

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

Отчет по лабораторной работе №2
по дисциплине «Объектно-Оrientированное Программирование»
на тему «Работа с текстовыми строками »

Выполнил:
студент группы БИВТ-24-5

Черных Богдан

Проверил:
Стучилин В. В.

1 ЦЕЛЬ РАБОТЫ:

Целью данной работы является изучение и применение методов работы со строковыми данными в C#. Это включает практику разбиения текста на слова, использование коллекций (например, Dictionary) для создания словаря кодов, замену слов на коды и обратно, а также выполнение анализа текста: вычисление процентного распределения слов по первой букве и поиск/суммирование числовых значений, встроенных в текст.

2. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ.

12. Считая, что в памяти компьютера хранится таблица кодов часто встречающихся слов, ввести текст в массив, заменяя слова кодами после ввода. Распечатать текст в исходном виде, т.е. заменяя коды словами.

13. Определить долю в процентах слов, начинающихся на различные буквы. Выписать эти буквы и доли начинающихся на них слов.

14. Текст содержит слова и целые числа от 1 до 10. Найти сумму включенных в текст чисел.

3. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Текстовые строки – переменные типа `string` – могут содержать любое количество символов. Каждый символ представлен в кодировке UNICODE, предполагающей представление одного символа в 2 байтах памяти. Работа с текстовыми строками обычно предполагает решение следующих задач: объединить две (или более) строки в одну, вырезать из строки фрагмент, найти в строке заданную подстроку и т. п.

Для текстовых строк определены операции: *конкатенации (объединение) строк*, эта операция может быть выполнена при помощи оператора `+` или метода `String.Concat`, сравнение строк: операторы равенства `==` и `!=`. Оператор `[]` служит для доступа (только для чтения) к отдельным символам объекта `string` (см. ниже).

Например, объединение строк

```
string str1="Катя";
```

```
string str2="Иванова";

string res = str1+ " "+ str2;

Console.WriteLine(res);
```

Строка `res` будет следующей: "Катя Иванова".

Этот же результат можно получить при помощи метода `Concat`:

```
string res1 = String.Concat(str1, " ",str2);
```

В списке аргументов метода `Concat` может быть не более четырех членов. В приведенном выше примере их три.

Текстовые строки имеет сходство с массивами в том смысле, что доступ к отдельным символам осуществляется по индексу (номеру позиции, нумерация начинается с 0). Переход к следующему символу легко осуществляется изменением номера позиции на 1. Однако следует иметь в виду, что **изменение однажды созданной строки не допускается**. Чтобы внести требуемые изменения, нужно создать новую строку или использовать соответствующий метод.

Для работы со строками определены следующие методы класса `String`.

Метод `Substring` – извлечение подстроки длиной *m*, начиная с *n*-го символа. Если *m* не указано, то выводится весь «хвост» строки, начиная с *n*-го символа. Например, применение метода `Substring` для строки `res`

```
string str1 = "Катя";

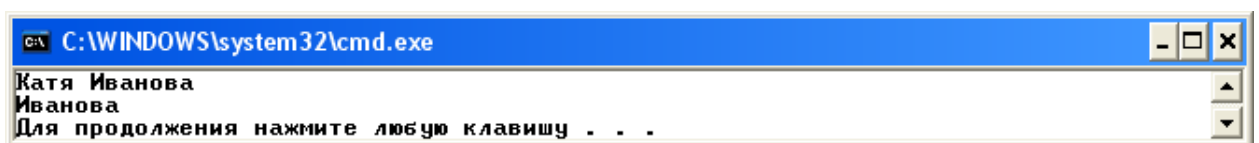
string str2 = "Иванова";

string res = str1 + " " + str2;

Console.WriteLine(res);

string res2 = res.Substring(5, 7);

Console.WriteLine(res2);
```



Метод `Insert` – вставка подстроки в исходную строку, начиная с *n*-ой позиции. При этом необходимо создать новую переменную, в которой вначале будет автоматически

продублирована исходная строка, а затем выполнена необходимая операция. (Еще раз отметим, что в С# прямая модификация существующей строки невозможна.) Например,

```
string res3 = "Катя";  
  
string res4 = res3.Insert(4, " Иванова");  
  
Console.WriteLine(res4);
```

В результате в `res4` будет "Катя Иванова".

Метод `Replace` – замена подстроки новой подстрокой или замена какого-либо символа во всем тексте на другой символ. Например,

```
string str = "Катя Иванова";  
  
string str1 = str.Replace("Катя", "Екатерина");
```

В результате будет `str1 = "Екатерина Иванова"`.

Метод `Remove` – удаление из строки фрагмента заданной длины *m*, начинающегося с заданной позиции *n*. Например,

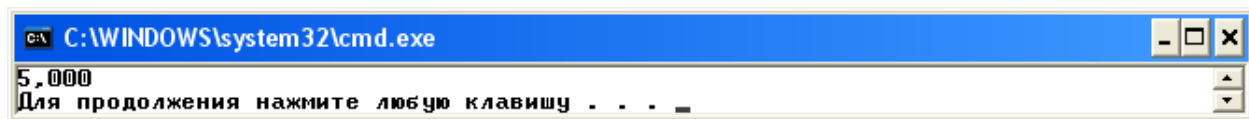
```
string res = "Катя Иванова – моя подруга";  
  
string res1 = res.Remove(4, 8);  
  
Console.WriteLine(res1);
```

Удаляются 8 символов, начиная с 4-го (пробел после слова "Катя"). В результате будет `res1 = "Катя – моя подруга"`.

Метод `ToString` – получение строкового представления объекта числового типа. Внутри скобок может ничего не находиться, или может находиться переменная (строка формата), в которой указан способ форматирования числа, сохраненного в строковом виде, при выводе на консоль. Например,

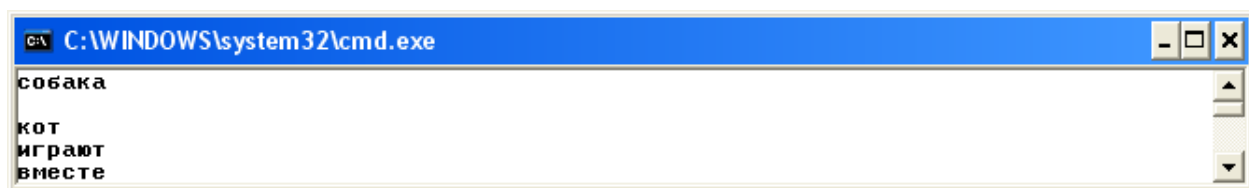
```
double a = 5.0;  
  
string s = a.ToString("f3");  
  
Console.WriteLine(s);  
  
Console.ReadKey();
```

Локальные		
Имя	Значение	Тип
a	5.0	double
s	"5,000"	string



Метод `Split` (применяется к экземпляру класса `String`, как и описанные выше методы) осуществляет разбор строки, т.е. позволяет выделить отдельные слова или другие сочетания символов, разделенные какими-либо разделителями, перечисляемыми в массиве символов типа `char[]`, являющемся аргументом метода `Split`. Например,

```
string str = "собака, кот играют вместе";
string[] strarr = str.Split(new Char[] { ' ', ',', '.' });
foreach (string res in strarr)
{
    Console.WriteLine(res);
}
```



Здесь формируется массив `strarr` из слов исходного текста, отделенных друг от друга пробелом или запятой. Список символов-разделителей помещается в массив и передается методу `Split` в качестве аргумента.

При выводе после первого слова выведена пустая строка, т. к. после первого разделителя сразу следует второй и между ними ничего нет (пусто).

Каждый отдельный символ строки является значением типа `char` и может быть выделен в переменную типа `char`. Например, при выполнении оператора

```
char s = str[5];
```

для `str` из предыдущего примера `s` получит значение `a`.

Для работы с отдельными символами строки можно использовать методы структуры `Char`. Например, статический метод

`Char.IsDigit(s)`, где `s` – переменная типа `char` (отдельный символ строки). Этот метод возвращает значение `true`, если `s` – цифра и `false`, если `s` – не цифра. Результат выполнения метода показывает, относится ли указанный символ Юникода к категории десятичных цифр или нет.

4. БЛОК-СХЕМА (ОПЦИОНАЛЬНО).

.....
.....
-

5. ЛИСТИНГ ПРОГРАММЫ

.....
.....

```
//svg does precious
using System;                                     // Аналог <iostream> для работы с консолью
и основными функциями
using System.Collections.Generic;                 // Аналог <vector>, <list>, <map>, <set>,
<unordered_map>, <unordered_set>, <stack>, <queue>
using System.Text;                               // Аналог <string>, <cstring> (для работы
со строками и StringBuilder)
using System.Linq;                               // Аналог <algorithm> (для работы с LINQ,
сортировок, поиска и т.д.)
using System.IO;                                 // Аналог <cstdio>, <fstream> (работа с
файлами)
using System.Globalization;                      // Аналог <iomanip> (для форматирования)
using System.Collections;                        // Работа с различными коллекциями
(например, ArrayList)
using System.Threading;                          // Потоки и многопоточность
using System.Runtime.Serialization;              // Аналог <stdexcept> (работа с
исключениями)
using System.Reflection;                        // Аналог <typeinfo> (информация о типах,
рефлексия)
using System.Diagnostics;                        // Аналог <utility>, <std::pair>
(вспомогательные функции и классы)
using System.ComponentModel;                     // Дополнительные утилиты и атрибуты
using System.Numerics;                          // Работа с большими числами и
математическими операциями
using System.Globalization;
using System.Diagnostics;
using System.Net;
using System.Numerics;
// Для работы с потоками данных:
using System.Threading.Tasks;                    // Асинхронные задачи
```

```

// Для работы с датами и временем:
using System.Timers;
using System.Collections.Generic;
using System.Text;
using System.Linq;
using System.IO; //important
using System.Globalization;
using System.Collections;
using System.Threading;
using System.Runtime.Serialization;
using System.Reflection;
using System.Diagnostics;
using System.ComponentModel;
using System.Numerics;
using System.Globalization;
using System.Diagnostics;
using System.Net;
using System.Numerics;
using System.Threading.Tasks;
using System.Security.Cryptography;
using System.Data;
using System.Data.SqlClient;
using System.Xml;
using System.Xml.Linq;
using System.Runtime.InteropServices;
using System.Security;
using System.Web;
using System.Media;
using System.Drawing;
using System.Configuration;
using System.Timers;
using System.Runtime.Remoting;
using System.Runtime.CompilerServices;
using System.Runtime.Versioning;
using System.CodeDom;
using System.CodeDom.Compiler;
using System.Collections.Concurrent;
using System.Runtime;
using System.Windows;
using System.Windows.Input;
using System.Security.Principal;
using System.Security.Permissions;
using System.Resources;
using C = System.Console; //console
using dl = System.Decimal; //decimal
using str = System.String; //string
using l = System.Int64; //long
using u = System.UInt64; //Ulong
using db = System.Double; //Double

/*
Izzspot - 19 years
Boogie B - 20 years
SJ - 21 years
Bandokay - life sentence

Youngest in the charge
OFB, we don't window shop
Bro caught him an opp and tried turn him off (Bow, bow)
In this X3, man's swervin' off (Skrr, skrr)
Free Boogie Bando, he got birded off (Free my bro, free my bro)
Whenever we get a burner loss
We just cop a next one and go burst it off (Ay)
Lil bro's tellin' me he got his earnings wrong
We just took him OT, now his trapline's gone (Ring, ring)
Hashtag
Bro backed this ting and just started squeezin' (Clarted)

```

```

When it broad day, it was freezin'
Hashtag fuckery, hashtag screamin'
One on the hand ting woi, left man leanin', leanin' (Fucker)
Show us cause it's good to feel it
Shortie's cooze and she must be dreamin'
Vogue
Ra-ra-racks came in, hello (Hello?)
Fu-fucked this girl on Vogue
Big Range, ain't no Evoque
Big chain, this ain't a choker
Sh-sh-she like when her neck get choked
I know that her boyfriend knows
You tell white lies like cocaine
I know that her boyfriend knows
You tell white- (Cocaine)
She hate when her man get home
*/
/* ;Adelante Barcelona, adelante Cataluña! Visca el Barça! Visca Catalunya!
    ;Al diablo con todos los demás, porque lo más importante en la vida es el
fútbol!
*/
//-----
----
//-----
----
//-----
----

```

```

namespace CombinedTasks
{
    class EncoderDecoder
    {
        private Dictionary<string, string> wordToCode;
        private Dictionary<string, string> codeToWord;
        public EncoderDecoder()
        {
            wordToCode = new Dictionary<string, string>()
            {
                { "и", "001" },
                { "в", "002" },
                { "на", "003" },
                { "с", "004" },
                { "Barcelona", "021" },
                { "best", "228" },
                { "club", "027" },
                { "world", "052" }
            };
            codeToWord = new Dictionary<string, string>();
            foreach (var pair in wordToCode)
                codeToWord.Add(pair.Value, pair.Key);
        }
        public string EncodeWord(string word) =>
            wordToCode.ContainsKey(word) ? wordToCode[word] : word;
        public string DecodeWord(string word) =>
            codeToWord.ContainsKey(word) ? codeToWord[word] : word;
    }

    class TextProcessor
    {
        private EncoderDecoder encoderDecoder;
        private string[] encodedText;
        public TextProcessor() => encoderDecoder = new EncoderDecoder();
        public void InputAndEncode()
        {
            Console.WriteLine("Задача 12. Введите текст:");
            string input = Console.ReadLine();

```



```

        string[] words = input.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
        encodedText = new string[words.Length];
        for (int i = 0; i < words.Length; i++)
            encodedText[i] = encoderDecoder.EncodeWord(words[i]);
    }
    public void PrintEncoded()
    {
        for (int i = 0; i < encodedText.Length; i++)
            Console.Write(encodedText[i] + (i < encodedText.Length - 1 ? " " :
""));
        Console.WriteLine();
    }
    public void PrintOriginal()
    {
        for (int i = 0; i < encodedText.Length; i++)
            Console.Write(encoderDecoder.DecodeWord(encodedText[i]) + (i <
encodedText.Length - 1 ? " " : ""));
        Console.WriteLine();
    }
}

class LetterCounter
{
    public static void CalculatePercentages(string text)
    {
        string[] words = text.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
        Dictionary<char, int> counts = new Dictionary<char, int>();
        foreach (string w in words)
        {
            char first = char.ToLower(w[0]);
            if (counts.ContainsKey(first))
                counts[first]++;
            else
                counts[first] = 1;
        }
        int total = words.Length;
        foreach (var pair in counts)
            Console.WriteLine(pair.Key + " " + ((pair.Value / (double)total) *
100.0).ToString("F2") + "%");
    }
}

class TextNumbers
{
    public static void SumNumbers(string text)
    {
        string[] parts = text.Split(new char[] { ' ', ',', '.', ';', '!', '?'
}, StringSplitOptions.RemoveEmptyEntries);
        int sum = 0;
        foreach (string part in parts)
            if (int.TryParse(part, out int number) && number >= 1 && number <=
10)
                sum += number;
        Console.WriteLine("Сумма чисел от 1 до 10: " + sum);
    }
}

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Выберите задачу (12, 13, 14):");
        string choice = Console.ReadLine();
        switch (choice)
        {

```

```

        case "12":
            TextProcessor processor = new TextProcessor();
            processor.InputAndEncode();
            Console.WriteLine("\nЗакодированный текст:");
            processor.PrintEncoded();
            Console.WriteLine("\nДекодированный (исходный) текст:");
            processor.PrintOriginal();
            break;
        case "13":
            Console.WriteLine("Задача 13. Введите текст:");
            string textForLetters = Console.ReadLine();
            LetterCounter.CalculatePercentages(textForLetters);
            break;
        case "14":
            Console.WriteLine("Задача 14. Введите текст:");
            string textForNumbers = Console.ReadLine();
            TextNumbers.SumNumbers(textForNumbers);
            break;
        default:
            Console.WriteLine("Неверный выбор");
            break;
    }
}
}
}

```

6. ПРИМЕР ВЫПОЛНЕНИЯ (СКРИНШОТЫ)

.....

.....



Консоль отладки Microsoft



Выберите задачу (12, 13, 14):

12

Задача 12. Введите текст:

Barcelona is the best club in the world

Закодированный текст:

005 is the 006 007 in the 008

Декодированный (исходный) текст:

Barcelona is the best club in the world



Консоль отладки Microsoft



Выберите задачу (12, 13, 14):

13

Задача 13. Введите текст:

чтобы избежать оверфиттинга в первую очередь можно подумать о снижении размерности модели

ч 8,33%

и 8,33%

о 25,00%

в 8,33%

п 16,67%

м 16,67%

с 8,33%

р 8,33%



Консоль отладки Microsoft



Выберите задачу (12, 13, 14):

14

Задача 14. Введите текст:

В классе было 3 стула, 8 учеников и 10 парт.

Сумма чисел от 1 до 10: 21

7. ВЫВОД

.....

.....

В ходе выполнения работы были реализованы три задачи, каждая из которых демонстрирует различные подходы к обработке текстовой информации. Задача 12 показывает, как можно применять методы кодирования и декодирования текста с использованием словаря, что помогает понять основы манипуляции строковыми данными. Задача 13 иллюстрирует применение методов анализа текста для вычисления долей слов, начинающихся на различные буквы, что полезно для статистического анализа. Задача 14 демонстрирует поиск и суммирование чисел, встроенных в текст, показывая, как можно извлекать и обрабатывать числовую информацию. В итоге, работа позволяет закрепить навыки работы со строками и коллекциями в C#, а также способствует пониманию принципов обработки текстовых данных.