**Контрольная работа №1**

студента Акимова Артемия

группы 241

Постановка задачи:

Необходимо подсчитывает общее число вхождений символов х и y в сообщении. Выполнить решение тремя способами и провести временной анализ работы блока алгоритма, отвечающий непосредственно за выполнение данной задачи.

Комментарии к предыдущему решению: я изначально решил поставленную задачу именно таким образом, простым способом прохода по строке как массиву символов. Но после изменил решение на поиск в строке не символов, а подстрок длиной в 1 символ, так как в простом варианте решения (представленном в данном случае) используется мало методов для членов классов String и StringBuilder, то есть требовалось по возможности избегать типа char.

Решение задачи с использование класса string:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Diagnostics;

namespace KP\_StringBuilder1

{

class Program

{

static void Main()

{

String a = Console.ReadLine();

Console.WriteLine("Дана строка: {0}", a);

char[] chars = new char[2];

for (int i = 0; i < 2 ; i++ )

{

chars[i] = char.Parse(Console.ReadLine());

}

int count = 0;

Stopwatch time = new Stopwatch();

time.Start();

for (int x = 0; x < a.Length; x++)

{

if ((a[x] == chars[0]) || (a[x] == chars[1]))

{

count++;

}

}

time.Stop();

Console.WriteLine("Символы {0} и {1} встречаются в ней {2} раз {3}", chars[0], chars[1], count, time.ElapsedTicks);

}

}

}

Решение задачи с использование класса StringBuilder:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Diagnostics;

namespace KP\_StringBuilder

{

class Program

{

static void Main()

{

StringBuilder a = new StringBuilder(Console.ReadLine());

Console.WriteLine("Дана строка: {0}", a);

char[] chars = new char[2];

for (int i = 0; i < 2 ; i++ )

{

chars[i] = char.Parse(Console.ReadLine());

}

int count = 0;

Stopwatch time = new Stopwatch();

time.Start();

for (int x = 0; x < a.Length; x++)

{

if ((a[x] == chars[0]) || (a[x] == chars[1]))

{

count++;

}

}

time.Stop();

Console.WriteLine("Символы {0} и {1} встречаются в ней {2} раз {3}", chars[0], chars[1], count, time.ElapsedTicks);

}

}

}

Решение задачи с использованием регулярных выражений

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.IO;

using System.Diagnostics;

namespace KP\_Regex

{

class Program

{

static void Main()

{

string p = @"1|2";

Regex r = new Regex(p, RegexOptions.IgnoreCase);

string a=Console.ReadLine();

int count=0;

Stopwatch time = new Stopwatch();

time.Start();

Match tel = r.Match(a);

MatchCollection ALL = r.Matches(a);

DateTime stop = DateTime.Now;

time.Stop();

Console.WriteLine("Количество символов 1 и 2 в тексте в тексте {0} {1}", ALL.Count, time.ElapsedTicks);

}

}

}

Пример: найти общее количество символов 1 и 2 в тексте

|  |  |  |
| --- | --- | --- |
| № теста | Входные данные | Выходные данные |
| 1 | В 1996 году, 12 мая того года, была зафиксирована самая жаркая погода того года – 41 градус Цельсия | 4 |
| 2 | 112111211АаАа789221 | 12 |
| 3 | 12 | 2 |
| 4 | 0 | 0 |
| 5 | 1X254//(254 единичек) | 254 |

Полученные результаты:

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Время выполнения (в тактах) | | |
| String | StringBuilder | Регулярные выражения |
| 1 | 14 | 15 | 3803 |
| 2 | 15 | 15 | 3794 |
| 3 | 15 | 14 | 3880 |
| 4 | 12 | 12 | 3712 |
| 5 | 36 | 31 | 3932 |
| Среднее значение | 18 | 17 | 3824 |

ВЫВОДЫ:

На одних и тех же входных данных можно заметить следующее: программа, разработанная с использованием класса StringBuilder работает быстрее, чем программа с использованием класса String даже на достаточно маленьких объеме входных данных. При увеличении количества итераций, превосходство класса StringBuilder будет расти.

В то же время программа, использующая регулярные выражения работает кардинально медленнее, чем StringBuilder и String. Это происходит из-за того, что все функции класса Regex работают гораздо медленнее, чем аналогичные функции класса String.В связи с этим следует сделать вывод: предпочтительно использовать класс StringBuilder, для решения конкретной задачи ввиду большей скорости работы (не учитывая затраты на память и погрешность производительности процессора в данный момент времени).