Турсунов Баходурхон

#8. .NET Syntax. Boolean operators. Ternary operator. If – operator. Switch – operator. Switch – expression.

1. **Объясните логические условные операторы AND (&&) и OR ("||") в C# и когда они возвращают истину.**

* Оператор AND возвращает true, только если оба условия истинны. И если хоть одно из условий является ложным, результат будет false.
* Оператор OR возвращает true, если хотя бы одно из условий истинно. И результат будет false, только если оба условия являются ложными.

1. **В чем разница между логическими операторами AND (&) и OR (|) и их условными аналогами (&& и "||")?**

* Логический оператор AND (&) вычисляет оба операнда независимо от того, что происходит с другими операндами. Даже если левый операнд является ложным, правый операнд все равно вычисляется. И возвращает true если оба операнда истинны.
* Логический оператор OR (|) вычисляет оба операнда независимо от того, что происходит с другими операндами. Даже если левый операнд является истинным, правый операнд все равно вычисляется. Возвращает true, если хотя бы один из операндов истинен.

1. **Какова роль оператора "else" и когда он выполняется по отношению к оператору "if"?**

Оператор else используется в паре с оператором if и предназначен для обработки альтернативного условия, которое выполняется в случае, если условие в операторе if оказывается ложным.

1. **Каково назначение оператора "switch" в C# и чем он отличается от структуры "if-else"?**

Оператор switch предназначен для обработки множества возможных значений выражения. Он предоставляет более удобный способ сравнения значения выражения с различными вариантами и выполнения соответствующего блока кода в зависимости от соответствия.

1. **Опишите компоненты выражения switch, включая выражение, плечи выражения switch и шаблоны.**

Выражение (switch):

Это выражение, которое вычисляется, и его значение сравнивается с различными вариантами (case-выражениями).

Пример:

int dayOfWeek = 3;

switch (dayOfWeek)

{

// ...

}

Плечи выражения switch (Switch Arms):

Плечи выражения switch содержат case-выражения, которые сравниваются с выражением switch. Когда значение выражения switch соответствует значению в каком-то case-выражении, выполняется соответствующий блок кода.

Пример:

int dayOfWeek = 3;

switch (dayOfWeek)

{

case 1:

Console.WriteLine("Понедельник");

break;

case 2:

Console.WriteLine("Вторник");

break;

// ...

}

Шаблоны (Patterns):

В C# 7.0 и выше появились шаблоны (pattern matching), которые позволяют более гибко сопоставлять значения. Это может быть использовано внутри case-выражений для более сложных условий сопоставления.

Пример с использованием шаблона:

object obj = "строка";

switch (obj)

{

case int i:

Console.WriteLine($"Это целое число: {i}");

break;

case string s:

Console.WriteLine($"Это строка: {s}");

break;

default:

Console.WriteLine("Неизвестный тип");

break;

}

В этом примере, шаблон case int i: соответствует, если obj является целым числом, а case string s: - если obj является строкой.

1. **Приведите пример выражения switch в C#, демонстрирующий, как оно преобразует значения на основе сопоставления шаблонов.**

using System;

class Program

{

static void Main()

{

object value = 42;

switch (value)

{

case int intValue:

Console.WriteLine($"Это целое число: {intValue}");

break;

case string stringValue:

Console.WriteLine($"Это строка: {stringValue}");

break;

case double doubleValue when doubleValue > 0:

Console.WriteLine($"Это положительное вещественное число: {doubleValue}");

break;

default:

Console.WriteLine("Неизвестный тип");

break;

}

}

}

В этом примере:

Если переменная value является целым числом (int), то выполняется первый блок кода.

Если value является строкой (string), то выполняется второй блок кода.

Если value является положительным вещественным числом (double), то выполняется третий блок кода. Обратите внимание на условие when doubleValue > 0, которое дополнительно проверяет, что значение положительно.

Если ни один из предыдущих вариантов не соответствует, то выполняется блок кода по умолчанию.

Practice:

1. Объявите две переменные age и isStudent. Напишите условие, которое будет истинным, если возраст (age) больше 18 и человек не является студентом (isStudent равно false).

class Program

{

static void Main(string[] args)

{

int age = int.Parse(Console.ReadLine());

bool isStudent = false;

if(age >= 18 || isStudent) Console.WriteLine("true");

else Console.WriteLine("false");

}

}

1. Напишите программу, которая выводит числа от 1 до 100. Если число делится на 3, выведите "Fizz". Если число делится на 5, выведите "Buzz". Если число делится и на 3, и на 5, выведите "FizzBuzz". В противном случае выведите само число.

class Program

{

static void Main(string[] args)

{

for (int i = 1; i <= 100; i++)

{

if (i % 3 == 0 && i % 5 == 0) Console.WriteLine("FizzBuzz");

else if (i % 3 == 0) Console.WriteLine("Fizz");

else if (i % 5 == 0) Console.WriteLine("Buzz");

else Console.WriteLine(i);

}

}

}

1. Даны две логические переменные, a и b. Напишите выражение, которое равно true только в том случае, если ровно одна из переменных a или b равна true.

class Program

{

static void Main(string[] args)

{

bool a = true;

bool b = false;

bool result = a ^ b; // здесь оператор ^ означает исключающее или.

Console.WriteLine(result);

}

}

1. У вас есть массив чисел. Напишите программу, которая выводит сумму всех нечетных чисел в массиве.

class Program

{

static void Main(string[] args)

{

int[] arr = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int resultSum = SumOfOddNumbers(arr);

Console.WriteLine(resultSum);

}

static int SumOfOddNumbers(int[] numbers)

{

int result = 0;

foreach (var num in numbers) // здесь используя цикл foreach проходимся по каждому элементу массива

{

if (num % 2 != 0) // здесь проверяется условие на нечетность

{

result += num; // а здесь нечетные числа суммируются

}

}

return result; // здесь все возвращается

}

}