

## Лабораторная работа №2

### Разработка приложений с разветвляющимися и циклическими алгоритмами

#### Блок или составной оператор

С помощью фигурных скобок несколько операторов языка можно объединить в единую синтаксическую конструкцию, называемую блоком или составным оператором:

```
{
    оператор_1;
    ...
    оператор_N;
}
```

В языке C# для выбора одной из нескольких возможностей используются две конструкции - `if` и `switch`. Первую из них обычно называют альтернативным выбором, вторую - разбором случаев.

#### Оператор `if`

Начнем с синтаксиса оператора `if`:

```
if (выражение 1)
    оператор 1;
[else if (выражение 2)
    оператор 2;
...
else if (выражение K)
    оператор K;
else
    оператор_N;]
```

Выражения **if** должны заключаться в круглые скобки и быть булевого типа. Точнее, выражения должны давать значения *true* или *false*. Then-ветвь оператора следует сразу за круглой скобкой без ключевого слова `then`, типичного для большинства языков программирования. Каждый из операторов может быть блоком.

Ветви **else** и **if**, позволяющие организовать выбор из многих возможностей, могут отсутствовать. Может быть опущена и заключительная `else`-ветвь. В этом случае краткая форма оператора **if** задает альтернативный выбор - делать или не делать - выполнять или не выполнять `then`-оператор.

Семантика оператора **if** проста и понятна. Выражения **if** проверяются в порядке их написания. Как только получено значение **true**, проверка прекращается и выполняется оператор (это может быть блок), который следует за выражением, получившим значение **true**. С завершением этого оператора завершается и оператор **if**. Ветвь **else**, если она есть, относится к ближайшему открытому **if**.

Пример:

```
bool condition = true;
if (condition)
{
    Console.WriteLine("The variable is set to true.");
}
```

```

    }
    else
    {
        Console.WriteLine("The variable is set to false.");
    }
}

```

## Оператор switch

Частным, но важным случаем выбора из нескольких вариантов является ситуация, при которой выбор варианта определяется значениями некоторого выражения.

Соответствующий оператор С# называется оператором **switch**. Вот его синтаксис:

```

switch (выражение)
{
    case константное выражение 1: [операторы 1]; оператор перехода_1;
    case константное выражение K: [операторы K]; оператор перехода_K;
    [default: [операторы_N]; оператор перехода_N;]
}

```

В каждой метке **case** указывается постоянное значение (*константное\_выражение*). Управление передается разделу оператора **switch**, метка **case** которого содержит постоянное значение, соответствующее значению выражения **switch**.

Выполнение списка операторов в выбранном разделе начинается с первого оператора и продолжается по списку, обычно до достижения оператора перехода, такого как **break**, **goto case**, **return** или **throw**. В этой точке управление передается за пределы оператора **switch** или к другой метке **case**. В языке С# синтаксически каждая case-ветвь должна заканчиваться оператором перехода, иначе возникнет ошибка периода компиляции.

Если метка **case** не содержит соответствующего значения, элемент управления передается в раздел **default**, если таковой имеется. Если раздела **default** нет, действия не предпринимаются и элемент управления передается за пределы оператора **switch**. Заметьте, по синтаксису допустимо, чтобы после двоеточия следовала пустая последовательность операторов, заканчивающаяся оператором перехода.

В синтаксической конструкции **switch** есть ограничения, накладываемые на **case** выражения, которые могут быть только константным выражением. Когда разбор случаев предполагает проверку попадания в некоторый диапазон значений, приходится прибегать к оператору **if** для формирования специальной переменной. Константные выражения в **case** должны иметь тот же тип, что и **switch**-выражение.

Пример:

```

int caseSwitch = 1;
switch (caseSwitch)
{
    case 1:
        Console.WriteLine("Case 1");
        break;
    case 2:

```

```

        Console.WriteLine("Case 2");
        break;
    default:
        Console.WriteLine("Default case");
        break;
}

```

## Операторы перехода

Операторы перехода, позволяют прервать естественный порядок выполнения операторов блока, в языке C# их несколько.

В структурном программировании признаются полезными "переходы вперед" (но не назад), позволяющие при выполнении некоторого условия выйти из цикла, из оператора выбора, из блока. Для этой цели можно использовать оператор **goto**, но лучше применять специально предназначенные для этих целей операторы **break** и **continue**.

Оператор **break** завершает ближайший внешний цикл или оператор **switch**, в котором он появляется. Управление передается оператору, следующему за завершенным оператором (если таковой имеется).

Оператор **continue** используется только в теле цикла. В отличие от оператора **break**, завершающего внутренний цикл, **continue** осуществляет переход к следующей итерации этого цикла.

Оператор **return**, позволяет завершить выполнение процедуры или функции.

## Циклы

Циклы являются управляющими конструкциями, позволяя в зависимости от определенных условий выполнять некоторое действие множество раз. В C# имеются следующие виды циклов:

- for
- foreach
- while
- do...while

### Цикл for

Цикл for имеет следующее формальное определение:

```

for ([инициализация счетчика]; [условие]; [изменение счетчика])
{
    // действия
}

```

Рассмотрим стандартный цикл for:

```

for (int i = 0; i < 9; i++)
{
    Console.WriteLine($"Квадрат числа {i} равен {i*i}");
}

```

Первая часть объявления цикла - *int i = 0* - создает и инициализирует счетчик *i*. Счетчик необязательно должен представлять тип *int*. Это может быть и другой числовой тип, например, *float*. И перед выполнением цикла его

значение будет равно 0. В данном случае это то же самое, что и объявление переменной.

Вторая часть - условие, при котором будет выполняться цикл. Пока условное выражение возвращает *true*, будет выполняться цикл. В данном случае цикл будет выполняться, пока счетчик *i* не достигнет 9.

И третья часть - приращение счетчика на единицу. Опять же нам необязательно увеличивать на единицу. Можно уменьшать: *i--*.

В итоге блок цикла сработает 9 раз, пока значение *i* не станет равным 9. И каждый раз это значение будет увеличиваться на 1.

Нам необязательно указывать все условия при объявлении цикла. Например, мы можем написать так:

```
int i = 0;
for (; ; )
{
    Console.WriteLine($"Квадрат числа {++i} равен {i * i}");
}
```

Формально определение цикла осталось тем же, только теперь блоки в определении у нас пустые: *for (; i <;)*. У нас нет инициализированной переменной-счетчика, нет условия, поэтому цикл будет работать вечно - бесконечный цикл.

Мы также можем опустить ряд блоков:

```
int i = 0;
for (; i < 9;)
{
    Console.WriteLine($"Квадрат числа {++i} равен {i * i}");
}
```

Этот пример, по сути, эквивалентен первому примеру: у нас также есть счетчик, только создан он вне цикла. У нас есть условие выполнения цикла. И есть приращение счетчика уже в самом блоке *for*.

### Цикл **do**

В цикле *do* сначала выполняется код цикла, а потом происходит проверка условия в инструкции *while*. И пока это условие истинно, цикл повторяется. Например:

```
int i = 6;
do
{
    Console.WriteLine(i);
    i--;
}
while (i > 0);
```

Здесь код цикла сработает 6 раз, пока *i* не станет равным нулю. Но важно отметить, что цикл *do* гарантирует хотя бы единократное выполнение действий, даже если условие в инструкции *while* не будет истинно. То есть мы можем написать:

```
int i = -1;
do
{
    Console.WriteLine(i);
}
```

```
        i--;  
    }  
    while (i > 0);
```

Хотя у нас переменная *i* меньше 0, цикл все равно один раз выполнится.

### Цикл *while*

В отличие от цикла *do* цикл *while* сразу проверяет истинность некоторого условия, и если условие истинно, то код цикла выполняется:

```
int i = 6;  
while (i > 0)  
{  
    Console.WriteLine(i);  
    i--;  
}
```

### Операторы *continue* и *break*

Иногда возникает ситуация, когда требуется выйти из цикла, не дожидаясь его завершения. В этом случае мы можем воспользоваться оператором *break*.

Например:

```
for (int i = 0; i < 9; i++)  
{  
    if (i == 5)  
        break;  
    Console.WriteLine(i);  
}
```

Хотя в условии цикла сказано, что цикл будет выполняться, пока счетчик *i* не достигнет значения 9, в реальности цикл сработает 5 раз. Так как при достижении счетчиком *i* значения 5, сработает оператор *break*, и цикл завершится.

Теперь поставим себе другую задачу. А что, если мы хотим, чтобы при проверке цикл не завершался, а просто пропускал текущую итерацию. Для этого мы можем воспользоваться оператором *continue*:

```
for (int i = 0; i < 9; i++)  
{  
    if (i == 5)  
        continue;  
    Console.WriteLine(i);  
}
```

## Задания на лабораторную работу

### Задание 1

- Даны два различных вещественных числа. Определить:
  - какое из них больше;
  - какое из них меньше.
- Определить максимальное и минимальное значения из двух различных вещественных чисел.
- Известны два расстояния: одно в километрах, другое — в футах (1 фут 0,305 м). Какое из расстояний меньше?

4. Известны две скорости: одна в километрах в час, другая — в метрах в секунду. Какая из скоростей больше?

5. Даны радиус круга и сторона квадрата. У какой фигуры площадь больше?

6. Даны объемы и массы двух тел из разных материалов. Материал какого из тел имеет большую плотность?

7. Известны сопротивления двух несоединенных друг с другом участков электрической цепи и напряжение на каждом из них. По какому участку протекает меньший ток?

8. Даны вещественные числа  $a$ ,  $b$ ,  $c$  ( $a \neq 0$ ). Выяснить, имеет ли уравнение  $ax^2 + bx + c = 0$  вещественные корни.

9. Для условий предыдущей задачи в случае наличия вещественных корней найти их, в противном случае — вывести на экран соответствующее сообщение. Вариант равенства корней отдельно не рассматривать.

10. Известны год и номер месяца рождения человека, а также год и номер месяца сегодняшнего дня (январь — 1 и т. д.). Определить возраст человека (число полных лет). В случае совпадения указанных номеров месяцев считать, что прошел полный год.

11. Известны площади круга и квадрата. Определить:

а) уместится ли круг в квадрате?

б) уместится ли квадрат в круге?

12. Известны площади круга и равностороннего треугольника. Определить:

а) уместится ли круг в треугольнике?

б) уместится ли треугольник в круге?

## Задание 2

1. Определить, является ли число  $a$  делителем числа  $b$ ?

2. Дано натуральное число. Определить:

а) является ли оно четным;

б) оканчивается ли оно цифрой 7.

3. Дано двузначное число. Определить:

а) какая из его цифр больше: первая или вторая;

б) одинаковы ли его цифры.

4. Дано двузначное число. Определить, равен ли квадрат этого числа учетверенной сумме кубов его цифр. Например, для числа 48 ответ положительный, для числа 52 — отрицательный.

5. Дано двузначное число. Определить:

а) является ли сумма его цифр двузначным числом;

б) больше ли числа  $a$  сумма его цифр.

6. Дано двузначное число. Определить:

- а) кратна ли трем сумма его цифр;
  - б) кратна ли сумма его цифр числу  $a$ .
7. Дано трехзначное число. Выяснить, является ли оно палиндромом ("перевертышем"), т. е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево.
8. Дано трехзначное число. Определить, какая из его цифр больше:
- а) первая или последняя;
  - б) первая или вторая;
  - в) вторая или последняя.
9. Дано трехзначное число. Определить, равен ли квадрат этого числа сумме кубов его цифр.
10. Дано трехзначное число. Определить:
- а) является ли сумма его цифр двузначным числом;
  - б) является ли произведение его цифр трехзначным числом;
  - в) больше ли числа  $a$  произведение его цифр;
  - г) кратна ли пяти сумма его цифр;
  - д) кратна ли сумма его цифр числу  $a$ .
11. Дано трехзначное число.
- а) Верно ли, что все его цифры одинаковые?
  - б) Определить, есть ли среди его цифр одинаковые.
12. Дано четырехзначное число. Определить:
- а) равна ли сумма двух первых его цифр сумме двух его последних цифр;
  - б) кратна ли трем сумма его цифр;
  - в) кратно ли четырем произведение его цифр;
  - г) кратно ли произведение его цифр числу  $a$ .

### Задание 3.

1. Даны три вещественных числа  $a, b, c$ . Проверить:
- а) выполняется ли неравенство  $a < b < c$ ;
  - б) выполняется ли неравенство  $b > a > c$ .
2. Определить, является ли число  $a$  делителем числа  $b$  или, наоборот, число  $b$  делителем числа  $a$ . Ответом должны служить сообщения "Да, одно из чисел является делителем другого" или "Нет, ни одно из чисел не является делителем другого".
3. Определить, верно ли, что при делении неотрицательного целого числа  $a$  на положительное число  $b$  получается остаток, равный одному из двух заданных чисел  $c$  или  $d$ .
4. Даны три вещественных числа  $a, b, c$ . Определить, имеется ли среди них хотя бы одна пара равных между собой чисел.
5. Определить, является ли треугольник со сторонами  $a, b, c$  равносторонним.

6. Определить, является ли треугольник со сторонами  $a$ ,  $b$ ,  $c$  равнобедренным.

7. Известен рост трех человек. Определить, одинаков ли их рост?

8. Даны вещественные положительные числа  $a$ ,  $b$ ,  $c$ ,  $d$ . Выяснить, можно ли прямоугольник со сторонами  $a$ ,  $b$  уместить внутри прямоугольника со сторонами  $c$ ,  $d$  так, чтобы каждая из сторон одного прямоугольника была параллельна или перпендикулярна каждой стороне второго прямоугольника.

9. Определить, войдет ли в конверт с внутренними размерами  $a$  и  $b$  мм прямоугольная открытка размером  $c$  и  $d$  мм. Для размещения открытки в конверте необходим зазор в 1 мм с каждой стороны.

10. Вася пытается высунуть голову в форточку размерами  $a$  и  $b$  см. Приняв условно, что его голова — круглая диаметром  $d$  см, определить, сможет ли Вася сделать это. Для прохождения головы в форточку необходим зазор в 1 см с каждой стороны.

11. Даны вещественные положительные числа  $a$ ,  $b$ ,  $c$ ,  $x$ ,  $y$ . Выяснить, пройдет ли кирпич с ребрами  $a$ ,  $b$ ,  $c$  в прямоугольное отверстие со сторонами  $x$  и  $y$ . Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.

12. Дано двузначное число. Определить:

- а) входит ли в него цифра 3;
- б) входит ли в него цифра  $a$ .

#### **Задание 4.**

1. Дано натуральное число. Определить:

- а) количество цифр 3 в нем;
- б) сколько раз в нем встречается последняя цифра;

2. Дано натуральное число. Определить:

- а) количество четных цифр в нем. Составное условие и более одного неполного условного оператора не использовать;
- б) сумму его цифр, больших пяти;

3. Дано натуральное число. Определить:

- а) произведение его цифр, больших семи;
- б) сколько раз в нем встречаются цифры 0 и 5 (всего).

4. Дано натуральное число. Определить:

- а) сколько раз в нем встречается цифра  $a$ ;
- б) количество его цифр, кратных  $z$  (значение  $z$  вводится с клавиатуры)

5. Дано натуральное число. Определить:

- а) сумму его цифр, больших  $a$  (значение  $a$  вводится с клавиатуры)
- б) сколько раз в нем встречаются цифры  $x$  и  $y$ .

6. Дана непустая последовательность целых чисел (организовать ввод значений в цикле), оканчивающаяся нулем. Найти:



а) сумму всех чисел последовательности, больших числа  $x$ ;  
б) количество всех четных чисел последовательности. 7. Дана последовательность ненулевых целых чисел (организовать ввод значений в цикле), оканчивающая нулем. Определить, сколько раз в этой последовательности меняется знак. (Например, в последовательности 10, -4, 12, 56, -4 знак меняется 3 раза.)

8. Дано натуральное число. Определить его максимальную цифру.

9. Дано натуральное число. Определить его минимальную цифру.

10. Дано натуральное число. Определить номер цифры 8 в нем, считая от конца числа. Если такой цифры нет, ответом должно быть число 0, если таких цифр в числе несколько — должен быть определен номер самой левой из них.

11. Напечатать все кратные тринадцати натуральные числа, меньшие 100.

12. Найти 15 первых натуральных чисел, делящихся нацело на 19 и находящихся в интервале, левая граница которого равна 100.

13. Найти 20 первых натуральных чисел, делящихся нацело на 13 или на 17 и находящихся в интервале, левая граница которого равна 500.

14. Найти 10 первых натуральных чисел, оканчивающихся на цифру 7, кратных числу 9 и находящихся в интервале, левая граница которого равна 100.

#### **Задание 5.**

1. Дано натуральное число. Верно ли, что сумма его цифр больше 10?

2. Дано натуральное число. Верно ли, что произведение его цифр меньше 50?

3. Дано натуральное число. Верно ли, что количество его цифр есть четное число?

4. Дано натуральное число. Верно ли, что это число четырехзначное?

5. Дано натуральное число.

а) Верно ли, что его первая цифра не превышает 6?

б) Верно ли, что оно начинается и заканчивается одной и той же цифрой?

6. Дано натуральное число. Определить, какая из его цифр больше: первая или последняя.

7. Дано натуральное число. Верно ли, что сумма его цифр меньше  $a$ ?

8. Дано натуральное число. Верно ли, что произведение его цифр больше  $b$ ?

9. Дано натуральное число.

а) Верно ли, что это число  $k$ -значное? Составное условие и вложенный условный оператор не использовать.

б) Верно ли, что его первая цифра превышает  $m$ ?

10. Дано натуральное число.

а) Верно ли, что сумма его цифр больше  $k$ , а само число четное?

б) Верно ли, что количество его цифр есть четное число, а само число не превышает  $b$ ?

11. Дано натуральное число.

а) Верно ли, что оно начинается цифрой  $x$  и заканчивается цифрой  $y$ ?

б) Верно ли, что произведение его цифр меньше  $a$ , а само число делится на  $b$ ?

12. Дано натуральное число. Определить:

а) Верно ли, что сумма его цифр больше  $m$ , а само число делится на  $n$ ?

б) есть ли в нем цифры 2 и 5.

13. Дано натуральное число.

а) Определить, есть ли в нем цифра  $a$ .

б) Верно ли, что в нем нет цифры  $b$ ?

14. Дано натуральное число.

в) Верно ли, что цифра  $a$  встречается в нем более  $k$  раз?

г) Определить, есть ли в нем цифры  $a$  и  $b$ .

### **Задание 6.**

1. Вывести на экран все целые числа от 100 до 200, кратные трем. 2. Вывести на экран все целые числа от  $a$  до  $b$ , кратные некоторому числу  $c$ .

3. Найти сумму положительных нечетных чисел, меньших 50.

4. Найти сумму целых положительных чисел из промежутка от  $a$  до  $b$ , кратных четырем.

5. Составить программу поиска трехзначных чисел, которые при делении на 47 дают в остатке 43, а при делении на 43 дают в остатке 47.

6. Составить программу поиска четырехзначных чисел, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.

7. Определить количество натуральных чисел из интервала от 100 до 500, сумма цифр которых равна 15.

8. Определить количество трехзначных натуральных чисел, сумм цифр которых равна целому числу  $n$  ( $0 < n < 27$ ).

9. Найти все двузначные числа, которые делятся на  $n$  или содержат цифру  $n$ .

10. Найти сумму целых положительных чисел, больших 30 и меньших 100, кратных трем и оканчивающихся на 2, 4 и 8.

11. Дано натуральное число. Получить все его делители.

12. Дано натуральное число. Найти сумму его делителей.

13. Дано натуральное число. Найти сумму его четных делителей.

14. Дано натуральное число. Определить количество его делителей.