

Условен оператор

- Синтаксис: `if (<условие>) <действие1> [else <действие2>]`
- Възможно е <действие2> да бъде друг условен оператор

`if (<условие1>) <действие1> else if (<условие2>) <действие2> [else <действие3>]`

Примери за условен оператор

- `int x,y;`
- `std::cin>>x>>y;`

- `if(x<y) std::cout<<"x is lower than y";else;`
- Горното е равносилно на `if(x<y) std::cout<<"x is lower than y";`

- `if(x<y) std::cout<<"x is lower than y\n";`
`else if(x>y) std::cout<<"x is higher than y\n";`
`else std::cout<<"x is equal to y\n";`

Score като команда на условен оператор

- `if(a<b)`

```
{
```

```
//Many lines code here
```

```
}
```

Вложени условни оператори

- `if (a > 0) if (b > 0) std::cout << 1; else std::cout << 3;`

- Какво ще стане ако а е -1?

<code>if (a > 0){</code>		<code>if(a>0){</code>
<code>if (b > 0){</code>		<code>if(b>0){</code>
<code>std::cout << 1;</code>		<code>std::cout<<1</code>
<code>}</code>		<code>}</code>
<code>}</code>	ИЛИ	<code>else{</code>
<code>else{</code>		<code>std::cout<<3;</code>
<code>std::cout << 3;</code>		<code>}</code>
<code>}</code>		<code>}</code>

Вложени условни оператори

- `if (a > 0) if (b > 0) std::cout << 1; else std::cout << 3;`
- Какво ще стане ако `a` е `-1`?

```
if(a>0){  
    if(b>0){  
        std::cout<<1  
    }  
    else{  
        std::cout<<3;  
    }  
}
```

Добри практики при писане на условен оператор

- `if (a > 0) if (b > 0) std::cout << 1; else std::cout << 3;`

```
if (a > 0)
{
    if (b > 0)
    {
        std::cout << 1;
    }
    else
    {
        std::cout << 3;
    }
}
```

Оператор за многозначен избор

```
• int a = 10;  
if(a==0)  
    std::cout<<"nula";  
else if (a==1)  
    std::cout<<"edno";  
else if(a==2)  
    std::cout<<"dve";  
.....  
else if(a==10)  
    std::cout<<"deset";
```

Оператор за многозначен избор

- Когато искаме да имаме няколко различни изхода в зависимост от стойността на 1 променлива, `if ; else` невинаги е оптимален
- `switch (<променлива>) {`
`{ case <стойност> : { <действие> } }`
`[default : { <действие> }]`
`}`

Оператор за многозначен избор

- Какво не е наред с кода от предния слайд?
- При въвеждане на 1 ,например, изходът ще бъде:

ednodvetrichetiripetshestsedemosemdevetdesettoobig

(оцветяването е само с цел да се чете по-лесно)

Оператор за прекъсване

- Ключова дума: `break`
- Операторът за прекъсване `break` има две функции
- Когато се използва в `switch-statement`, при извикването си казва на програмата да спре да изпълнява командите от `switch`
- Няма смисъл да се слага след последния случай
- Втората функция ще разгледаме по-късно

Оператор за многозначен избор

- `switch(a)`
{
 `case 1: std::cout<<"edno"; break;`
 `case 2: std::cout<<"dve"; break;`

 `case 10: std::cout<<"deset";break;`
 `default: std::cout<<"too big";`
}