

0

3

5. Каква стойност държи променливата **result**:

```
int a = 5;  
int b = 2;  
double result = a / b;
```

2.5

7

Отговор  
2.0

1



6. Какъв би бил резултатът, ако се опитаме да изпълним следната команда:

```
Console.WriteLine(1 + 1 + "4" + 2 + 1);
```

Compile  
time error

9

243

Отговор 2421



# Логически изрази и проверки

Оператори за сравнение

# Оператори за сравнение



Оператор	Означение	Работи за
Равенство	==	числа, дати, други сравними типове
Различно	!=	
По-голямо	>	
По-голямо или равно	>=	
По-малко	<	
По-малко или равно	<=	

- В програмирането можем да сравняваме стойности
  - Резултатът от логическите изрази е **true** или **false**

```
int a = 5;  
int b = 10;  
Console.WriteLine(a < b);  
Console.WriteLine(a > 0);  
Console.WriteLine(a > 100);  
Console.WriteLine(a < a);  
Console.WriteLine(a <= 5);  
Console.WriteLine(b == 2 * a);
```

*// true*

*// true*

*// false*

*// false*

*// true*

*// true*



- Сравняване на текст чрез оператор за равенство (==)

```
string a = "Example";  
string b = a;  
Console.WriteLine(a == b); // true
```

```
string a = Console.ReadLine();  
string b = Console.ReadLine();  
Console.WriteLine(a == b); // true
```

Въвеждане на  
еднаква стойност

- **bool** – ключова дума, с която се инициализира булева променлива
- Има само следните две стойности **true** (вярно) или **false** (грешно)
- Може да се създаде и с условие, което се свежда до true или false

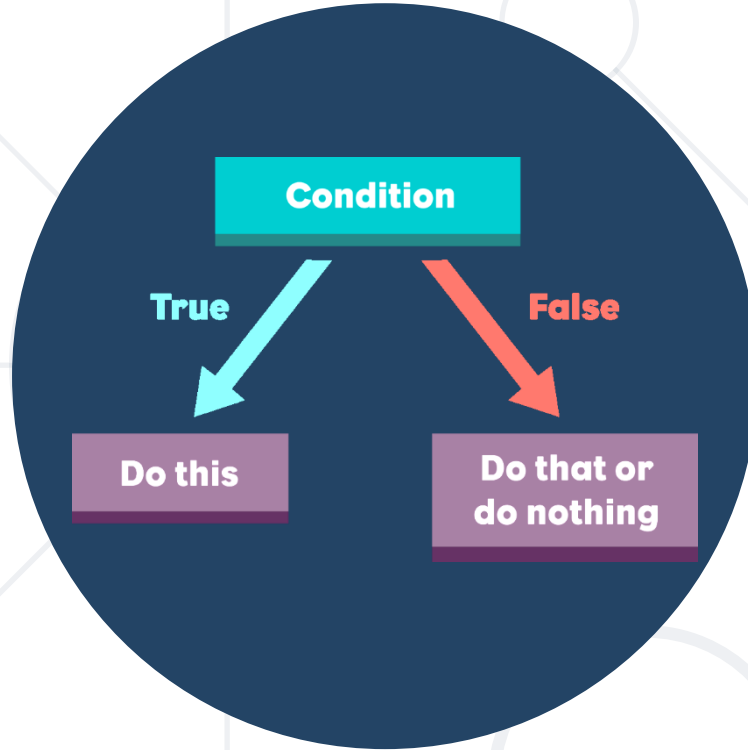
```
bool isValid = true;
```

```
bool isPositive = a > 0;
```

# Булева променлива - Пример

```
int a = 5;  
bool isPositive = a > 0;  
Console.WriteLine(isPositive); // true
```

```
int a = -5;  
bool isPositive = a > 0;  
Console.WriteLine(isPositive); // false
```




# Условни конструкции

Прости проверки



# Прости проверки

- Често проверяваме условия и извършваме действия според резултата



Условие  
(булев израз)

```
if (...)  
{  
    // код за изпълнение  
}
```

Код за изпълнение при  
вярност на условието

- Резултатът е **true** или **false**

- Напишете **програма**, която:
  - **Чете** оценка (**число**), въведена от потребителя
  - **Проверява** дали е отлична
  - **Отпечатва на конзолата** "Excellent!", ако оценката е по-голяма или равна на 5.50
- Пример:



**Read input**

**grade  $\geq$  5.50**

false

**No output**

true

**Print output**



# Прости проверки – If-else

- При **невярност** (**false**) на условието, можем да изпълним други действия – чрез **else** конструкция



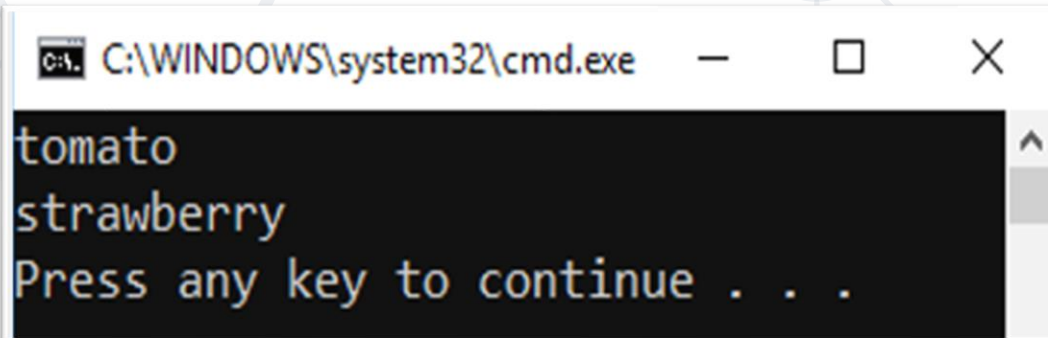
```
if (...)
{
    // код за изпълнение
}
else
{
    // код за изпълнение
}
```

Код за изпълнение  
при невярност на  
условието

- Къдравите скоби **{ }** въвеждат блок (група команди)

```
string color = "red";  
if (color == "red")  
{  
    Console.WriteLine("tomato");  
    Console.WriteLine("strawberry");  
}  
else  
{  
    Console.WriteLine("banana");  
    Console.WriteLine("bye");  
}
```

Изпълняват се  
редовете в  
съответния блок



```
C:\WINDOWS\system32\cmd.exe  
tomato  
strawberry  
Press any key to continue . . .
```

- Ако **махнем скобите**, се изпълнява съответния блок

```
string color = "red";  
if (color == "red")  
    Console.WriteLine("tomato");  
else  
    Console.WriteLine("banana");  
Console.WriteLine("bye");
```



Изпълнява се винаги – не е част от if/else конструкцията

# По-голямото число – условие

- Напишете програма, която:
  - Чете две **цели** числа
  - Отпечатва на конзолата **по-голямото** от тях
- Пример:

5  
8



8

7  
3



7



Read input

$\text{num1} > \text{num2}$

Print output

false

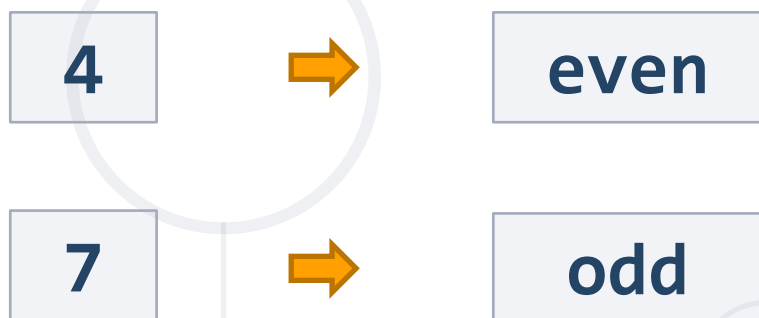
true

Print output



# Четно или нечетно число – условие

- Напишете програма, която:
  - Проверява дали едно число е **четно** или **нечетно**
  - Ако е четно отпечатва на конзолата **"even"**
  - Ако е нечетно отпечатва на конзолата **"odd"**
- Пример:



# Четно или нечетно – решение

```
int num = int.Parse(Console.ReadLine());  
if (num % 2 == 0)  
{  
    Console.WriteLine("even");  
}  
else  
{  
    Console.WriteLine("odd");  
}
```



**Закръгляне и форматиране**

- В програмирането можем да закръгляме дробни числа

- Закръгляне до следващо (**по-голямо**) цяло число:

```
double up = Math.Ceiling(23.45); // 24
```

- Закръгляне до предишно (**по-малко**) цяло число:

```
double down = Math.Floor(45.67); // 45
```

- Намиране на **абсолютна** стойност

```
int example1 = Math.Abs(-50); // 50  
int example2 = Math.Abs(50); // 50
```



- Закръгляне до 2 знака след десетичната запетая:

```
double round = Math.Round(45.67852, 2); // 45.68
```

- Форматиране до 2 знака след десетичната запетая:

```
Console.WriteLine("{0:F2}", 123.456); // 123.46
```

Брой символи след  
десетичната запетая

- Разлика между форматиране и закръгляне:

```
Console.WriteLine(Math.Round(45.60000, 4)); // 45.6  
Console.WriteLine("{0:F4}", 45.60000); // 45.6000
```



# Дебъгване

Прости операции с дебъгер

- Процес на проследяване на изпълнението на програмата
- Това ни позволява да откриваме грешки (бъгове)




Breakpoint

```
9  class Program
10 {
11     0 references
12     static void Main(string[] args)
13     {
14         string currentDay = Console.ReadLine();
15         double myMoney = 0;
16         if (currentDay.Equals("31.12.2016"))
17         {
18             double salary = double.Parse(Console.ReadLine());
19             myMoney = myMoney + salary;
20         }
21     }
22 }
23 }
```

# Дебъгване във Visual Studio

- Натискане на **[F5]** ще стартира програмата в debug режим
- Можем да преминем към следващата стъпка с **[F10]**
- Можем да създаваме **[F9]** стопери – breakpoints
  - До тях можем директно да стигнем използвайки **[F9]**



```
11 static void Main(string[] args)
12 {
13     string currentDay = Console.ReadLine();
14     double myMoney = 0;
15
16     if (currentDay.Equals("31.12.2016"))
17     {
18         double salary = double.Parse(Console.ReadLine());
19         myMoney = myMoney + salary;
20     }
21     Console.WriteLine(myMoney);
22 }
```





**Серии от проверки**

По-сложни условни конструкции

# Сerii от проверки

- Конструкцията `if/else` - `if/else...` е серия от проверки



```
if (...)  
    // код за изпълнение  
else if (...)  
    // код за изпълнение  
else if (...)  
    // код
```


**TRUE OR FALSE?**

- При истинност на едно условие, **не се продължава** към проверяване на следващите условия

# Серия от проверки – пример

- Програмата проверява първото условие, установява, че е вярно и приключва

```
int a = 7;  
if (a > 4)  
    Console.WriteLine("Bigger than 4");  
else if (a > 5)  
    Console.WriteLine("Bigger than 5");  
else  
    Console.WriteLine("Equal to 7");
```



Извежда на  
конзолата само  
"Bigger than 4"

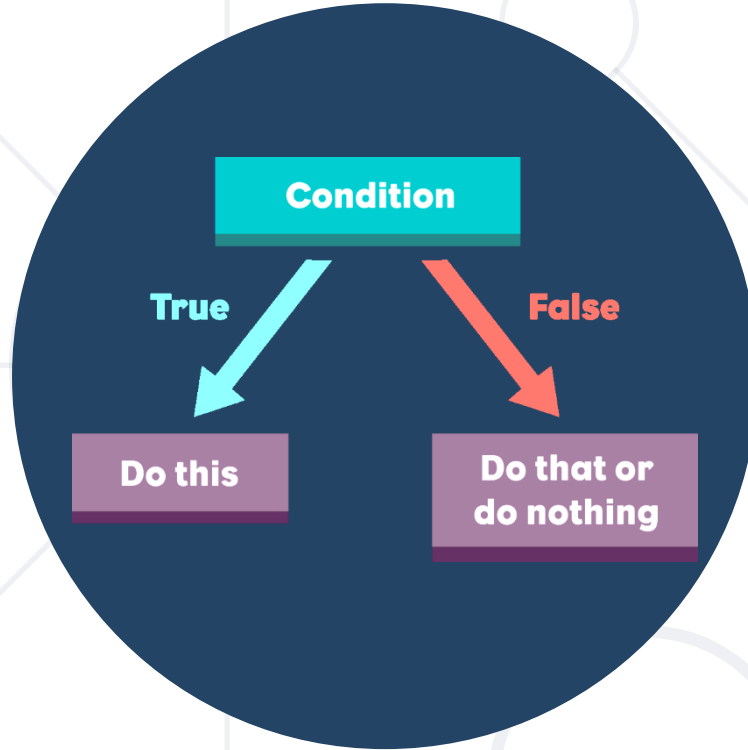


**Живот на променлива**

Диапазон на използване

- Обхват, в който може да бъде използвана
  - Пример: Променливата **salary** съществува **само** в блока от код на **if**-конструкцията

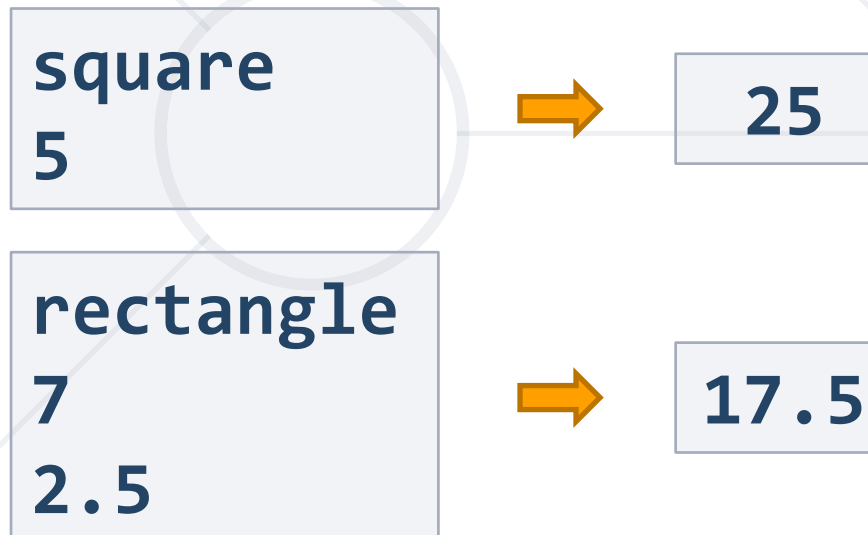
```
string currentDay = "Monday";  
if (currentDay == "Monday")  
{  
    double salary = double.Parse(Console.ReadLine());  
}  
Console.WriteLine(salary); // Error!
```



# Условни конструкции

Решаване на задачи в клас (лаб)

- Напишете програма, която:
  - Прочита **вид** на **геометрична фигура** ("square", "rectangle", "circle" или "triangle")
  - Пресмята **лицето** спрямо вида на фигурата
- Примерен вход и изход:



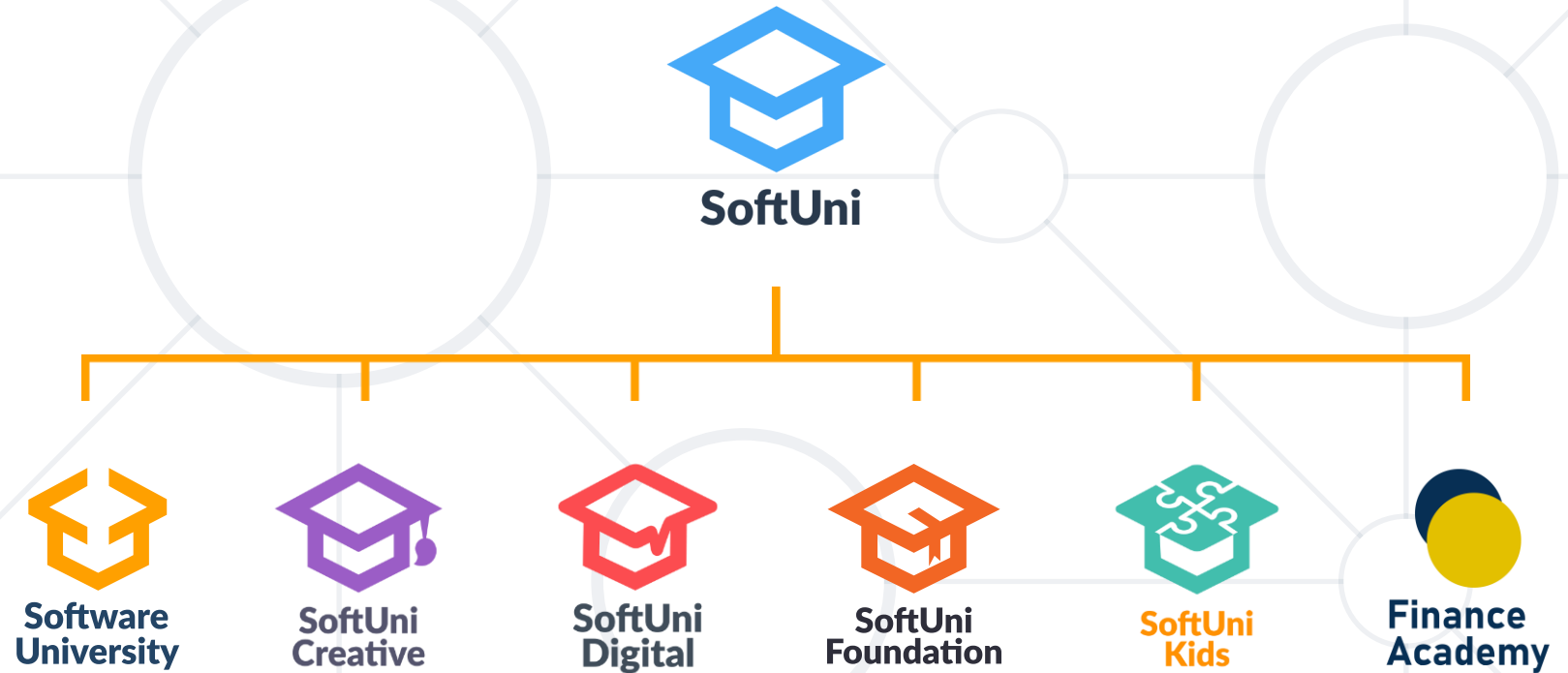
```
string shape = Console.ReadLine();
double area = 0;
if(shape == "square")
    double side = double.Parse(Console.ReadLine());
    area = side * side;
else if(shape == "rectangle")
    double sideA = double.Parse(Console.ReadLine());
    double sideB = double.Parse(Console.ReadLine());
    area = sideA * sideB;
//TODO: add more conditions
Console.WriteLine(area);
```



- Оператори за сравнение
- Конструкции за проверка на условие – **if** и **if-else**
- Закръгляне и форматиране
- Серии от проверки
- Дебъгване
- Живот на променливата



# 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](http://softuni.bg)
- Фондация "Софтуерен университет"
  - [softuni.foundation](http://softuni.foundation)
- Софтуерен университет @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

