Практическая работа 9
Использование методов оптимизации в модуле

Цель занятия: Получить практический опыт использования методов оптимизации кода в VisualStudio 2015

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio)
Microsoft Visual Studio 2015

Краткие теоретические сведения

Профилировкой называется измерение производительности как всей программы в целом, так и отдельных ее фрагментов, с целью нахождения "горячих" точек (Hot Spots), — тех участков программы, на выполнение которых расходуется наибольшее количество времени.

Подавляющее большинство вычислительных алгоритмов так или иначе сводятся к циклам, — т. е. многократным повторениям одного фрагмента кода, причем зачастую циклы обрабатываются не последовательно, а образуют более или менее глубокие иерархии, организованные по типу "матрешки". В результате, большую долю всего времени выполнения, программа проводит в циклах с наибольшим уровнем вложения, и именно их оптимизация дает наилучший прирост производительности.

Когда же алгоритм программы прост, а ее исходный текст свободно умещается в сотню–другую строк, то "горячие" точки нетрудно обнаружить и визуальным просмотром листинга. Но с увеличением объема кода это становится все сложнее и сложнее. В программе, состоящей из тысяч сложно взаимодействующих друг с другом функций (часть из которых это функции внешних библиотек и API — Application Programming Interface, интерфейс прикладного программирования — операционной системы) далеко не так очевидно: какая же именно из них в наибольшей степени ответственна за низкую производительность приложения. Естественный выход — прибегнуть к помощи специализированных программных средств.

Профилировщик (так же называемый "профайлером") — основной инструмент оптимизатора программ. Основная цель профилировки — исследовать характер поведения приложения во всех его точках. Под "точкой" в зависимости от степени детализации может подразумеваться как отдельная машинная команда, так целая конструкция языка высокого уровня (например: функция, цикл или одна–единственная строка исходного текста).

Большинство современных профилировщиков поддерживают следующий набор базовых операций:

- определение общего времени исполнения каждой точки программы (total [spots] timing);

- определение удельного времени исполнения каждой точки программы ([spots] timing);

- определение причины и/или источника конфликтов и пенальти (penalty information);

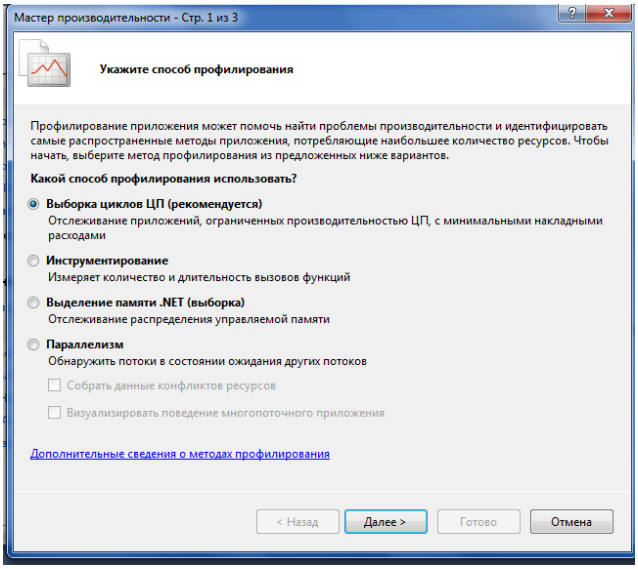
- определение количества вызовов той или иной точки программы ([spots] count);

- определение степени покрытия программы ([spots] covering).

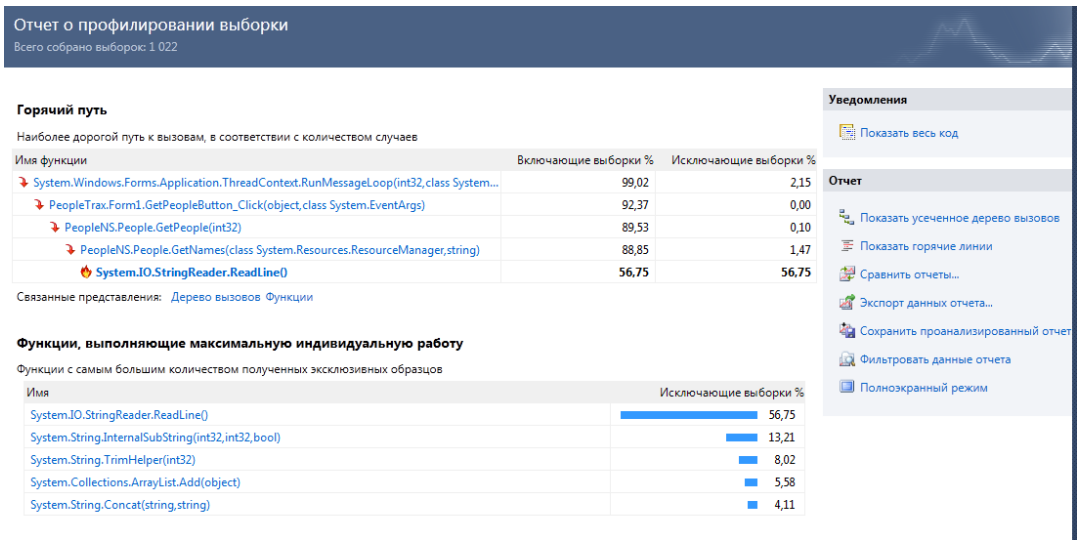
Запуск профилировщика .NET

Чтобы запустить профилировщик, перейдите в Visual Studio к Анализ и выберите Запустить мастер производительности.

Открывается следующее диалоговое окно:



Последовательно выбирая окна, получите отчет о профилировании выборки.



Задания

1 Изучить теоретические сведения и задание к работе
2 В соответствии с вариантом задания провести оптимизацию модуля.

Порядок выполнения работы

Содержание отчета

1. Название работы
2. Цель работы
3. Перечень оборудования и программного обеспечения
4. Порядок выполнения
5. Листинг оптимизации модуля в соответствии с вариантом задания.
6. Вывод.

Варианты заданий

Варианты заданий представлены в практической работе 1.

Используемая литература

1 Рудаков А.В. Технология разработки программных продуктов: учеб. пособие для студ. учреждений сред. проф. образования – 11-е изд., стер. – М.: Издательский центр «Академия», 2017. – 208 с.

2 Федорова Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учеб. пособие для студ. учреждений сред. проф. образования / Г.Н. Федорова.–: М.: КУРС : ИНФРА – М, 2017. – 334 с

3 Федорова Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: Учебное пособие для СПО.- М.: КУРС, 2018. – 333 с.– ЭБС Знаниум

4 http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx.