**Циклы в Си-шарп. Операторы break и continue**

**Циклы** служат для многократного повторения некоторого фрагмента кода.  
В Си-шарп есть четыре оператора циклов: for, while, do-while, foreach.  
  
**Цикл for**  
  
Этот цикл используется тогда, когда наперед известно, сколько повторений нужно сделать. Он имеет следующую структуру:

for (инициализация счетчика; условие продолжения; итерация)  
{  
   //блок кода, который будет повторяться  
}

Пример программы, которая выводит на экран числа 0, 1, 2, 3, 4:

static void Main(string[] args)  
{  
   for (int i = 0; i < 5; i++) // цикл выполнится 5 раз  
   {  
      Console.WriteLine(i);  
   }  
}

Сначала происходит создание и инициализация счетчика, i=0. Дальше идет проверка условия ( i < 5), если результат будет «истина», то дальше выполняется блок кода в теле цикла. В конце итерации происходит изменение значения счетчика (в данном примере увеличение на единицу). После этого вновь происходит проверка условия и так далее. Когда условие будет «ложь», цикл работу завершит.  
  
Пример программы, которая находит и выводит на экран сумму элементов массива:

static void Main(string[] args)  
{  
   int[] numbers = { 4, 7, 1, 23, 43 };  
   int s = 0;  
   for (int i = 0; i < numbers.Length; i++)  
   {  
      s += numbers[i];  
   }  
   Console.WriteLine(s);  
   Console.ReadKey();  
}

Пример цикла for, когда счетчик уменьшается после каждой итерации:

for (int i = 5; i > 0; i--) //выполнится 5 раз  
{  
   Console.WriteLine(i);  
}

Счетчик можно изменять не только на единицу. Пример программы, которая выводит чётные числа (по число 50):

for (int i = 0; i <= 50; i+=2) //выполнится 26 раз  
{  
   Console.WriteLine(i);  
}

**Цикл while**  
  
Слово while переводится, как «пока», что хорошо его характеризует. Он продолжает выполнятся до тех пор, пока «истинно» некоторое условие. Он имеет такую структуру:

while (условие продолжения)  
{  
   //блок кода, который будет повторяться  
}

Сначала проверяется условие, а дальше выполняется блок кода.  
  
Пример той же программы, которая выводит на экран числа 0, 1, 2, 3, 4:

int i = 0;  
while (i < 5)  
{  
   Console.WriteLine(i);  
   i++;  
}

Цикл может выполнятся «вечно», если задать всегда истинное условие:

while (true)  
{  
   Console.WriteLine("Вечный цикл");  
}

**Цикл do-while**  
  
Этот тот же цикл while, только здесь сначала выполняется блок кода, а уже потом идет проверка условия. Это гарантирует хотя бы один проход цикла.

do  
{  
//блок кода, который будет повторяться  
}  
while (условие продолжения);

Пример программы, которая не завершит работу, пока с клавиатуры не введут число 5:

static void Main(string[] args)  
{  
   int number;  
   do  
   {  
      Console.WriteLine("Введите число 5");  
      number = Convert.ToInt32(Console.ReadLine());  
   }  
   while (number != 5);  
}

**Оператор break**  
  
Из любого цикла можно досрочно выйти, используя оператор break. Использовать данный оператор есть смысл после удовлетворения некоторого условия, иначе цикл завершится на первой итерации.  
  
Пример программы, которая проверяет, есть ли в массиве число кратное 13-ти. Найдя такое число, нет смысла дальше проверять остальные элементы массива, и здесь мы используем оператор break:

static void Main(string[] args)  
{  
   int[] numbers = { 4, 7, 13, 20, 33, 23, 54 };  
   bool b = false;  
   for (int i = 0; i < numbers.Length; i++)  
   {  
      if (numbers[i] % 13 == 0)  
      {  
         b = true;  
         break;  
      }  
   }  
   Console.WriteLine(b ? "В массиве есть число кратное 13" : "В массиве нет числа кратного 13");  
   Console.ReadKey();  
}

**Оператор continue**  
  
Данный оператор позволяет перейти к следующей итерации, не завершив до конца текущую.  
  
Пример программы, которая находит сумму нечетных элементов массива:

static void Main(string[] args)  
{  
   int[] numbers = { 4, 7, 13, 20, 33, 23, 54 };  
   int s = 0;  
   for (int i = 0; i < numbers.Length; i++)  
   {  
      if (numbers[i] % 2 == 0)  
         continue; //переход к следующей итерации  
      s += numbers[i];  
      }  
   Console.WriteLine(s);  
   Console.ReadKey();  
}

**Домашнее задание**  
  
**Задача 1.** Протестировать представленные выше операторы и предоставить отчет о результатах работы.

**Задача 2**. Вывести на экран 20 элементов последовательности 1, 4, 7, 10, 13…  
**Задача 3**. Напишите программу, которая будет «спрашивать» правильный пароль, до тех пор, пока он не будет введен. Правильный пароль пусть будет «root».  
**Задача 4**. Дано два массива одинаковой длины (по 10 элементов). Создайте третий массив, который будет отображать сумму первых двух массивов. Первый элемент третьего массива равен сумме первых элементов двух первых массивов и так далее.