

## Практическое занятие № 23, 24

### Программирование модуля. Создание библиотеки подпрограмм.

**Цель:** Получение практических навыков по собиранию собственной библиотеки – сборки (.dll); навыки работы с внешними библиотеками.

**Норма времени:** 2 часа.

**Оборудование:** Компьютер, среда программирования Visual Studio.

### Порядок выполнения работы

#### 1. Создание библиотек

Рано или поздно, возникает необходимость объединении наработок в единый пакет (или в несколько пакетов), чтобы созданные один раз классы можно было использовать в разных программах. В C# такие пакеты называют *сборками*. Сборки могут быть оформлены в виде DLL.

**Этап 1:** Необходимо создать новый проект в Visual Studio

Для этого, выбираем пункт главного меню «Файл — Создать — Проект», как показано на рисунке 1.

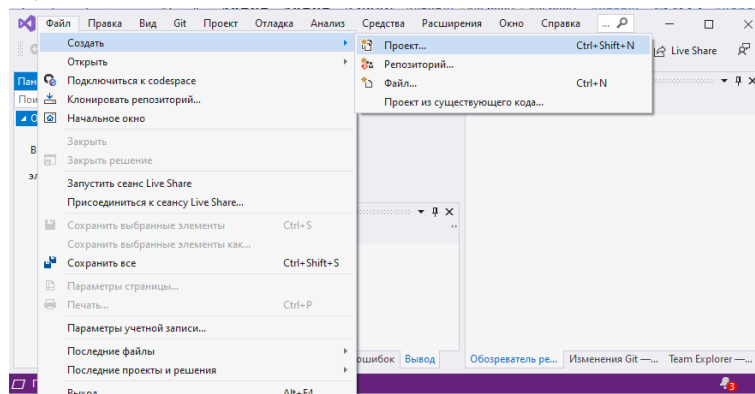


Рисунок 1. Создание нового проекта в MS Visual Studio.

После чего, откроется окно «Создание проекта», в котором нужно выбрать в левой части окна «Шаблоны — Visual C# — Windows», в главной рабочей области выбрать пункт «Библиотека классов», затем ввести имя библиотеки и указать расположение создаваемого проекта.

После указания всех параметров нужно нажать на кнопку «ОК» и дождаться создания проекта. После успешного создания проекта мы увидим примерно следующее окно (отмечены важные участки на изображении):

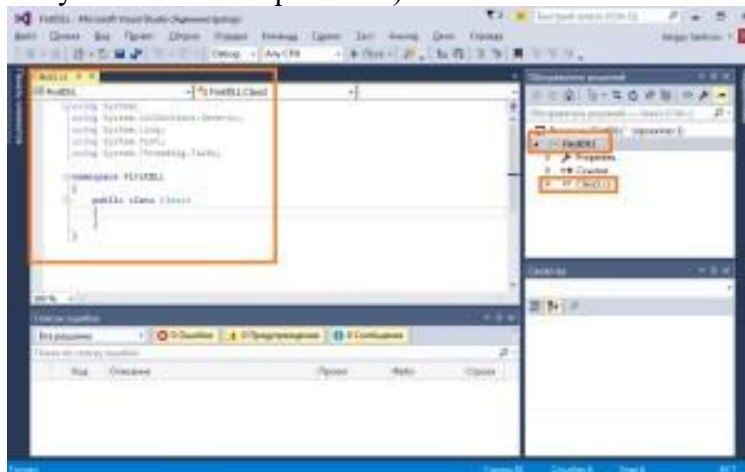


Рисунок 2. Созданный проект библиотеки

Мы видим, что у нас есть класс с именем «Class1» (это видно в обозревателе решений, или Solution Explorer в английской версии программы). Переименуем его, присвоим ему осмысленное название - «INIManager», в библиотеку мы поместим класс, из статьи [Работа с INI-файлами в C#](#), чтобы не придумывать что-то на ходу. Так вот, чтобы переименовать классы, нам нужно навести на его имя в обозревателе решений мышью и вызвать контекстное меню, в котором выбрать пункт «Переименовать», как показано на рисунке 3.

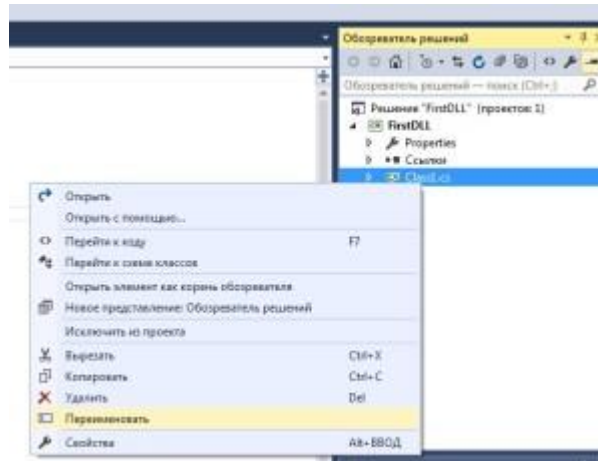
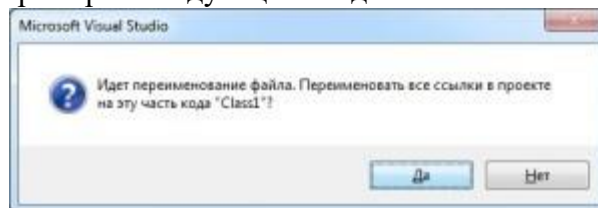


Рисунок 3. Переименование класса

После чего, в области редактирования (которая автоматически станет активной, ввести «INIManager» и нажать на клавишу Enter), в результате появится сообщение с запросом подтверждения примерно следующего вида:



В котором нужно нажать на кнопку «Да», таким образом, будет переименован и класс и файл исходного кода, в котором он содержится.

А теперь, вставим в класс следующий код:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Runtime.InteropServices;

namespace ClassLibrary1
{
    public class INIManager
    {
        //Конструктор, принимающий путь к INI-файлу
        public INIManager(string aPath)
        {
            path = aPath;
        }

        //Конструктор без аргументов (путь к INI-файлу нужно будет задать отдельно)
        public INIManager() : this("") { }

        //Возвращает значение из INI-файла (по указанным секции и ключу)
        public string GetPrivateString(string aSection, string aKey)
        {
            //Для получения значения
        }
    }
}
```

```

        StringBuilder buffer = new StringBuilder(SIZE);

        //Получить значение в buffer
        GetPrivateString(aSection, aKey, null, buffer, SIZE, path);

        //Вернуть полученное значение
        return buffer.ToString();
    }

    //Пишет значение в INI-файл (по указанным секции и ключу)
    public void WritePrivateString(string aSection, string aKey, string aValue)
    {
        //Записать значение в INI-файл
        WritePrivateString(aSection, aKey, aValue, path);
    }

    //Возвращает или устанавливает путь к INI файлу
    public string Path { get { return path; } set { path = value; } }

    //Поля класса
    private const int SIZE = 1024; //Максимальный размер (для чтения значения из
файла)
    private string path = null; //Для хранения пути к INI-файлу

    //Импорт функции GetPrivateProfileString (для чтения значений) из библиотеки
kernel32.dll
    [DllImport("kernel32.dll", EntryPoint = "GetPrivateProfileString")]
    private static extern int GetPrivateString(string section, string key, string
def, StringBuilder buffer, int size, string path);

    //Импорт функции WritePrivateProfileString (для записи значений) из библиотеки
kernel32.dll
    [DllImport("kernel32.dll", EntryPoint = "WritePrivateProfileString")]
    private static extern int WritePrivateString(string section, string key, string
str, string path);
}
}

```

И в самом начале файла с классом, в блоке директив using, подключить пространство имен «System.Runtime.InteropServices» (добавленная строка отмечена):

Всё, теперь можно собрать проект («Сборка — Собрать решение» в главном меню русской версии Visual Studio). Только запускать проект не нужно, всё равно это бессмысленно, DLL так просто не запустить.

Необходимо найти созданную DLL на диске своего ПК. Для этого нужно перейти в папку, в которой мы создавали проект (путь мы указывали при создании проекта). В моем случае, это «C:\Users\Пользователь\source\repos\ClassLibrary1\bin\Debug», и в папке

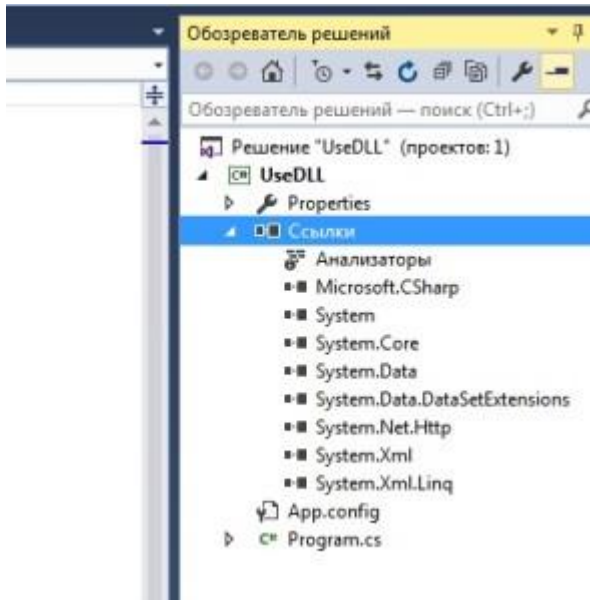


«Debug» будет файл «ClassLibrary1.dll». Который нужно скопировать и сохранить в удобное место, чтобы потом подключить к другому проекту.

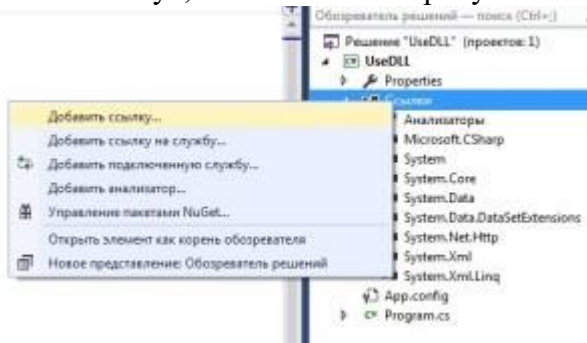
***Первую библиотеку мы создали!***

**Этап 2:** Создадим новый проект консольного приложения.

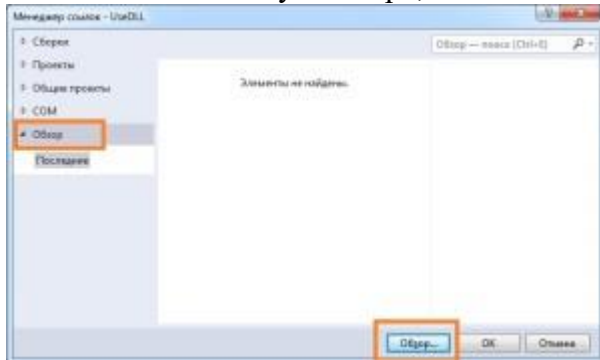
В окне Visual Studio, в обозревателе решений, ищем группу «Ссылки» и раскрываем её, как показано на рисунке ниже.



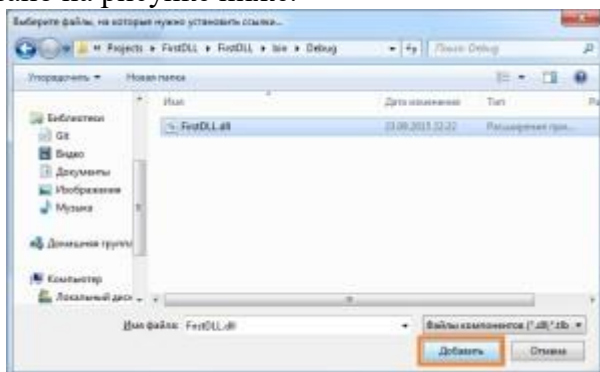
А теперь, вызовем контекстное меню на пункте «Ссылки», и выберем в нем «Добавить ссылку», как показано на рисунке ниже.



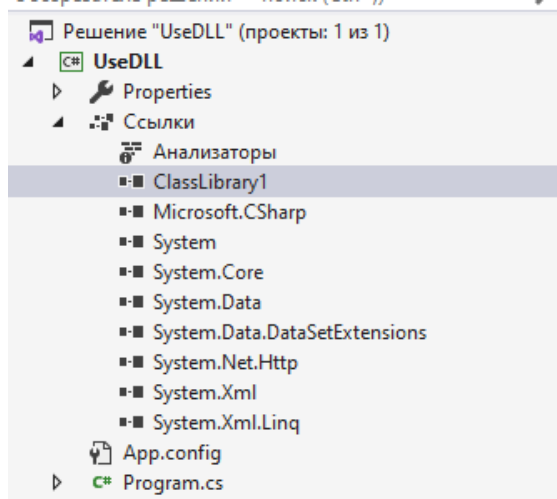
В появившемся окне, выбираем, в левой области выбираем пункт «Обзор» и в низу окна, нажимаем на кнопку «Обзор», как показано на рисунке ниже.



В появившемся окне, перейти в папку, в которое лежит библиотека (DLL) созданная в предыдущей этапе, выбрать эту библиотеку и нажать на кнопку «Добавить», как показано на рисунке ниже.



После чего, нажать на кнопку «ОК» в предыдущем окне. В результате, список ссылок проекта, будет пополнен еще одной:



Теперь, мы можем использовать в нашей программе класс «INIManager» из подключенной библиотеки. Но перед этим, мы должны выполнить еще одну операцию. Рассмотрим фрагмент кода подключенной библиотеки (он остался у нас после предыдущего шага). Обратите внимание на выделенную строку, это объявление пространства имен (namespace), как бы некоего контейнера, в котором находится класс «INIManager». И мы не сможем воспользоваться классом, пока не укажем системе что нужно взять во внимание это пространство имен. Т.е. нам нужно подключить пространство имен в нашем проекте. Для этого, добавим строку «using ClassLibrary1;» в конец блока директив using, который расположен в самом начале основного файла проекта.

Вот теперь, можно использовать класс из подключенной библиотеки, а сделать это можно написав в методе «Main» такой код:

```
//Создание объекта, для работы с файлом
    INIManager manager = new INIManager("C:\\my.ini");

    //Получить значение по ключу name из секции main
    string name = manager.GetPrivateString("main", "name");

    //Вывести полученное имя в консоль
    Console.WriteLine(name);

    //Записать значение по ключу age в секции main
    manager.WritePrivateString("main", "age", "21");
    Console.ReadKey();
```

Данный код будет работать, при условии наличия файла «my.ini» на диске «С» нашего ПК. Вот такой вот пример работы со внешними библиотеками.

**Задание для самостоятельного выполнения:** Выполните индивидуальные задания. Сохраните проекты. Создать приложение для решения задачи, формулы для расчета располагаются в модуле.

**Вариант 1.** Вычислить площадь и периметр прямоугольника, если задана длина одной стороны (a) и коэффициент n (%), позволяющий вычислить длину второй стороны (b=n\*a).

**Вариант 2.** Вычислить объем призмы, боковые грани которой квадраты, а основанием служит равносторонний треугольник, вписанный в круг радиуса R.

**Вариант 3.** Удалить часть символьной строки, заключенной в скобки (вместе со скобками).

**Вариант 4.** Дан зашифрованный русский текст. Каждая буква заменяется на следующую за ней (буква я заменяется на а). Получить новый текст, содержащий расшифровку данного текста.

**Вариант 5.** Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.

**Вариант 6.** Треугольник задан тремя сторонами. Вычислить его медианы.

**Вариант 7.** Определить, сколько раз в строке встречается заданное слово.

**Вариант 8.** Дан текст на русском языке и некоторые два слова. Определить, сколько раз они входят в текст и сколько раз они входят непосредственно друг за другом.

**Вариант 9.** Вычислить периметр и площадь прямоугольного треугольника по заданным длинам двух катетов  $a$  и  $b$ .

**Вариант 10.** В шар радиуса  $R$  вписан конус с углом  $\alpha$  при вершине в осевом сечении конуса. Определить объем и полную поверхность конуса.

**Вариант 11.** Проверить, одинаковое ли число открывающихся и закрывающихся скобок в данной строке.

**Вариант 12.** Дан произвольный текст. Проверить, правильно ли в нем расставлены круглые скобки.

**Вариант 13.** Вычислить площади геометрических фигур: прямоугольника и треугольника по заданным сторонам.

**Вариант 14.** Вычислить диагональ и площадь прямоугольника, вписанного в окружность радиуса  $R$ , если отношение его сторон равно  $n$ .

**Вариант 15.** Строка содержит произвольный русский текст. Проверить каких букв в нем больше: гласных или согласных.

**Вариант 16.** Дан текст на русском языке и некоторая буква. Найти слово, содержащее наибольшее количество указанных букв.

**Вариант 17.** По известному радиусу вычислить объем и площадь поверхности шара.

**Вариант 18.** Даны две стороны треугольника и угол между ними. Определить третью сторону, площадь треугольника и радиус описанной окружности.

**Вариант 19.** В строке имеется одна точка с запятой (;). Подсчитать количество символов до точки с запятой и после нее.

**Вариант 20.** Дан текст на русском языке. Подсчитать количество слов, начинающихся и заканчивающихся на одну и ту же букву.

**Вариант 21.** Даны два числа. Вычислить их сумму, разность, произведение и частное.

**Вариант 22.** Определить плату за электроэнергию, если известны: старое и новое показания счетчика, стоимость одного квт/часа электроэнергии, количество просроченных дней и размер пени за один день просрочки.

**Вариант 23.** Дан текст на русском языке и некоторая буква. Найти слово, содержащее наибольшее количество указанных букв.

**Вариант 24.** Дан текст. Сколько слов в тексте? Сколько цифр в тексте?

### **Контрольные вопросы**

1. Чем объясняется широкое использование модулей?
2. Какую структуру имеет модуль?
3. Может ли имя процедуры в вызывающем приложении не совпадать с именем процедуры в библиотеке?
4. По какому признаку компилятор определяет, что код процедуры расположен вне приложения?
5. В чём принципиальное отличие динамического импорта от статического?

