**Лабораторная работа №1**

**Тема:** Знакомство со средой программирования Microsoft Visual Studio .NET

**Цель:** изучить возможности и пользовательский интерфейс системы программирования Microsoft Visual Studio .NET, а также возможности программной платформы .NET Framework по разработке приложений различных предметных областей. Создать минимальное консольное приложение, изучить структуру и основные элементы его кода.

## Основные сведения о системе Microsoft Visual Studio .NET

Microsoft Visual Studio .NET – это интегрированная среда разработки (Integrated Development Environment (IDE)) для создания, документирования, запуска и отладки программ, написанных на языках платформы .NET Framework.

Как и любая другая хорошая среда разработки, Visual Studio .NET включает средства управления проектами, редактор исходного текста, конструкторы пользовательского интерфейса, мастера создания приложений, компиляторы, компоновщики, инструменты, утилиты, документацию и отладчики. Она позволяет создавать приложения для 32- и 64-разрядных Windows-платформ, а также новой платформы .NET Framework. Одно из важнейших усовершенствований – возможность работы с разными языками в единой среде разработки.

В семействе продуктов Visual Studio используется единая интегрированная среда разработки (IDE), состоящая из нескольких элементов: строки меню, панели инструментов «Стандартная», различных закрепленных или автоматически скрываемых окон инструментов в левой, нижней или правой областях, а также области редакторов. Набор доступных окон инструментов, меню и панелей инструментов зависит от типа проекта или файла, в котором выполняется разработка. Расположение окон инструментов и других элементов интегрированной среды разработки может изменяться в зависимости от примененных параметров и настроек, выполняемых пользователем в процессе работы.

Выбор используемых редакторов и конструкторов зависит от типа создаваемого файла или документа.

Представление конструктора WindowsForms позволяет определить расположение элементов управления и других объектов пользовательского интерфейса. Элементы управления можно легко перемещать из панели элементов и располагать в рабочей области конструирования.

В среде Visual Studio предусмотрен мощный набор средств построения и отладки. Благодаря конфигурациям построения можно выбирать компоненты для построения, исключать компоненты, которые не требуется включать в построение, а также определять, как будут построены выбранные проекты и для какой платформы. Конфигурации построений доступны как для решений, так и для проектов.

## Основные сведения о .NET Framework

Платформа .NET Framework – это интегрированный компонент Windows, который поддерживает создание и выполнение нового поколения приложений и веб-служб XML.

Двумя основными компонентами платформы .NET Framework являются общеязыковая среда выполнения (**CLR**) и библиотека классов .NET Framework.

Основой платформы .NET Framework является среда CLR. Среду выполнения можно считать агентом, который управляет кодом во время выполнения и предоставляет основные службы, такие, как управление памятью, управление потоками и удаленное взаимодействие. При этом накладываются условия строгой типизации и другие виды проверки точности кода, обеспечивающие безопасность и надежность. Фактически основной задачей среды выполнения является управление кодом. Код, который обращается к среде выполнения, называют управляемым кодом, а код, который не обращается к среде выполнения, называют неуправляемым кодом.

Другой основной компонент платформы .NET Framework, библиотека классов, представляет полную объектно-ориентированную коллекцию типов, которые применяются для разработки приложений, начиная от обычных, запускаемых из командной строки или с графическим интерфейсом пользователя, и заканчивая приложениями, использующими последние технологические возможности ASP.NET, такие, как Web Forms и веб-службы XML. Библиотека классов платформы .NET Framework представляет собой кол- лекцию типов, которые тесно интегрируются со средой CLR. Библиотека классов является объектно-ориентированной и предоставляет типы, из которых управляемый код пользователя может наследовать функции. Это не только упрощает работу с типами .NET Framework, но также уменьшает время, затрачиваемое на изучение новых средств платформы .NET Framework. Кроме того, компоненты независимых производителей можно легко объединять с классами. Как и ожидается от объектно-ориентированной библиотеки классов, типы

.NET Framework позволяют решать типовые задачи программирования, включая работу со строками, сбор данных, подключения к базам данных и доступ к файлам. В дополнение к этим обычным задачам библиотека классов содержит типы, поддерживающие многие специализированные сценарии разработки. На- пример, можно использовать платформу .NET Framework для разработки следующих типов приложений и служб:

* консольные приложения;
* приложения с графическим интерфейсом пользователя Windows (Windows Forms);
* приложения ASP.NET;
* веб-службы.

Архитектура платформы .NET Framework приведена на рис. 1.1.

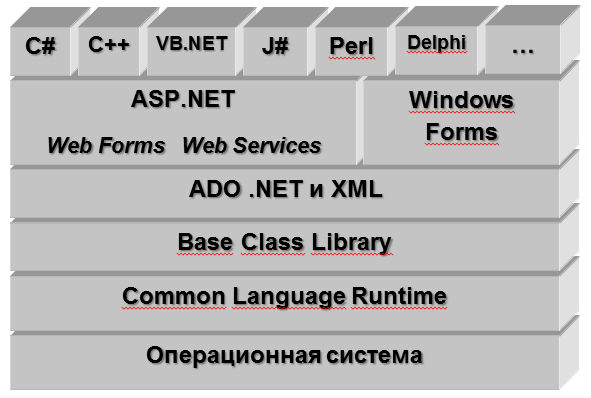


Рис. 1.1. Архитектура платформы .NET Framework

В состав платформы входят следующие компоненты (снизу вверх):

* Common Language Runtime (CLR) – среда управляемого исполнения кода .NET, реализуемая виртуальной машиной;
* Base Class Library – библиотека классов, используемая при создании пользовательских приложений. В ее состав входят средства ввода-вывода, сетевого взаимодействия, работа со строками, коллекциями объектов, средствами интернационализации и т. д.;
* ADO .NET и XML – средства обеспечения доступа к системам управления базами данных и обработки структурированной информации, представлен- ной на языке XML;
* ASP.NET – средства разработки приложений, размещаемых в сети Интернет и реализованных в виде распределенных сервис-ориентированных архитектур. Сюда же входят средства организации веб-интерфейса пользователя;
* Windows Forms – средства разработки графического интерфейса пользователя в приложениях. Здесь расположены элементы управления, размещаемые в пользовательских формах, обеспечивающих пользователю удобство работы с прикладными программами;
* языки программирования, поддерживающие спецификации .NET Framework и гарантирующие межъязыковую совместимость кода приложения. Различные языки введены для удобства пользователя.

## Разработка простого консольного приложения

Большинство типов приложений в Microsoft Visual Studio .NET может быть создано с использованием так называемого мастера – сценария, исполняемого средой разработки для создания структуры и заготовок кода приложений. Существует множество мастеров, применяемых для создания практически любых типов приложений.

В меню «Файл» выберите пункт «Новый проект». Появится окно, в котором будут перечислены все известные мастера, которые среда Microsoft Visual Studio .NET предоставляет для создания приложений различных типов. Для создания консольного приложения следует выбрать категорию мастеров Visual C# и в данной категории указать «Консольное приложение». Затем следует указать имя проекта и нажать OK.

После завершения работы мастер создает заготовку кода:

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

}

}

}

Приведенный код не выполняет никаких действий, он лишь содержит объявление используемых системных библиотек, а также заготовку класса Program со статическим методом Main, являющимся точкой входа в приложение.

Расширим этот пример стандартным сообщением «Hello, World!», выводимым в консоль. Для этого код дополняется следующими строками:

Console.WriteLine("Hello World!");

Console.ReadKey();

Для того чтобы собрать приложение, следует выбрать пункт меню «Сборка» (Build), а в нем команду «Собрать решение» (Build Solution). После построения программы компилятор сообщит о завершении или укажет ошибки в тексте программы.

## Задания к лабораторной работе

## Ввести имя. Вывести приветсвие человеку, вводившему имя.

## Ввести 2 числа. Вывести их сумму, разность, произведение и частное. Вывод осуществить одной командой вывода.

## Вводится двухзначное число. Вывести сумму его цифр.

1. В соответствии с вариантом необходимо написать простейшую программу.

***Программа должна вычислять:***

**Варианты:**

* + 1. Среднее арифметическое 2 вещественных чисел.
    2. Остаток от деления 2 вещественных чисел.
    3. Произведение 3 вещественных чисел.
    4. Частное 2 целых чисел.
    5. Сумму квадратов 2 вещественных чисел.
    6. Разность квадратов 2 вещественных чисел.
    7. Произведение квадратов 2 вещ-х чисел.
    8. Частное квадратов 2 вещественных чисел.
    9. Квадрат суммы 2 вещественных чисел.
    10. Квадрат разности 2 вещественных чисел.
    11. Квадрат произведения 2 вещественных чисел.
    12. Квадрат частного 2 вещественных чисел.
    13. Первое введённое число плюс квадрат второго.
    14. Сумму кубов 2 вещественных чисел.
    15. Разность кубов 2 вещественных чисел.

## 

## Контрольные вопросы

1. Какие основные компоненты входят в состав платформы Microsoft .NET Framework?
2. Какие основные элементы входят в состав среды разработки Microsoft Visual Studio .NET?
3. Как осуществляется считывание с консоли?
4. Кок организовать вывод на консоль?
5. Какие типы данных вы знаете? Использовали в программе?
6. Каким образом настроить папку для сохранения проектов?