

Теоретический материал

КОНСТРУКЦИЯ **if..else**

Условные конструкции - один из базовых компонентов многих языков программирования, которые направляют работу программы по одному из путей в зависимости от определенных условий. Одной из таких конструкций в языке программирования C# является конструкция **if..else**

Конструкция **if..else** проверяет истинность некоторого условия и в зависимости от результатов проверки выполняет определенный код.

Ее простейшая форма состоит из блока **if**:

```
if (условие)
{
    выполняемые инструкции
}
```

После ключевого слова **if** ставится условие. Условие должно представлять значение типа **bool**. Это может быть непосредственно значение типа **bool** или результат условного выражения или другого выражения, которое возвращает значение типа **bool**. И если это условие истинно (равно **true**), то срабатывает код, который помещен далее после условия внутри фигурных скобок.

Если блок **if** содержит одну инструкцию, то мы можно его сократить, убрав фигурные скобки.

Также можно соединить сразу несколько условий, используя логические операторы.

Блок **else** выполняется, если условие после **if** ложно, то есть равно **false**. Если блок **else** содержит только одну инструкцию, то опять же можно его сократить, убрав фигурные скобки. Язык C# поддерживает синтаксис конструкции **else if**.

КОНСТРУКЦИЯ **switch**

Конструкция **switch** имеет следующее формальное определение:

```
switch (выражение)
{
    case значение1:
        код, выполняемый если выражение имеет значение1
        break;
    case значение2:
        код, выполняемый если выражение имеет значение2
        break;
    //.....
    case значениеN:
        код, выполняемый если выражение имеет значениеN
        break;
    default:
        код, выполняемый если выражение не имеет ни одно из выше указанных значений
        break;
}
```

После ключевого слова **switch** в скобках идет сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после оператора **case**. И если совпадение будет найдено, то будет

выполняться определенный блок **case**. В конце каждого блока **case** должен ставиться один из операторов перехода: **break**, **goto case**, **return** или **throw**. Как правило, используется оператор **break**. При его применении другие блоки **case** выполняться не будут.

Если необходимо, чтобы, после выполнения текущего блока **case** выполнялся другой блок **case**, то можно использовать вместо **break** оператор **goto case**:

```
int number = 1;
switch (number)
{
    case 1:
        Console.WriteLine("case 1");
        goto case 5; // переход к case 5
    case 3:
        Console.WriteLine("case 3");
        break;
    case 5:
        Console.WriteLine("case 5");
        break;
    default:
        Console.WriteLine("default");
        break;
}
```

ТЕРНАРНАЯ ОПЕРАЦИЯ

Тернарную операция позволяет проверить некоторое условие и в зависимости от его истинности выполнить некоторые действия. Она имеет

следующий синтаксис:

[первый операнд - условие] ? [второй операнд] : [третий операнд]

Здесь сразу три операнда. В зависимости от условия тернарная операция возвращает второй или третий операнд: если условие равно **true**, то возвращается второй операнд; если условие равно **false**, то третий. Например:

```
int x=3;  
int y=2;  
int z = x < y ? (x+y) : (x-y);  
Console.WriteLine(z); // 1
```

Здесь первый операнд (то есть условие) представляет выражение $x < y$.

Если оно равно **true**, то возвращается второй операнд - $(x+y)$, если условие равно **false**, то возвращается третий операнд - $(x-y)$.

Результат тернарной операции (то есть второй или третий операнд в зависимости от условия) присваивается переменной z .

Задание 1

Задача:

Реализуйте программу-календарь для месяца мая. При вводе номера дня месяца должна проводиться проверка, является ли этот день выходным или рабочим. Также добавьте возможность начать месяц с любого дня недели. Месяц может начаться с любого дня недели (для этого нужно ввести число — номер дня недели, с которого начинается месяц. Убедитесь, что введённое число корректно.).

Выходными считаются все субботы и воскресенья, а также даты с 1 по 5 и с 8 по 10 мая включительно.

Пример работы:

```
Введите номер дня недели, с которого начинается месяц (1-пн,...7-вс) 3
Введите день месяца18
-----Проверяем выходной ли день -----
Выходной день
```

Совет: используйте реальный календарь для проверки.

Решение:

```
using System;

namespace CalendarMay
{
    class Program
    {
        static void Main()
        {
            int startDayOfWeek;
            int dayOfMonth;
            Console.Write("Введите номер дня недели, с которого начинается месяц (1-пн, ..., 7-вс): ");
            if (!int.TryParse(Console.ReadLine(), out startDayOfWeek) || startDayOfWeek < 1 || startDayOfWeek > 7)
            {
                Console.WriteLine("Ошибка: введите число от 1 до 7.");
                return;
            }
            Console.Write("Введите день месяца: ");
            if (!int.TryParse(Console.ReadLine(), out dayOfMonth) || dayOfMonth < 1 || dayOfMonth > 31)
            {
                Console.WriteLine("Ошибка: введите число от 1 до 31.");
                return;
            }
            bool isWeekend = IsWeekend(dayOfMonth, startDayOfWeek);
            bool isHoliday = IsSpecialHoliday(dayOfMonth);

            if (isWeekend || isHoliday)
            {
                Console.WriteLine("-----Проверяем выходной ли день-----");
                Console.WriteLine("Выходной день");
            }
            else
            {
                Console.WriteLine("-----Проверяем выходной ли день-----");
                Console.WriteLine("Рабочий день");
            }
        }
        static bool IsWeekend(int day, int startDayOfWeek)
        {
            int dayOfWeek = (startDayOfWeek + day - 1) % 7;
            return dayOfWeek == 6 || dayOfWeek == 0;
        }
        static bool IsSpecialHoliday(int day)
        {
            return (day >= 1 && day <= 5) || (day >= 8 && day <= 10);
        }
    }
}
```

Ответ:

	<div>Консоль отладки Microsoft Visual Studio</div> <div>Введите номер дня недели, с которого начинается месяц (1-пн, ..., 7-вс): 3 Введите день месяца: 21 -----Проверяем выходной ли день----- Рабочий день</div>
	<div>Консоль отладки Microsoft Visual Studio</div> <div>Введите номер дня недели, с которого начинается месяц (1-пн, ..., 7-вс): sad Ошибка: введите число от 1 до 7.</div>
	<div>Консоль отладки Microsoft Visual Studio</div> <div>Введите номер дня недели, с которого начинается месяц (1-пн, ..., 7-вс): 7 Введите день месяца: 32 Ошибка: введите число от 1 до 31.</div>

Задание 2

Задача:

	<p>Реализуйте программу для банкомата.</p> <p>Пользователь хочет обналичить N рублей в банкомате. В банкомате присутствуют купюры достоинством 100, 200, 500, 1000, 2000 и 5000 рублей. Банкомат должен быть настроен таким образом, чтобы выдавать наименьшим количеством купюр, запрошенную сумму.</p> <p>После запроса суммы программа должна выдать ответ сколько купюр каждого номинала нужно выдать (4 по 200, 1 по 1 000 и т.д.) для получения суммы ровно N рублей, либо выводить сообщение, что выдать ровно N рублей невозможно, например, если N не делится на 100.</p> <p>Учтите, что банкомат не может выдать за один раз более 150 000 рублей.</p>
--	--

Решение:

	<pre>using System; namespace ATM { class Program { static void Main() { Console.Write("Введите сумму для выдачи (кратную 100): "); } } }</pre>
--	---

```

if (!int.TryParse(Console.ReadLine(), out int amount))
{
    Console.WriteLine("Ошибка: введите корректное число.");
    return;
}
if (amount < 0) {
    Console.WriteLine("Ошибка: сумма быть не отрицательной.");
    return;
}
if (amount > 150000)
{
    Console.WriteLine("Ошибка: сумма не должна превышать 150 000 рублей.");
    return;
}
if (amount % 100 != 0)
{
    Console.WriteLine("Ошибка: сумма должна быть кратна 100 рублям.");
    return;
}
int[] denominations = { 5000, 2000, 1000, 500, 200, 100 };
int[] counts = new int[denominations.Length];
int totalNotes = 0;

int remaining = amount;
for (int i = 0; i < denominations.Length; i++)
{
    counts[i] = remaining / denominations[i];
    remaining %= denominations[i];
    totalNotes += counts[i];
}
if (remaining != 0)
{
    Console.WriteLine("Ошибка: невозможно выдать точную сумму.");
    return;
}
Console.WriteLine("Выдача:");
for (int i = 0; i < denominations.Length; i++)
{
    if (counts[i] > 0)
    {
        Console.WriteLine($"{counts[i]} купюра(ы) по {denominations[i]} руб.");
    }
}

Console.WriteLine($"Общее количество купюр: {totalNotes}");
}
}

```

Ответ:

Консоль отладки Microsoft Visual Studio

Введите сумму для выдачи (кратную 100): -1000
 Ошибка: сумма быть не отрицательной.

Консоль отладки Microsoft Visual Studio

Введите сумму для выдачи (кратную 100): 150100
 Ошибка: сумма не должна превышать 150 000 рублей.

Консоль отладки Microsoft Visual Studio

Введите сумму для выдачи (кратную 100): 148800

Выдача:

29 купюра(ы) по 5000 руб.

1 купюра(ы) по 2000 руб.

1 купюра(ы) по 1000 руб.

1 купюра(ы) по 500 руб.

1 купюра(ы) по 200 руб.

1 купюра(ы) по 100 руб.

Общее количество купюр: 34