

## Практическая работа №12

### Ввод и обработка последовательностей.

Цель работы: сформировать умения по использованию процедурного языка программирования для построения логически правильных и эффективных программ с итерационными циклами.

Норма времени: 2 часа.

Оборудование: персональный компьютер, Visual Studio 2019.

#### Ход работы

**Последовательности** — это абстрактный набор данных, которые можно перебирать.

В отличие от других структур последовательности не хранятся в памяти (поэтому «абстрактный»), т.е. память используется только в конкретный момент времени, когда происходит перебор значений последовательности, и, как правило, нам необходим только 1 элемент (в единицу времени).

Опишем использование операторов цикла для обработки **числовых последовательностей**.

Алгоритм для обработки последовательностей относится к одному из двух типов: поиск, проверка условий.

Для последовательностей характерно, что в каждый момент времени нам доступен только один элемент **последовательности**. Поэтому все алгоритмы строятся с учетом однократного последовательного просмотра.

Рассмотрим несколько программ. В каждой из них одновременно рассматривается только очередной элемент последовательности, без учета остальных (массивы не используются). Алгоритмы для решения таких задач называются алгоритмами с линейным поиском.

**Пример 1.** Дана непустая последовательность целых чисел, оканчивающаяся нулем. Найти:

- а) сумму всех чисел последовательности;
- б) количество всех чисел последовательности.

```
static void Main(string[] args)
{
    int n=0, s = 0, a=1;
    while (a!=0)
    {
        Console.Write("Введите целое число: ");
        a = Convert.ToInt32(Console.ReadLine());
        s += a;
        n++;
    }
    Console.WriteLine("Сумма = {0}, всего чисел = {1}", s,n);
    Console.ReadKey();
}
```

**Пример 2.** Дана непустая последовательность неотрицательных целых чисел, оканчивающаяся отрицательным числом. Найти среднее арифметическое всех чисел последовательности (без учета отрицательного числа).

```
int n = 0, s = 0, a;
do
{
    Console.WriteLine("Введите целое число: ");
    a = Convert.ToInt32(Console.ReadLine());
    s += a;
    n++;
} while (a >= 0);
double d = s - a;
n--;
d = d / n;
Console.WriteLine("Среднее арифметическое = {0}", d);
```

**Пример 3.** Последовательность Фибоначчи образуется так: первый и второй члены последовательности равны 1, каждый следующий равен сумме двух предыдущих (1, 1, 2, 3, 5, 8, 13, ...). Найти:

а) первое число в последовательности Фибоначчи, большее  $n$  (значение  $n$  вводится с клавиатуры;  $n > 1$ );

б) сумму всех чисел в последовательности Фибоначчи, которые не превосходят 1000.

```
c = Vvod_Chisla_Int("n");
a = b = 1;
do
{
    d = a;
    a += b;
    b = d;
}
while (a <= c);
Console.WriteLine("Ответ а): {0} > {1}", a, c);
a = b = 1;
e = 2;
do
{
    d = a;
    a += b;
    b = d;
    e += a;
}
while (a <= 1000);
e -= a;
Console.WriteLine("Ответ б): {0}", e);
Console.ReadKey();
```

Здесь было показано, как использовать операторы цикла для обработки числовых последовательностей.

### **Задания для самостоятельного решения:**

1. Найти сумму элементов последовательности.
2. Найти минимальный элемент в последовательности.
3. Найти второй по значению элемент последовательности.
4. Определить, сколько раз в последовательности встречается заданное число.
5. Найти произведение элементов последовательности.
6. Найти сумму модулей элементов последовательности.
7. Вычислить сумму квадратов элементов последовательности.
8. Определить среднее арифметическое элементов последовательности.
9. Определить среднее геометрическое элементов последовательности.
10. Найти произведение модулей элементов последовательности.
11. Определить, сколько нулей в последовательности.
12. Определить количество ненулевых элементов последовательности.

### **Контрольные вопросы:**

1. Дайте классификацию циклических процессов с указанием их особенностей.
2. Укажите отличительные особенности каждого из трех операторов цикла.