Лабораторная работа №6

“Исследование способов профилирования программного обеспечения”

1 ЦЕЛЬ РАБОТЫ

Исследовать критические по времени выполнения участки программного кода и возможности их устранения. Приобрести практические навыки анализа программ с помощью профайлера EQATECProfiler.

2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Разработать программу на основе библиотеки классов, реализованной и протестированной в предыдущей работе. Программа должна как можно более полно использовать функциональность класса. При необходимости для наглядности профилирования в методы класса следует искусственно внести задержку выполнения.

3.2. Выполнить профилирование разработанной программы, выявить функции, на выполнение которых тратится наибольшее время.

3.3. Модифицировать программу с целью оптимизации времени выполнения.

3.4. Выполнить повторное профилирование программы, сравнить новые результаты и полученные ранее, сделать выводы.

3 ТЕКСТ ПРОГРАММЫ

using System;

using System.Threading;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

EInString eStr = new EInString();

eStr.CheckAndPrint();

}

}

class EInString

{

private static string str;

public bool result;

public EInString()

{

str = "qwertfgeeeeeunm";

}

public EInString(string \_str)

{

str = \_str;

}

public void CheckAndPrint()

{

if (EIn(str))

{

Console.WriteLine("В строке есть повторяющаяся 5 раз буква \"e\"");

result = true;

}

else

{

Console.WriteLine("В строке нет повторяющейся 5 раз буквы \"e\"");

result = false;

}

}

private bool EIn(string str)

{

int n = 0;

for (int i = 0; i < str.Length; i++)

{

if (str[i] == 'e' || str[i] == 'е')

{

n++;

if (n == 5) return true;

}

else n = 0;

}

return false;

}

}

}

4 РЕЗУЛЬТАТЫ

Вносим в методы классов искусственные задержки для наглядности профилирования.

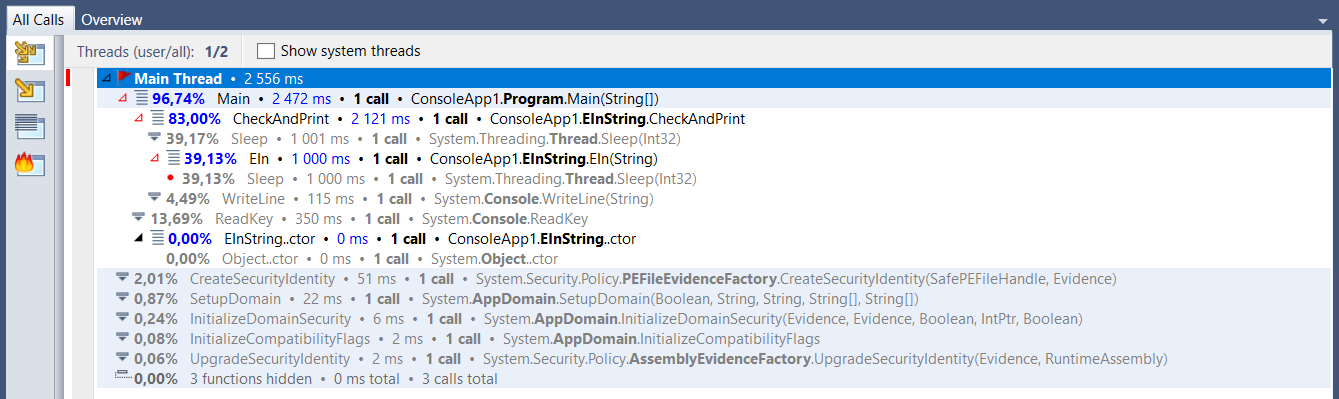


Рисунок 1 – Результат профилирования программы со внесенными модификациями

Для оптимизации времени выполнения программы, удаляем задержку и метод ReadKey(). В итоге, получаем, что время выполнения сократилось более, чем в 10 раз.

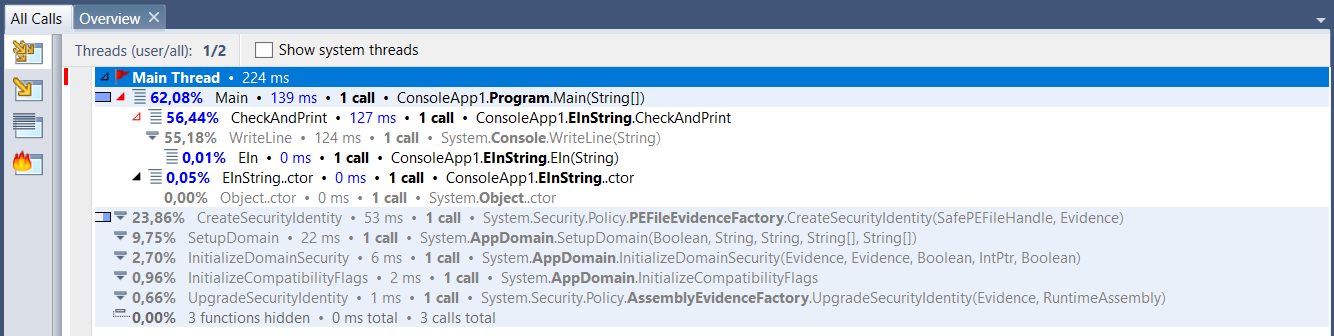


Рисунок 2 – Результат профилирования исходной программы

ВЫВОД

В ходе выполнения лабораторной работы были получены практические навыки анализа программ с помощью профайлера dotTrace. С помощью профилировщика было выявлено время выполнения кода и в итоге, оно было оптимизированно более, чем в 10 раз.