If-else конструкции

1. Същност на условната логика

Понякога се налага да извършим някакви действия в нашата програма само и единствено, ако имаме изпълнено дадено условие.

***Пример:***

Велизар може да ходи на дискотека само, ако има навършени 18 години, в противен случай не.

***Като код това изглежда така:***

int age = int.Parse(Console.ReadLine()); //Въвеждаме годините на Велизар

if (age >= 18)

{

Console.WriteLine("Велизар може да ходи на дискотека!");

}

else

{

Console.WriteLine("Велизар не може да ходи на дискотека! Още е малък.");

}

***Важно: Знакът >= означава „по-голямо или равно“!***

Както може да видите по-горе за използване на условна логика в **C#**, а и в програмните езици като цяло, се използват ключовите думи ‘if’ и ‘else’.

if (условие)

{

Някакво действие, което се изпълнява, ако условието е покрито

}

else

{

Някакво действие, което се изпълнява, ако условието НЕ е покрито

}

1. If – else if- else конструкции

Понякога се налага нашите условни конструкции да бъдат малко по-комплексни.

Пример:

int age = int.Parse(Console.ReadLine()); //Въвеждаме годините на Велизар

if (age < 18)

{

Console.WriteLine("Велизар не може да ходи на дискотека! Още е малък.");

}

else if (age > 60)

{

Console.WriteLine("Велизар е прекалено стар за дискотека!");

}

else

{

Console.WriteLine($"Велизар може да ходи на дискотека, защото е на {age} години.");

}

***Важно:*** В този пример използвахме езиковата конструкция „else if“. Тук в случай, че първото условие не е покрито, програмата ще мине през второто условие и ако и то не е покрито ще изпълни третото условие в “else” секцията.

1. По-комплексни условия

Много често се налага условията, които поставяме да бъдат малко по-сложни от тези по-горе:

Пример:

int age = int.Parse(Console.ReadLine()); //Въвеждаме годините на Велизар

if (age < 0)

{

Console.WriteLine("Грешно въведени години!");

}

if (age < 18 && age >= 0)

{

Console.WriteLine("Велизар не може да ходи на дискотека! Още е малък.");

}

else if (age >= 18 && age <= 60)

{

Console.WriteLine($"Велизар може да ходи на дискотека, защото е на {age} години.");

}

else

{

Console.WriteLine("Велизар НЕ може да ходи на дискотека, защото е стар.");

}

Тук „&&“ означава „и“. Т.е. условието важи, ако първата И втората му част са валидни.

Пример: age < 18 И age >= 0 --> Когато искаме да съединим две условие с оператор за И, пишем „&&“.

Следващата таблица показва работата с логическите оператори **(&&,** **||** и **!**)

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Description** | **Example (а = true, b = false)** |
| && | Logical AND | Връща True, ако всички условия, които обхваща са True | a && b (false) |
| || | Logical OR | Връща True, ако поне едно от условията, които обхваща е True | a || b (true) |
| ! | Logical NOT | Връща обратното състояние | ! (a && b) (true) |

Таблица, показваща използването на **AND** и **OR** операторите:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operand1** | **Operand2** | **AND** | **OR** |
| true | true | true | true |
| true | false | false | true |
| false | true | false | true |
| false | false | false | false |

Таблица, показваща използването на **NOT** оператора:

|  |  |
| --- | --- |
| **Operand** | **NOT** |
| true | false |
| false | true |

1. Вложени условия

В кодът ни е абсолютно допустимо да използваме вложени if-else условия.

Пример:

int age = int.Parse(Console.ReadLine()); //Въвеждаме годините на Велизар

if (age < 0)

{

Console.WriteLine("Грешно въведени години!");

}

if (age < 18 && age >= 0)

{

Console.WriteLine("Велизар не може да ходи на дискотека! Още е малък.");

}

else if (age >= 18 && age <= 100)

{

if (age > 60)

{

Console.WriteLine("Велизар НЕ може да ходи на дискотека, защото е стар.");

}

else

{

Console.WriteLine($"Велизар може да ходи на дискотека, защото е на {age} години.");

}

}

else

{

Console.WriteLine("Велизар най-вероятно не е сред живите :D.");

}