

Оператор условного перехода **if**

1. В каких случаях целесообразно применять оператор условного перехода?

Оператор условного перехода применяется в случаях, когда ход решения задачи зависит от выполнения (невыполнения) некоторого условия. В языке **С#** существует две формы оператора условного перехода:

- полная форма **if ... else ...** ;
- сокращенная форма **if ...** .

2. Какой вид имеет полная форма оператора условного перехода **if**?

Полная форма оператора условного перехода **if**:

```
if (условие) оператор1;  
else оператор2;
```

где *условие* – это некоторое условное выражение, а *оператор1* и *оператор2* – адресаты операторов **if** и **else**. Оператор **else** не является обязательным. Если результат условия равен **true** (условие выполняется), то выполняется *оператор1*. В противном случае выполняется *оператор2*.

После **if** и **else** может быть несколько операторов. В этом случае они берутся в операторные скобки:

```
if (условие)  
{  
    // несколько операторов  
    ...  
}  
else  
{  
    // несколько операторов  
    ...  
}
```

3. Какой вид имеет сокращенная форма оператора условного перехода?

Сокращенная форма оператора условного перехода **if**:

```
if (условие) оператор;
```

В этом случае *оператор* выполняется, если выполняется *условие* (равно **true**). Если значение условия равно **false**, то ничего не выполняется.

Если, после выполнения условия, нужно выполнить несколько операторов, то эти операторы берутся в фигурные скобки:

```
if (условие)  
{  
    // несколько операторов  
}
```

4. Примеры использования оператора **if**, который имеет полную форму представления.

Пример 1. Дано два числа **x** и **y**. Написать фрагмент кода, который находит максимальное значение между этими числами.

```
double x, y, max;  
x = 5;  
y = -3;  
if (x > y)  
    max = x;  
else
```

```
max = y;
```

```
// после выполнения max = 5
```

Пример 2. Даны два вещественных числа x и y . Определить x/y предусмотрев возможное деление на нуль с выводом сообщения «Деление на нуль».

```
double x, y, res;
```

```
x = 5;
```

```
y = 0;
```

```
// проверка деления на нуль
```

```
if (y == 0)
```

```
    label1.Text = "Деление на нуль";
```

```
else
```

```
{
```

```
    res = x / y;
```

```
    label1.Text = res.ToString();
```

```
}
```

Пример 3. Определить, попадает ли значение целой величины a в отрезок $[x; y]$, где x, y – целые числа.

...

```
double x, y, a;
```

```
x = 3;
```

```
y = 9;
```

```
a = 8;
```

```
if ((x<=a)&&(a<=y))
```

```
    label1.Text = "Попадает";
```

```
else
```

```
    label1.Text = "Не попадает";
```

...

5. Примеры использования оператора if, имеющего сокращенную форму представления.

Пример 1. Задача из пункта 4. Даны два числа x и y . Написать фрагмент кода, который находит максимальное значение между этими числами.

```
double x, y, max;
```

```
x = 33;
```

```
y = 9;
```

```
// другой метод поиска максимального значения
```

```
max = x;
```

```
if (max < y) max = y; // сокращенная форма оператора if
```

```
label1.Text = max.ToString();
```

Пример 2. Дано вещественное число x . Найти максимальное значение между выражениями:

- $3 + \sin(x)$
- $1 + x$
- $4 - x^2$

Фрагмент кода:

```
double x, r1, r2, r3;
```

```
double max; // максимальное значение
```

```

x = 2;
r1 = 3 + Math.Sin(x);
r2 = 1 + x;
r3 = 4 - x * x;
max = r1;
if (max < r2) max = r2; // оператор if
if (max < r3) max = r3;
label1.Text = max.ToString();

```

6. Примеры применения вложенных операторов if.

Оператор if-else может содержать другой оператор if-else.

Пример 1. Применение вложенных операторов в полной форме оператора if.

Написать фрагмент кода, который выводит в нисходящем порядке значения трех переменных *x*, *y*, *z*.

...

```

double x, y, z;
x = 22;
y = 88;
z = 2;

// вложенные операторы if
if (x > y)
{
    if (x > z)
    {
        if (y > z) // x - y - z
            label1.Text = x.ToString()+";"+y.ToString()+"; "+z.ToString();
        else // x - z - y
            label1.Text = x.ToString()+";"+z.ToString()+"; "+y.ToString();
    }
    else // z - x - y
        label1.Text = z.ToString()+";"+x.ToString()+"; "+y.ToString();
}
else
{
    if (y > z)
    {
        if (x > z) // y - x - z
            label1.Text = y.ToString()+";"+x.ToString()+"; "+z.ToString();
        else // y - z - x
            label1.Text = y.ToString()+"; "+z.ToString()+"; "+x.ToString();
    }
    else // z - y - x
        label1.Text = z.ToString()+"; "+y.ToString()+"; "+x.ToString();
}

```

Пример 2. Применение вложенных операторов в сокращенной форме оператора if. Решение примера 1 другим способом.

...

```

double x, y, z;

x = 7;
y = 6;
z = 8;

```

// полная + сокращенная форма оператора if + вложенные операторы

```
if (x > y)
{
    if (y > z) // z - y - z
        label1.Text = x.ToString() + ";" + y.ToString() + ";" + z.ToString();
    if (y <= z)
    {
        if (x > z) // x - z - y
            label1.Text = x.ToString() + ";" + z.ToString() + ";" + y.ToString();
        if (x <= z) // z - x - y
            label1.Text = z.ToString() + ";" + x.ToString() + ";" + y.ToString();
    }
}
else
{
    if (y < z) // z - y - z
        label1.Text = z.ToString() + ";" + y.ToString() + ";" + x.ToString();
    if (y >= z)
    {
        if (x > z) // y - x - z
            label1.Text = y.ToString() + ";" + x.ToString() + ";" + z.ToString();
        if (x <= z) // y - z - x
            label1.Text = y.ToString() + ";" + z.ToString() + ";" + x.ToString();
    }
}
```

...

7. Какой вид и принцип работы конструкции if-else-if?

Часто в программах целесообразным есть применение многоступенчатой конструкции

if - else - if

которая состоит из вложенных операторов if.

Общая форма конструкции:

```
if (условие1)
    оператор1;
else if (условие2)
    оператор2;
else if (условие3)
    оператор3;
...
else
    операторN;
```

В такой конструкции условные выражения исчисляются по принципу вниз. Выполнение осуществляется до тех пор, пока не выполнится одно из условий.

Если *условие1* равно *true*, то выполняется *оператор1*. Все другие операторы пропускаются. Аналогично, если истинно *условие2*, то выполняется *оператор2*.

Если ни одно из условий не выполняется, то выполняется *операторN*.

8. Пример применения конструкции if-else-if.

Дано целое число *n*, указывающее номер дня недели от 1 до 7. По указанному номеру *n* вывести название соответствующего дня недели.

Фрагмент кода, решающего данную задачу с использованием конструкции if-else-if:

```

int n;

n = 3;
if (n == 1) label1.Text = "Понедельник";
else
if (n == 2) label1.Text = "Вторник";
else
if (n == 3) label1.Text = "Среда";
else
if (n == 4) label1.Text = "Четверг";
else
if (n == 5) label1.Text = "Пятница";
else
if (n == 6) label1.Text = "Суббота";
else label1.Text = "Воскресенье";

```

Оператор выбора **switch**

1. Какое применение в программе имеет оператор выбора **switch**?

Оператор выбора **switch** обеспечивает многонаправленное ветвление в программе. Этот оператор позволяет сделать выбор среди нескольких альтернативных вариантов дальнейшего выполнения программы.

В некоторых случаях оператор выбора **switch** может иметь более эффективное использование чем использование вложенных операторов **if**.

2. Какой общий вид имеет оператор выбора **switch**?

Общий вид оператора **switch**:

```

switch (выражение)
{
    case константа1:
        операторы1;
    break;
    case константа2:
        операторы2;
    break;
    ...
    case константаN:
        операторыN;
    break;
    ...
    default:
        операторы;
    break;
}

```

где

- *выражение* – заданное выражение целочисленного (**char**, **byte**, **short**, **int**) типа, перечисления или строчного (**string**) типа;
- *константа1*, *константа2*, ... *константаN* – константы выбора, тип которых может быть совместим с типом выражения. Среди констант выбора не должно быть двух с одинаковыми значениями;
- *оператор1*, *оператор2*, ..., *операторN*, *оператор* – последовательность операторов, которые выполняются в случае, если значение константы выбора совпадет со значением выражения.

Если ни одна из констант выбора не совпадает с заданным выражением, то выполняются операторы, которые следуют за словом **default**.

3. Какое назначение блока **default** в операторе **switch**?

Операторы, указанные после слова **default**, выполняются в том случае, если значение выражения не совпадает ни с одной из констант. Блок (ветвь) **default** не является обязательной.

В случае, если блок **default** отсутствует в операторе **switch** и ни одна из констант не совпадает со значением выражения, то никаких действий не выполняется.

4. Пример использования оператора **switch** без использования блока **default**.

Дано целое число *n*, указывающее номер дня недели от 1 до 7. По указанному номеру *n* вывести название соответствующего дня недели.

Фрагмент кода, который решает данную задачу с использованием оператора **switch**:

```
int n;
```

```
n = 7;
```

```
switch (n)
```

```
{
    case 1: label1.Text = "Понедельник";
    break;
    case 2: label1.Text = "Вторник";
    break;
    case 3: label1.Text = "Среда";
    break;
    case 4: label1.Text = "Четверг";
    break;
    case 5: label1.Text = "Пятница";
    break;
    case 6: label1.Text = "Суббота";
    break;
    case 7: label1.Text = "Воскресенье";
    break;
}
```

5. Пример использования оператора **switch** с использованием блока **default**.

Написать фрагмент программного кода, который за заданным номером месяца *n* определяет количество дней в этом месяце. Количество дней заносится в переменную *k*.

```
...
int n, k;
```

```
...
switch (n)
```

```
{
    case 2:
        k = 28;
    break;
    case 4: case 6: case 9: case 11:
        k = 30;
    break;
    default:
        k = 31;
    break;
}
```

```
...
```

6. Пример программы, использующей вложенный оператор **switch**

Оператор **switch** может быть вложен в другой управляющий оператор, которым может быть:

- оператор условного перехода **if**;
- оператор выбора **switch**;
- оператор цикла **for**;
- оператор цикла **while**;
- оператор цикла **do..while**;

- оператор цикла `foreach`.

Количество уровней вложения неограничено. Значит, оператор `switch` может быть размещен на произвольном уровне вложения.

В примере, в операторе `switch` размещаются вложенный оператор `if` и вложенный оператор `switch`. Программа вычисляет количество дней в месяце исходя из введенного значения номера месяца `month` и года `year`. Учитывается тот фактор, что год может быть високосным.

Текст программы, созданной по шаблону [Console Application](#), следующий

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication8
{
    class Program
    {
        static void Main(string[] args)
        {
            // Вложенный оператор switch
            int year; // номер года
            int month; // месяц в году
            int days; // количество дней в году

            // ввести номер года
            Console.Write("Year = ");
            year = Convert.ToInt32(Console.ReadLine());
            Console.Write("Month = ");
            month = Convert.ToInt32(Console.ReadLine());

            // оператор switch, содержащий другой вложенный оператор switch
            switch (month)
            {
                case 4:
                case 6:
                case 9:
                case 11:
                    days = 30;
                    break;
                case 2:
                    bool f; // флажок, определяющий високосный год

                    // вложенный оператор if
                    if (year % 400 == 0)
                        f = true;
                    else
                        if (year % 100 == 0)
                            f = false;
                        else
                            if (year % 4 == 0)
                                f = true;
                            else
                                f = false;

                    // вложенный оператор switch
```

```

        switch (f)
        {
            case true:
                days = 29;
                break;
            default:
                days = 28;
                break;
        }
        break;
    default:
        days = 31;
        break;
    }
    Console.WriteLine("Days = {0}", days);
}
}
}

```

Результат выполнения программы

Year = 2100

Month = 2

Days = 28

7. Применение строк в операторе **switch**. Пример

Кроме целочисленных данных в операторе **switch** могут использоваться строки символов.

Пример. В примере по названию месяца выводится количество дней в этом месяце. Принимается, что в феврале 28 дней.

using System;

```

namespace ConsoleApp12
{
    class Program
    {
        static void Main(string[] args)
        {
            // Использование строк в операторе выбора switch
            string month; // название месяца

            // 1. Ввести название месяца
            month = Console.ReadLine();

            // 2. Вывести количество дней в месяце
            switch (month)
            {
                case "January":
                case "March":
                case "May":
                case "July":
                case "August":
                case "October":
                case "December":
                    Console.WriteLine("Number of days = 31");
                    break;
                case "February":
                    Console.WriteLine("Number of days = 28");

```



```
        break;
    case "April":
    case "June":
    case "September":
    case "November":
        Console.WriteLine("Number of days = 30");
        break;
    default:
        Console.WriteLine("Incorrect input");
        break;
    }
}
}
```