**Выполнил:** Козлов Максим Васильевич, группа ИСТб-19-2

# Лабораторное задание №1

## Задача на строки

### Постановка задачи

**Задача №11.** Даны два слова. Для каждой буквы первого слова определить, входит ли она во второе слово. Повторяющиеся буквы первого слова не рассматривать. Например, если заданные слова процессор и информация, то для букв первого из них ответом должно быть: нет да да да нет нет.

**Словесное описание алгоритма**

**Main()**

1. Заводим переменную (first\_str) под первую строку и присваиваем значение функции **GetString()**;
2. Заводим переменную (second\_str) под вторую строку и присваиваем значение функции **GetString()**;
3. Заводим переменную под ответ (answer) и присваиваем значение функции **GetMatches()**;
4. Выведем значение answer на экран;

**GetString()**

1. Заводим переменную flag и присваиваем ей true;
2. Заводим строковую переменную str под возвращаемое функцией значение и делаем её пустой;
3. ПОКА str является пустой ИЛИ flag = false ВЫПОЛНИТЬ
   1. Присваиваем true переменной flag;
   2. Выведем значение message на экран;
   3. Заводим переменную (input) под вводимую строку и присваиваем вводимое пользователем значение;
   4. Ожидаем ввод пользователя;
   5. Заводим массив symbols и присваиваем значение преобразованной в массив символов строки input;
   6. i = 0;
   7. ПОКА i < количество элементов массива symbols ВЫПОЛНИТЬ
      1. ЕСЛИ элемент массива symbols [i] является буквой, ТО
         1. Дополним строку str элементом массива symbols [i];

ИНАЧЕ

* + - 1. Переменной flag присваиваем false;

ЕСЛИ ВСЕ

* + 1. i = i+1;

ПОКА ВСЕ

* 1. ЕСЛИ flag = false, ТО
     1. Выведем на экран “Ваша строка содержит некорректные символы!”;

ЕСЛИ ВСЕ

ПОКА ВСЕ

1. Возвращаем значение переменной str;

**GetMatches()**

1. Заводим строковую переменную matches под возвращаемое функцией значение и делаем её пустой;
2. i=0;
3. ПОКА i <= (количество букв в строке first\_str - 2) ВЫПОЛНИТЬ
   1. j = j+1;
   2. ПОКА j <= (количество букв в строке first\_str – 1) ВЫПОЛНИТЬ
      1. ЕСЛИ буква first\_str [i] совпадает с буквой first\_str [j], ТО
         1. Удалить в строке first\_str символ first\_str [j];

ЕСЛИ ВСЕ

* + 1. j = j+1;

ПОКА ВСЕ

* 1. i=i+1;

ПОКА ВСЕ

1. i=0;
2. ПОКА i <= (количество букв в строке first\_str) ВЫПОЛНИТЬ
   1. ЕСЛИ вторая строка (second\_str) содержит букву первой строки (first\_str [i]), ТО
      1. Дополним строку matches: буква, что подходит по условию (first\_str [i]) + “найдено” + количество совпадений (значение функции **GetMatchesCount()**);

ИНАЧЕ

* + 1. Дополним строку matches: буква, что не подходит по условию (first\_str [i]) + “не найдено совпадений”;

ЕСЛИ ВСЕ

* 1. i=i+1;

ПОКА ВСЕ

1. Возвращаем значение переменной matches;

**GetMatchesCount()**

1. Заводим переменную count под возвращаемое функцией значение;
2. Заводим массив символов str и присваиваем к нему преобразованную в массив символов строку second\_str;
3. i = 0;
4. ПОКА i< количество букв в str ВЫПОЛНИТЬ
   1. ЕСЛИ буква str [i] совпадает с буквой SearchedChar,ТО
      1. count = count+1;

ЕСЛИ ВСЕ

1. Заводим строковую переменную s под возвращаемое функцией значение и делаем её пустой;
2. ЕСЛИ count = 1, ТО
   1. Задаём переменной s: count + “совпадение”;
3. ЕСЛИ count <= 1 И count НЕ РАВЕН 1 И count НЕ РАВЕН 0, ТО
   1. Задаём переменной s: count + “совпадения”;

ИНАЧЕ

* 1. ЕСЛИ count НЕ РАВЕН 1, ТО
     1. Задаём переменной s: count + “совпадений”;
  2. ЕСЛИ ВСЕ

1. ЕСЛИ ВСЕ
2. Возвращаем значение переменной s;

### Таблица спецификаций

**Main()**

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| first\_str | Первая строка | Текст |
| second\_str | Вторая строка | Текст |
| Выходные величины | | |
| answer | Вывод совпавших символов и количество совпадений этих символов. | Текст |

**GetString()**

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| input | Данные, которые вводит пользователь. | Текст |
| Выходные величины | | |
| str | Проверенная на правильность строка. | Текст |

**GetMatches()**

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| first\_str | Первая строка | Текст |
| second\_str | Вторая строка | Текст |
| Выходные величины | | |
| matches | Сообщение, содержащее совпавшие символы и их количество первой строки, которые нашлись во второй строке. | Текст |

**GetMatchesCount()**

|  |  |  |
| --- | --- | --- |
| Имя | Назначение | Тип/Диапазон |
| Входные величины | | |
| SearchedChar | Символ первой строки, который нашёлся во второй строке. | Символ |
| second\_str | Вторая строка | Текст |
| Выходные величины | | |
| s | Сообщение, содержащее количество символов первой строки, которые нашлись во второй строке. | Текст |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Проверка на совпадения символов первой строки с символами второй строки. | first\_str = “мама”, second\_str = “рама” | answer =  “м найдено 1 совпадение  а найдено 2 совпадения” |
| 2 | Проверка на совпадения символов первой строки с символами второй строки. | first\_str = “профессор”, second\_str = “каша” | answer =  “п не найдено совпадений  р не найдено совпадений  о не найдено совпадений  ф не найдено совпадений  е не найдено совпадений  с не найдено совпадений” |
| 3 | Проверка на корректность ввода | first\_str = 200 | Вывод сообщения на экран: “Ваша строка содержит некорректные символы!” |
| 4 | Проверка на совпадения символов первой строки с символами второй строки (есть символы, что нашлись, и символы, что не нашлись) | first\_str = “профессор”  second\_str = “процессор” | answer =  п найдено 1 совпадение  р найдено 2 совпадения  о найдено 2 совпадения  ф не найдено совпадений  е найдено 1 совпадение  с найдено 2 совпадения |

### Код программы

**Main()**

|  |
| --- |
| string first\_str = GetString("Введите первое слово");  string second\_str = GetString("Введите второе слово");  string answer = GetMatches(first\_str, second\_str);  Console.WriteLine(answer);  Console.ReadKey(); |

**GetString()**

|  |
| --- |
| bool flag = true;// нужен для проверки: есть ли в строке небуквенные символы  string str = "";  while ((str=="")||(flag==false))  {  Console.WriteLine(message);  string input = Console.ReadLine();  char [] symbols = input. ToCharArray();  for (int i = 0; i < symbols. Length; i++)  {  if (Char.IsLetter(symbols[i]))  {  str += symbols[i];  }  else  {  flag = false;  }  }  if (flag==false)  {  Console.WriteLine("Ваша строка содержит некорректные символы! \n");  }  }  return str; |

**GetMatches()**

|  |
| --- |
| string matches = "";  //удаляем повторяющиеся символы  for (int i = 0; i <= first\_str.Length - 2; i++)  {  for (int j = i + 1; j <= first\_str.Length - 1; j++)  {  if (Char.ToLower(first\_str[i]) == Char.ToLower(first\_str[j]))  {  first\_str = first\_str.Remove(j, 1);  }  }  }  //проверяем вторую строку на содержание символов первой строки  for (int i = 0; i < first\_str.Length; i++)  {  if (second\_str.Contains(Char.ToLower(first\_str[i])))  {  matches += first\_str[i] + " найдено " + GetMatchesCount(first\_str[i], second\_str)+"\n";// функция вызывается для каждого символа, нашедшегося во второй строке.  }  else  {  matches += first\_str[i] + " не найдено совпадений\n";  }  }  return matches; |

**GetMatchesCount()**

|  |
| --- |
| int count = 0;  char[] str = second\_str.ToCharArray();\\преобразуем в массив символов входную строку для проверки каждого символа строки second\_str  for (int i = 0; i < str.Length; i++)  {  if (Char.ToLower(str[i]) == Char.ToLower(SearchedChar))  {  count++;//если символ str совпал с входным символом, то count увеличивается  }  }  string s = "";  if (count == 1)  {  s = Convert.ToString(count) + " совпадение";  }  if ((count<=4)&&(count!=1)&&(count!=0))  {  s = Convert.ToString(count) + " совпадения";  }  else if(count!=1)  {  s = Convert.ToString(count) + " совпадений";  }  return s; |

### Автотесты

**Проверка теста 1**

|  |
| --- |
| [TestMethod()]  public void GetMatchesTest1()  {  string s = "м найдено 1 совпадение\nа найдено 2 совпадения\n";  var str = Logic.GetMatches("мама", "рама");  Assert.AreEqual(str, s);  } |

**Проверка теста 2**

|  |
| --- |
| [TestMethod()]  public void GetMatchesTest2()  {  string s = "п не найдено совпадений\nр не найдено совпадений\nо не найдено совпадений\nф не найдено совпадений\nе не найдено совпадений\nс не найдено совпадений\n";  var str = Logic.GetMatches("профессор", "каша");  Assert.AreEqual(str, s);  } |

**Проверка теста 4**

|  |
| --- |
| [TestMethod()]  public void GetMatchesTest()  {  string s = "п найдено 1 совпадение\nр найдено 2 совпадения\nо найдено 2 совпадения\nф не найдено совпадений\nе найдено 1 совпадение\nс найдено 2 совпадения\n";  var str = Logic.GetMatches("профессор", "процессор");  Assert.AreEqual(str, s);  } |