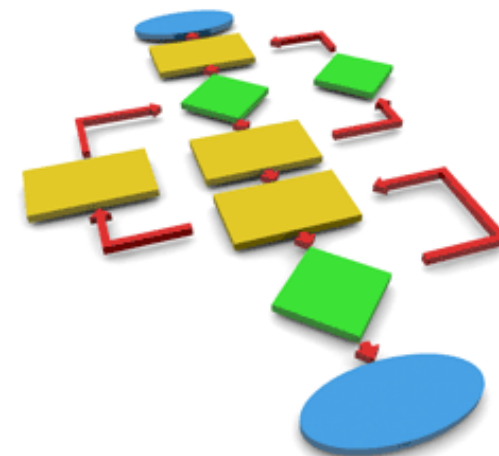
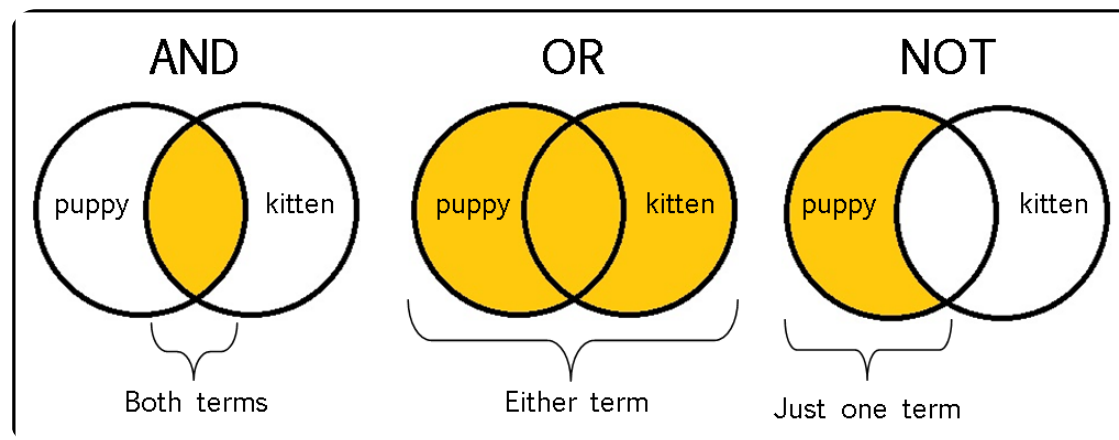


# Булев тип данни

*Оператори за сравнение и логически операции*

Увод в  
програмирането



# Съдържание

1. Какво е булев тип данни?
2. Оператори за сравнение
3. Логически операции
  - логическо И
  - логическо ИЛИ
  - логическо отрицание
4. Логически изрази
  - ...и как могат да бъдат опростени



# Булев тип

- Булевия тип (**bool**) съдържа **true** (истина) или **false** (лъжа):

```
int a = 1;
int b = 2;

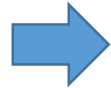
bool greaterAB = (a > b);
Console.WriteLine(greaterAB);    // False

bool equalA1 = (a == 1);
Console.WriteLine(equalA1);      // True
```

# Задача: Булева променлива

- Въведете **низ**, конвертирайте го към променлива от **булев** тип и **изведете** **“Yes”** ако в нея има **true** и **“No”** в противен случай.

True



Yes

False



No

```
string input = Console.ReadLine();
bool variable = Convert.ToBoolean(input);
if (variable == true) {
    Console.WriteLine("Yes");
} else {
    Console.WriteLine("No");
}
```

Използвайте  
[Convert.ToBoolean\(string\)](#).

# Сравняване на числа

- В програмирането можем да сравняваме стойности:

```
var a = 5;  
var b = 10;  
Console.WriteLine(a < b);           // True  
Console.WriteLine(a > 0);           // True  
Console.WriteLine(a > 100);         // False  
Console.WriteLine(a < a);           // False  
Console.WriteLine(a <= 5);          // True  
Console.WriteLine(b == 2 * a);      // True
```

Оператор < (по-малко)

Оператор >  
(по-голямо)

Оператор <=  
(по-малко  
или равно)

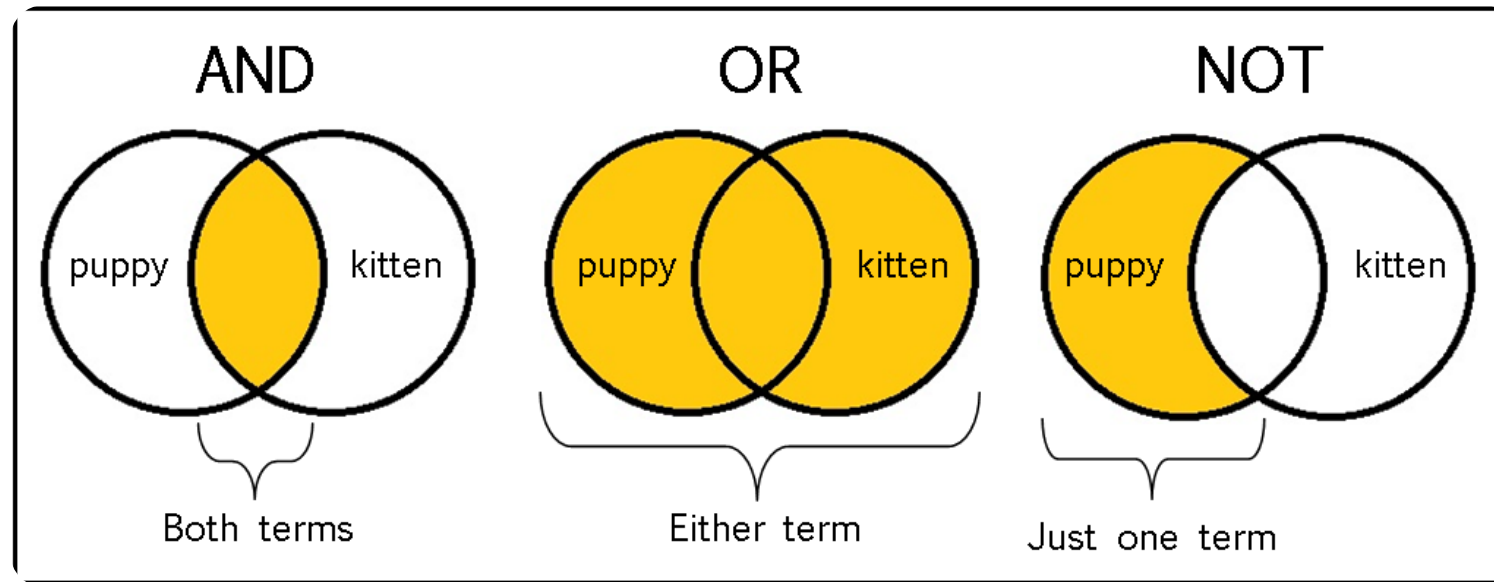
Оператор == (равно)

# Оператори за сравнение

Оператор	Означение	Работи за
Проверка за равенство	==	числа, текстове, дати
Проверка за различно	!=	
По-голямо	>	числа, дати, други сравними типове
По-голямо или равно	>=	
По-малко	<	
По-малко или равно	<=	

■ Пример:

```
var result = (5 <= 6);
Console.WriteLine(result); // True
```



# Логически операции и изрази

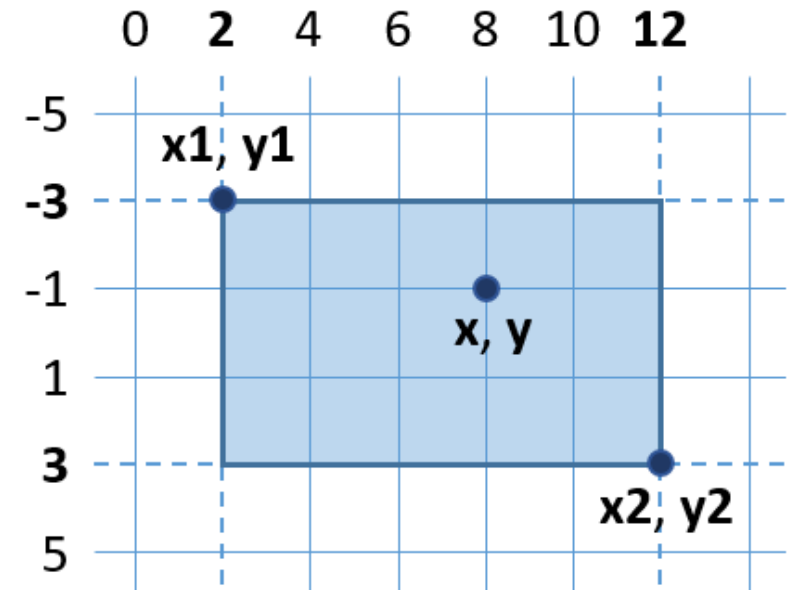
Логическо "и", логическо "или", логическо отрицание и скоби

# Логическо "И"

- Логическо "И" (оператор **&&**) означава няколко условия да са изпълнени едновременно

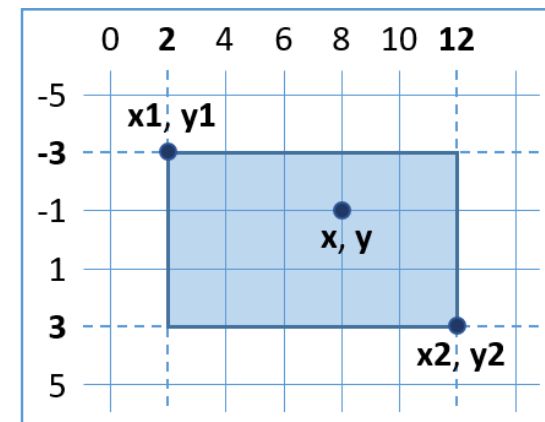
```
var inside = (x >= x1 && x <= x2 && y >= y1 && y <= y2);
```

- Пример: проверка дали точка **{x, y}** се намира вътре в правоъгълника **{x1, y1} – {x2, y2}**
- Необходимо е точката **{x, y}** да е:
  - надясно от **x1** и наляво от **x2** и надолу от **y1** и нагоре от **y2**





# Пример: Точка в правоъгълник



- Точка е **вътрешна** за даден правоъгълник, ако е:
  - надясно от лявата му страна, наляво от дясната му страна, надолу от горната му страна и нагоре от долната му страна

```
var x1 = double.Parse(Console.ReadLine());
var y1 = double.Parse(Console.ReadLine());
// TODO: finish this for x2,y2,x,y
var inside = x >= x1 && x <= x2 && y >= y1 && y <= y2)
if (inside)
    { Console.WriteLine("Inside"); }
else
    { Console.WriteLine("Outside"); }
```

# Логическо "ИЛИ"

- Логическо "ИЛИ" (оператор `||`) означава да е изпълнено поне едно измежду няколко условия

```
if (s == "banana" || s == "apple" || s == "kiwi")  
    Console.WriteLine("fruit");
```

- Задача: плод или зеленчук?

lemon



fruit

- Плодовете "fruit" са: banana, apple, kiwi, cherry, lemon, grapes
- Зеленчуците "vegetable" са: tomato, cucumber, pepper, carrot
- Всички останали са "unknown"

java



unknown

## Пример: Плод или зеленчук?

Решение на задачата "плод или зеленчук":

```
var s = Console.ReadLine();  
if (s == "banana" || s == "apple" || s == "kiwi" ||  
    s == "cherry" || s == "lemon" || s == "grapes")  
    { Console.WriteLine("fruit"); }  
else if (s == "tomato" || s == "cucumber" ||  
         s == "pepper" || s == "carrot")  
    { Console.WriteLine("vegetable"); }  
else  
    { Console.WriteLine("unknown"); }
```

# Логическо отрицание

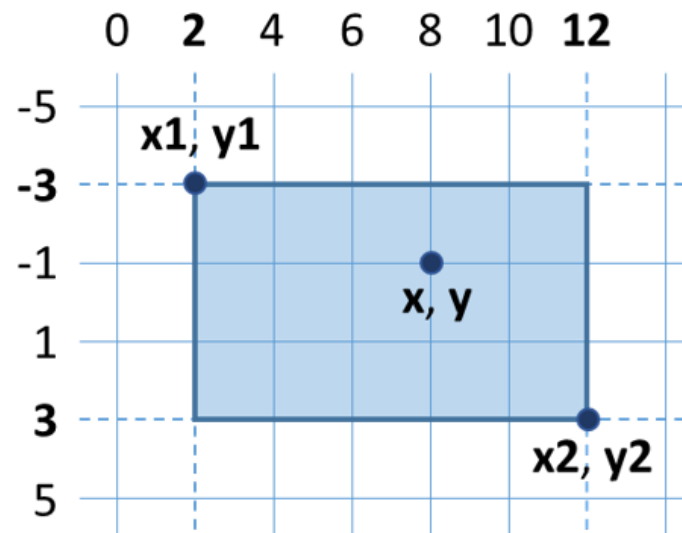
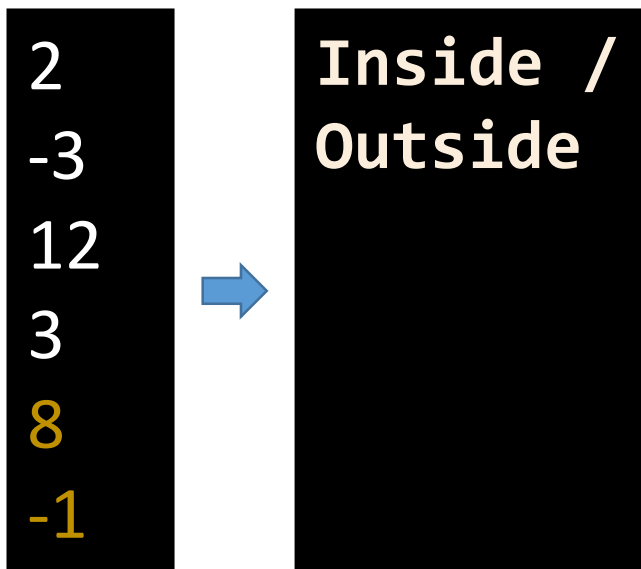
- Логическо **отрицание** (оператор **!**) означава да **не е изпълнено** дадено условие
- Пример:
  - Дадено число е **валидно**, ако е в диапазона **[100...200]** или е **0**
  - Да се направи проверка за **невалидно число**

```

var inRange = (num >= 100 && num <= 200) || num == 0;
if (!inRange)
{ Console.WriteLine("invalid"); }
    
```

# Пример: Точка върху страна на правоъгълник

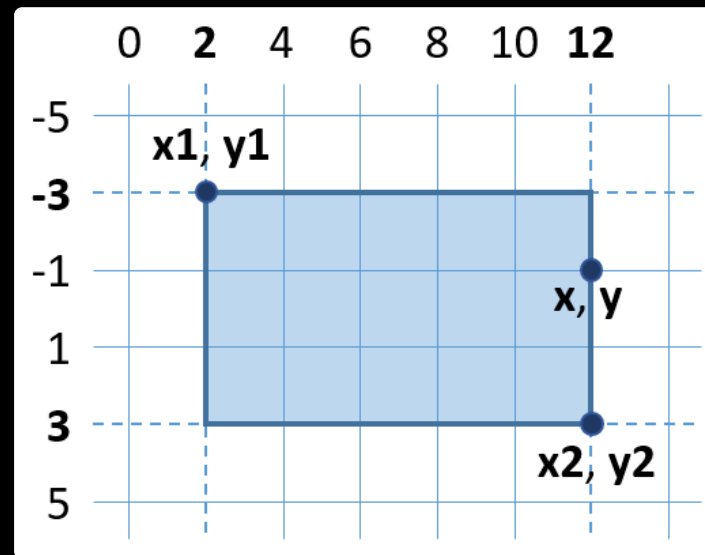
- Да се напише програма, която чете 6 десетични числа **x1**, **y1**, **x2**, **y2**, **x** и **y**
  - Печата дали точката е **върху страна от правоъгълника** или не
  - Ограничения: **x1** < **x2** и **y1** < **y2**



# По-сложни логически условия

- Точка лежи **върху някоя от страните** на правоъгълник, ако:
  - **x** съвпада с **x1** или **x2** и същевременно **y** е между **y1** и **y2** или
  - **y** съвпада с **y1** или **y2** и същевременно **x** е между **x1** и **x2**

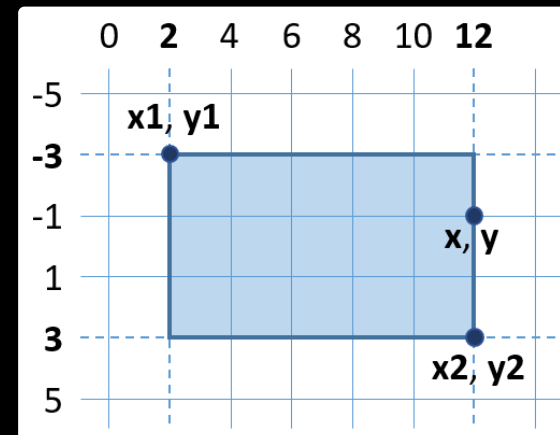
```
if (((x == x1 || x == x2) &&  
    (y >= y1 && y <= y2)) ||  
    ((y == y1 || y == y2) &&  
    (x >= x1 && x <= x2)))  
{ Console.WriteLine("Border"); }  
else  
{ Console.WriteLine("Inside / Outside"); }
```



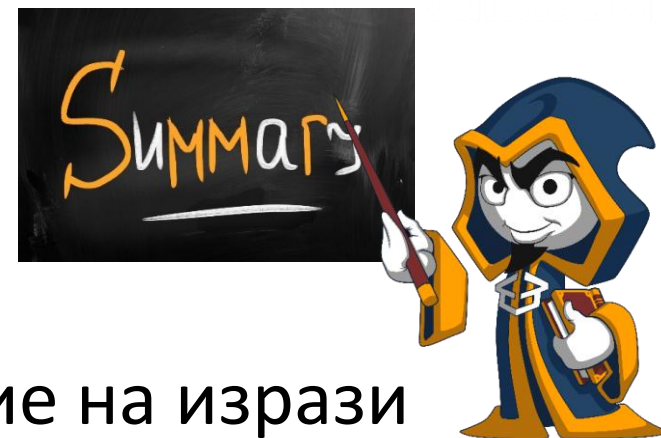
# Опростиране на логически условия

- Предходното условие може да се опрости ето така:

```
var onLeftSide = (x == x1) && (y >= y1) && (y <= y2);  
var onRightSide = (x == x2) && (y >= y1) && (y <= y2);  
var onUpSide = (y == y1) && (x >= x1) && (x <= x2);  
var onDownSide = (y == y2) && (x >= x1) && (x <= x2);  
  
if (onLeftSide || onRightSide ||  
    onUpSide || onDownSide)  
{ Console.WriteLine("Border"); }  
else  
{ Console.WriteLine("Inside / Outside"); }
```



# Какво научихме днес?



- Булев тип: съдържа **True** или **False**
- Можем да използваме оператори за сравнение на изрази

```
Console.WriteLine(5 > 10);      // False  
Console.WriteLine(10 == 2 * 5); // True
```

- По-сложни проверки с **&&**, **||**, **!** и **()**

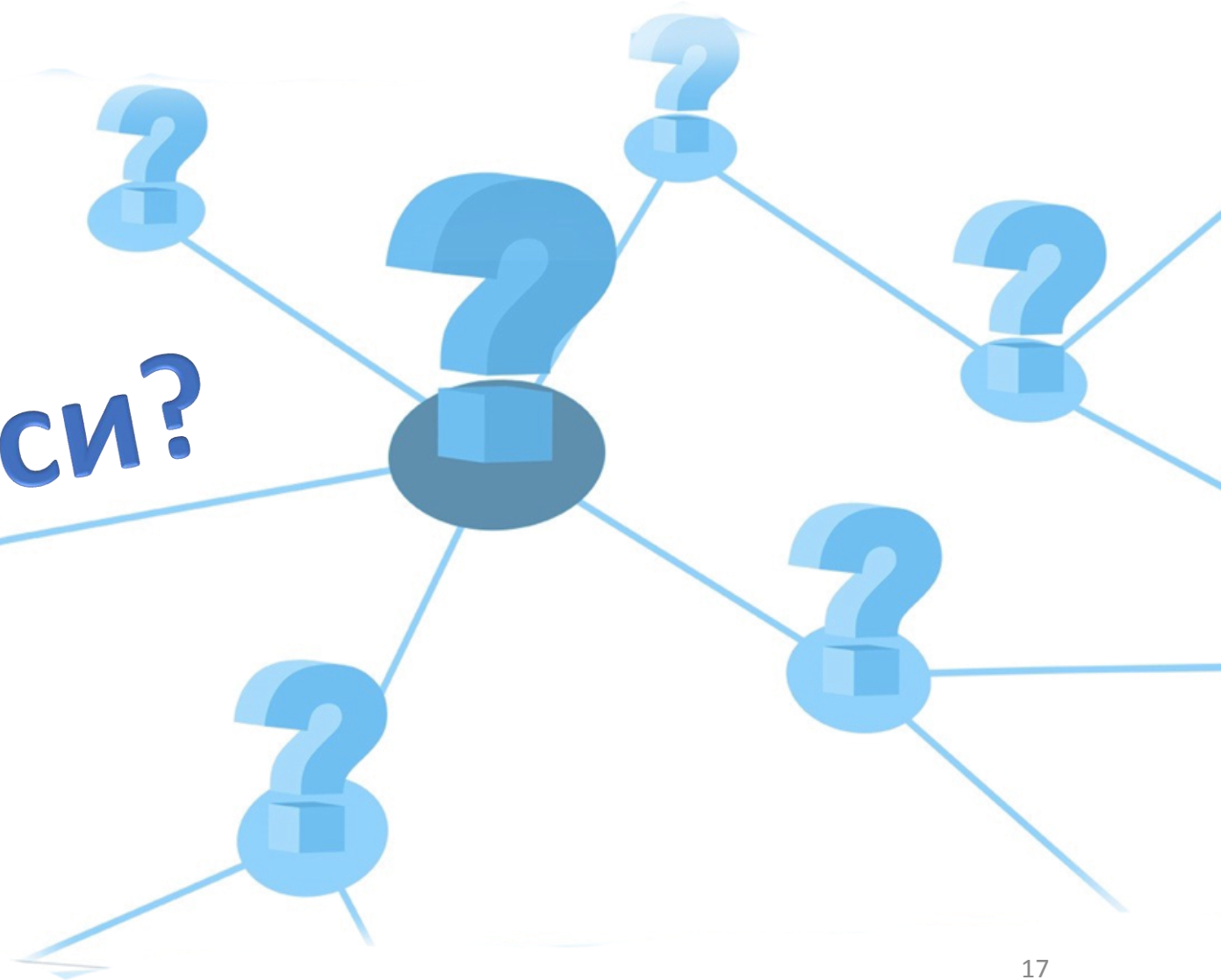
```
if ((x == left || x == right) && y >= top && y <= bottom)  
    Console.WriteLine("Point on the left or right side.");
```



# Булев тип данни



**Въпроси?**



# Договор за ползване

Този курс (слайдове, примери, задачи и др.) се разпространяват под свободен лиценз "[Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International](https://creativecommons.org/licenses/by-nc-sa/4.0/)"



Базиран е на учебните материали на [НП „Обучение за ИТ Кариера“](#).

Може да съдържа части от следните източници:

- Книга "[Основи на програмирането със C#](#)" от Светлин Наков и колектив с лиценз [CC-BY-SA](#)