

Министерство образования и науки Российской Федерации
ФГАОУ ВО «КФУ им. В.И. Вернадского»
физико-технический институт (структурное подразделение)

Кафедра компьютерной инженерии и моделирования

Шенгелай Всеволод Михайлович

отчет по лабораторной работе №1

по дисциплине

«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

Направление подготовки:
09.03.04 "Программная инженерия"

Оценка – 90



Симферополь, 2021

Лабораторная работа №1

Тема: Введение в C# и .NET Framework

Цель работы: Научиться создавать простейшие консольные и WPF приложения на языке C# в среде Visual Studio, изучить возможность создания самодокументируемых приложений.

Описание ключевых понятий:

IDE (Integrated Development Environment) — это набор инструментов разработчика ПО, собранный в составе единого приложения и облегчающий труд программиста при написании приложений.

CLR (Common Language Runtime, виртуальная машина) – общезыковая исполнительная среда, обеспечивает выполнение сборки и является основным компонентом .NET Framework. Под Виртуальной Машиной понимают абстракцию операционной системы высокого уровня, которая обеспечивает выполнение (управляемого) программного кода. Управляемый код – программный код, который при своём выполнении способен использовать службы, предоставляемые CLR. Сама CLR состоит из двух главных компонентов: ядра (mscorlib.dll) и библиотеки базовых классов FCL (mscorlib.dll).

FCL (.NET Framework Class Library) - стандартная библиотека классов платформы «.NET Framework».

CLS (Common Language Specification — общая спецификация для языков программирования) – представляет собой набор правил, которые во всех подробностях описывают минимальный и полный комплект функциональных возможностей, которые должен обязательно поддерживать каждый отдельно взятый .NET-компилятор для того, чтобы генерировать такой программный код, который мог бы обслуживаться CLR и к которому в то же время могли бы единообразным образом получать доступ все языки, ориентированные на платформу .NET. В конечном итоге CLS является своего рода набором правил, которых должны придерживаться создатели компиляторов при желании, чтобы их продукты могли без проблем функционировать в мире .NET. Изящество этой идеи

состоит в том, что ограничения в применении CLS-совместимых средств налагаются только на общедоступные и защищенные члены классов и на общедоступные классы. В пределах приватной реализации классов мы вольны писать любой несовместимый с CLS код, поскольку код из других сборок (единиц управляемого кода) в любом случае не имеет доступа к этой части кода.

WPF - Windows Presentation Foundation — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык разметки XAML. В основе WPF лежит векторная система визуализации, основанная на DirectX.

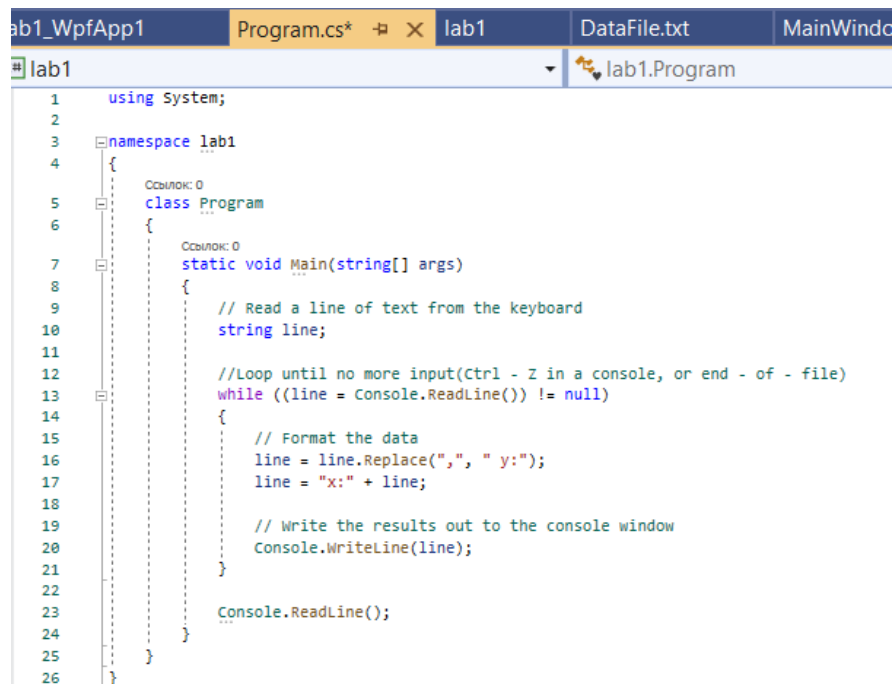
Перед выполнением лабораторной работы изучена следующая литература:

1. Презентация лектора курса: «Основы Net Framework» и «Возможности MSVS».
2. Прочитана Глава1 – «Первое знакомство с Visual Studio 2010» книги Майо ДЖ. Самоучитель Microsoft visual studio 2010.
3. Пройдены начальные разделы видеокурса: «C#. От простого к сложному» Семена Алексеева.
4. Получены начальные сведения о спецификации языка C#. Корпорация Microsoft. Версия 5.0. (Приложение А, Комментарии к документации).
5. Получены начальные сведения о языке C# на сайте MSDN Microsoft и Metanit.com.

Выполнены 3 задания, описанных в методических указания к выполнению лабораторных работ.

Задание 1: Создано консольное приложение с функционалом, описанным в методических указаниях.

В этом упражнении мы сначала создаём приложение консольного ввода-вывода. Затем мы используем перенаправление ввода-вывода для запуска приложения с использованием данных, хранящихся в файле.

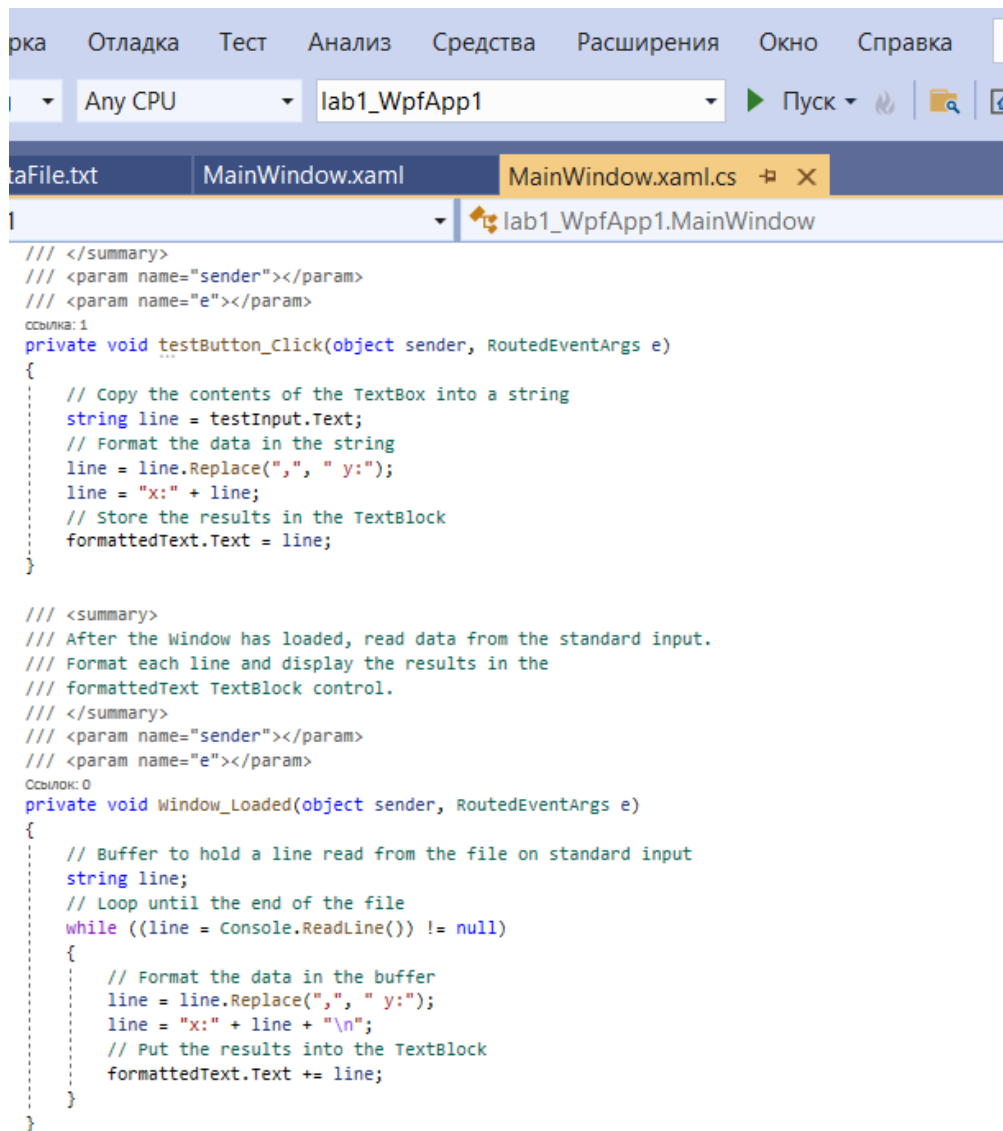


```
1 using System;
2
3 namespace lab1
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             // Read a line of text from the keyboard
10            string line;
11
12            // Loop until no more input (Ctrl - Z in a console, or end - of - file)
13            while ((line = Console.ReadLine()) != null)
14            {
15                // Format the data
16                line = line.Replace(",", " y:");
17                line = "x:" + line;
18
19                // Write the results out to the console window
20                Console.WriteLine(line);
21            }
22
23            Console.ReadLine();
24        }
25    }
26 }
```

Рисунок 1. Код нашего консольного приложения

Задание 2: Создано WPF приложение с функционалом, описанным в методических указаниях.

В этом задании мы создаём простое приложение WPF, которое обеспечивает функциональность, аналогичную консольному приложению, которое мы разработали в задании 1. При загрузке главного окна сразу же выводим в TextBlock отформатированный текст, прочтенный из файла. Мы можем сами внести данные для форматирования в TextBox. К кнопке “testButton” мы добавили обработчик события - при нажатии на неё, текст из TextBox будет помещён в строковую переменную и отформатирован тем же образом.



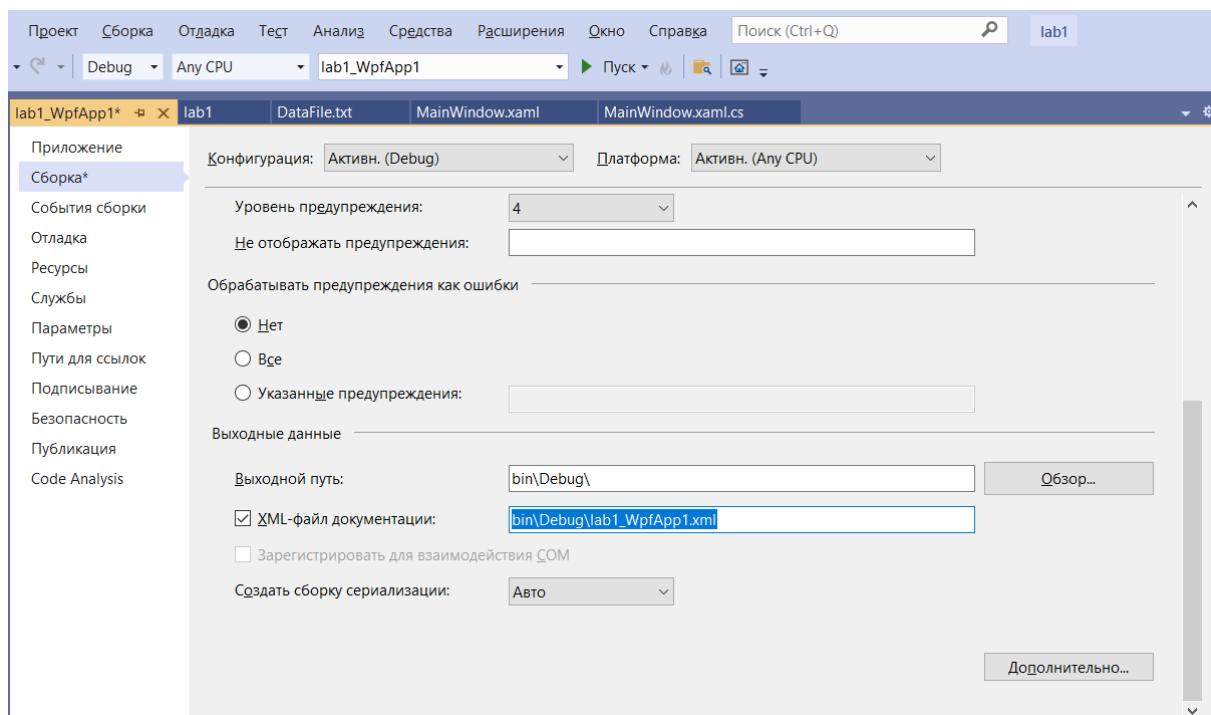


Рисунок 3. Создание XML - файла документации

Представлены 3 проекта, реализованных в Visual Studio Common Edition 2019. Проекты представлены преподавателю в электронной форме, продемонстрирована их работоспособность, разъяснены детали программного кода.

Вопросы, заданные преподавателем:

1) Что такое IDE и какие её преимущества?

IDE (Integrated Development Environment) – это интегрированная, единая среда разработки, которая используется разработчиками для создания различного программного обеспечения. IDE представляет собой комплекс из нескольких инструментов, а именно: текстового редактора, компилятора либо интерпретатора, средств автоматизации сборки и отладчика. Помимо этого, IDE может содержать инструменты для интеграции с системами управления версиями и другие полезные утилиты. Есть IDE, которые предназначены для работы только с одним языком программирования, однако большинство современных IDE позволяет работать сразу с несколькими.

2) Основные различия между WPF и Windows Forms.

WPF основан на графической библиотеке DirectX, а WinForms – на GDI+. GDI+ весьма устаревшая и неоптимизированная технология, в отличие от более современного DirectX. UI и C# бизнес-логика могут быть четко разделены в WPF.

3) Зачем нужен XML файл документации?

Для удобства оформления комментариев в коде.

4) Как осуществляется перенаправление ввода/вывода в командной строке?

С помощью знаков “<” и “>”.