



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

*Институт Принтмедиа и информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 20

Дисциплина: Введение в программирование.

Выполнил(а):

студент(ка) группы 191-726

Синельникова К.Т.

(Дата)

(Подпись)

Проверил: асс. Кононенко К.М.

(Дата)

(Подпись)

Замечания: _____

Москва

2019

Оглавление

Теория	3
Задания	5
Код программы.....	6
Результат программы.....	10

Теория

Объявление массивов

Массивы объявляются так же, как и другие переменные, при помощи операторов Dim, Static, Private или Public. Отличие скалярных переменных (которые не являются массивами) от переменных массивов заключается в том, что для массива, как правило, необходимо указывать размер. Массив с указанным размером является массивом фиксированного размера. Массив, размер которого можно изменить во время выполнения программы, является динамическим массивом.

Индексация массива от 0 или 1 зависит от оператора Option Base. Если не указано Option Base 1, все индексы массива будут начинаться с нуля.

Объявление статического массива

Первый аргумент определяет количество строк, второй — столбцов.

Как и в случае объявления любой другой переменной, если для объявленного массива не указать тип данных, его элементам будет присвоен тип данных Variant. Каждый числовой элемент Variant массива использует 16 байтов. Каждый строчный элемент Variant использует 22 байта. Чтобы написать как можно более компактный код, четко объявите для своих массивов тип данных, отличный от Variant.

Максимальный размер массивов зависит от операционной системы и доступного объема памяти. Использование массивов, размер которых превышает объем доступной оперативной памяти вашего компьютера, приводит к снижению скорости, поскольку системе необходимо выполнять запись данных и чтение с диска.

Объявление динамического массива

Объявив динамический массив, вы сможете менять его размер во время выполнения кода. Используйте операторы `Static`, `Dim`, `Private` или `Public`, чтобы объявить массив, не указывая значение в скобках. Вы можете неявно объявить массив в процедуре при помощи оператора `ReDim`. Будьте внимательны и вводите имя массива без ошибок при использовании оператора `ReDim`. Даже если в модуль включен оператор `Option Explicit`, будет создан второй массив.

В процедуре внутри области массива используйте оператор `ReDim`, чтобы изменить количество измерений, задать количество элементов и определить нижнюю и верхнюю границы каждого измерения. Вы можете менять динамический массив при помощи оператора `ReDim` в любое время. Однако значения внутри массива при этом не сохраняются. Используйте `ReDim Preserve` для расширения массива, сохраняя при этом текущие значения.

Задания

1. Дан символ C . Вывести два символа, первый из которых предшествует символу C в кодовой таблице, а второй следует за символом C .
2. Дана непустая строка S . Вывести строку, содержащую символы строки S , между которыми вставлено по одному пробелу.
3. Дана строка. Подсчитать количество содержащихся в ней прописных латинских букв.
4. Дан символ C и строка S . Удвоить каждое вхождение символа C в строку S .
5. Даны строки S и S_0 . Найти количество вхождений строки S_0 в строку S .

Код программы

Листинг 1 — Задание 1 (Предшествующий - предыдущий символ)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_20
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             char C;
13.             int L, P, a;
14.             Console.Write("Введите символ: ");
15.             C = Convert.ToChar(Console.ReadLine());
16.             a = Convert.ToInt32(C);
17.             P = a - 1;
18.             L = a + 1;
19.             Console.WriteLine("Предшествующий символ: " + Convert.ToChar(P));
20.             Console.WriteLine("Последующий символ: " + Convert.ToChar(L));
21.             Console.ReadKey();
22.         }
23.     }
24. }
```

Листинг 2 — Задание 2 (Один пробел)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_20
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             List<char> myList = new List<char> { };
13.             string S;
14.             char P;
15.             Console.Write("Введите строку: ");
16.             S = Console.ReadLine();
17.             for (int i = 0; i < S.Length; i++)
18.             {
19.                 P = Convert.ToChar(S[i]);
20.                 myList.Add(P);
21.                 myList.Add(' ');
22.             }
23.             Console.Write("Обработанная строка: ");
24.             for (int i = 0; i < S.Length * 2; i++)
25.             {
26.                 Console.Write(myList[i]);
27.             }
28.             Console.ReadKey();
29.         }
30.     }
31. }
```

Листинг 3 — Задание 3 (Кол-во латинских прописных букв)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_20
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             string S;
13.             int Q, c = 0;
14.             Console.Write("Введите строку: ");
15.             S = Console.ReadLine();
16.             for (int i = 0; i < S.Length; i++)
17.             {
18.                 Q = Convert.ToInt32(S[i]);
19.                 if (Q > 64 && Q < 91)
20.                 {
21.                     c++;
22.                 }
23.             }
24.             Console.Write("Количествово прописных букв: " + c);
25.             Console.ReadKey();
26.         }
27.     }
28. }
```

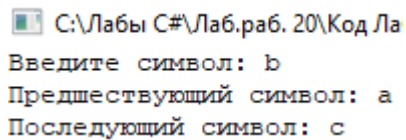
Листинг 4 — Задание 4 (Удвоить вхождение символа)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_20
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             List<char> myList = new List<char> { };
13.             string S;
14.             char h;
15.             char C;
16.             int k = 0;
17.             Console.Write("Введите символ: ");
18.             C = Convert.ToChar(Console.ReadLine());
19.             Console.Write("Введите строку: ");
20.             S = Console.ReadLine();
21.             for (int i = 0; i < S.Length; i++)
22.             {
23.                 h = Convert.ToChar(S[i]);
24.                 if (h == C)
25.                 {
26.                     myList.Add(h);
27.                     myList.Add(h);
28.                     k++;
29.                 }
30.                 else { myList.Add(h); }
31.             }
32.             Console.Write("Обработанная строка: ");
33.             for (int i = 0; i < S.Length + k; i++)
34.             {
35.                 Console.Write(myList[i]);
36.             }
37.             Console.ReadKey();
38.         }
39.     }
40. }
```


Листинг 5 — Задание 5 (Найти количество вхождений)

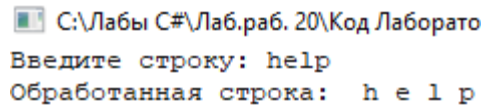
```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_20
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             string s, s0;
13.             Console.Write("Введите строку S: ");
14.             s = Console.ReadLine();
15.             Console.Write("Введите строку S0: ");
16.             s0 = Console.ReadLine();
17.             int k = s.Length;
18.             s = s.Replace(s0, "");
19.             Console.Write("Количество вхождений: " + (k - s.Length) /
20.                 s0.Length);
21.             Console.ReadKey();
22.         }
23.     }
```

Результат программы



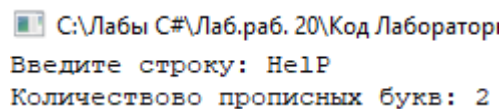
```
C:\Лабы С#\Лаб.раб. 20\Код Ла
Введите символ: b
Предшествующий символ: a
Последующий символ: c
```

Рисунок 6 — Результат выполнения программы 1



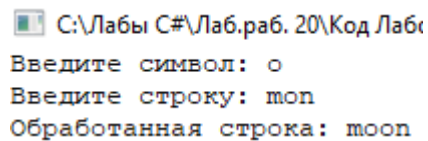
```
C:\Лабы С#\Лаб.раб. 20\Код Лаборато
Введите строку: help
Обработанная строка: h e l p
```

Рисунок 7 — Результат выполнения программы 2



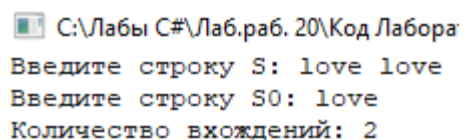
```
C:\Лабы С#\Лаб.раб. 20\Код Лаборатори
Введите строку: He1P
Количество прописных букв: 2
```

Рисунок 8 — Результат выполнения программы 3



```
C:\Лабы С#\Лаб.раб. 20\Код Лаб
Введите символ: o
Введите строку: moп
Обработанная строка: moop
```

Рисунок 9 — Результат выполнения программы 4



```
C:\Лабы С#\Лаб.раб. 20\Код Лабо
Введите строку S: love love
Введите строку S0: love
Количество вхождений: 2
```

Рисунок 10 — Результат выполнения программы 5