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 ConsoleApp6

{

class Program

{

static void Main(string[] args)

{

Student student = new Student("Долженко Илья Андреевич", 32, "ТПО", 98, DateTime.Now);

Console.WriteLine($"Студент: {student.getName()}, Группа: {student.getGroup()}");

Console.WriteLine($"Предмет: {student.getSubject()}, Оценка: {student.getMark()}, Дата: {student.getTime()}");

Console.WriteLine("----------------------------------------------------");

Student student2 = new Student("Волобуев Юрий Сергеевич", 32, "ТПО", 65, DateTime.Now);

Console.WriteLine($"Студент: {student2.getName()}, Группа: {student2.getGroup()}");

Console.WriteLine($"Предмет: {student2.getSubject()}, Оценка: {student2.getMark()}, Дата: {student2.getTime()}");

Console.ReadKey();

}

}

public class Student

{

private string name;

private int group;

private int mark;

private string subject;

private DateTime time;

public Student(string name, int group, string subject, int mark, DateTime time)

{

this.group = group;

this.name = name;

this.subject = subject;

this.mark = mark;

this.time = time;

}

public string getName()

{

return name;

}

public int getGroup()

{

return group;

}

public int getMark()

{

return mark;

}

public string getSubject()

{

return subject;

}

public DateTime getTime()

{

return time;

}

}

}

4 РЕЗУЛЬТАТЫ

Для наглядности профилирования была искусственно создана задержка выполнения в каждом методе класса.

На рисунке 1 видно, что до модификации программы время ее выполнения составляет 1с. На выполнение функции getMark тратится наибольшее количество времени (241мс).

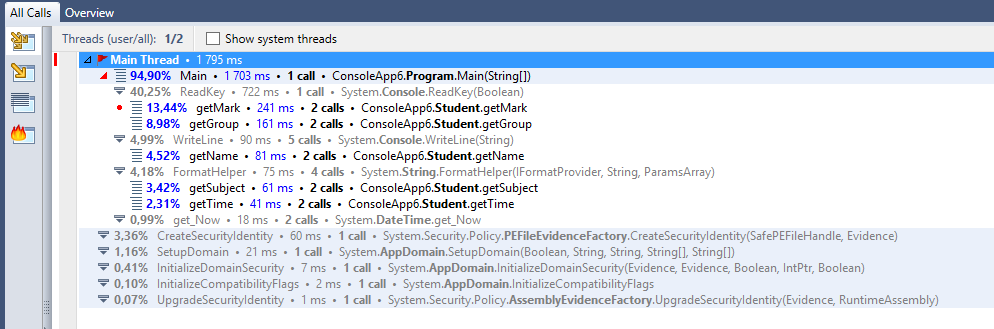


Рисунок 1 – Время выполнения методов до модификации

Убрав искусственную задержку в методах, мы получаем время выполнения программы 200мс, что в 5 раз быстрее выполнения программы до модификации.

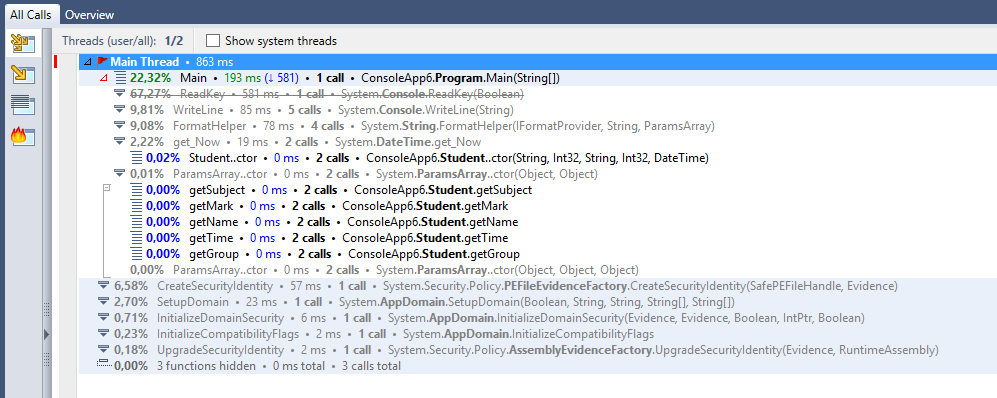


Рисунок 2 – Время выполнения методов после модификации

ВЫВОД

В ходе выполнения лабораторной работы были получены практические навыки анализа программ с помощью профайлера JetBrains dotTrace. Была проанализирована программа на наличие критических участков кода и была выполнена ее модификация, которая ускорила программу в 5 раз.