

Пользовательская документация по использованию библиотеки «lib.dll»

1. Используемые программные компоненты при разработке библиотеки:

Операционная система: Windows 10

Редактор кода: Visual Studio 2019

Платформа: .Net Core 3.1

2. Подготовка к использованию

- 1) Для начала работы с библиотекой подключите ее в References (Зависимости).
Кликните ПКМ по «Зависимости» в Обозревателе решений и нажмите «Добавить ссылку на проект». Откроется диалоговое окно «Диспетчер ссылок»:

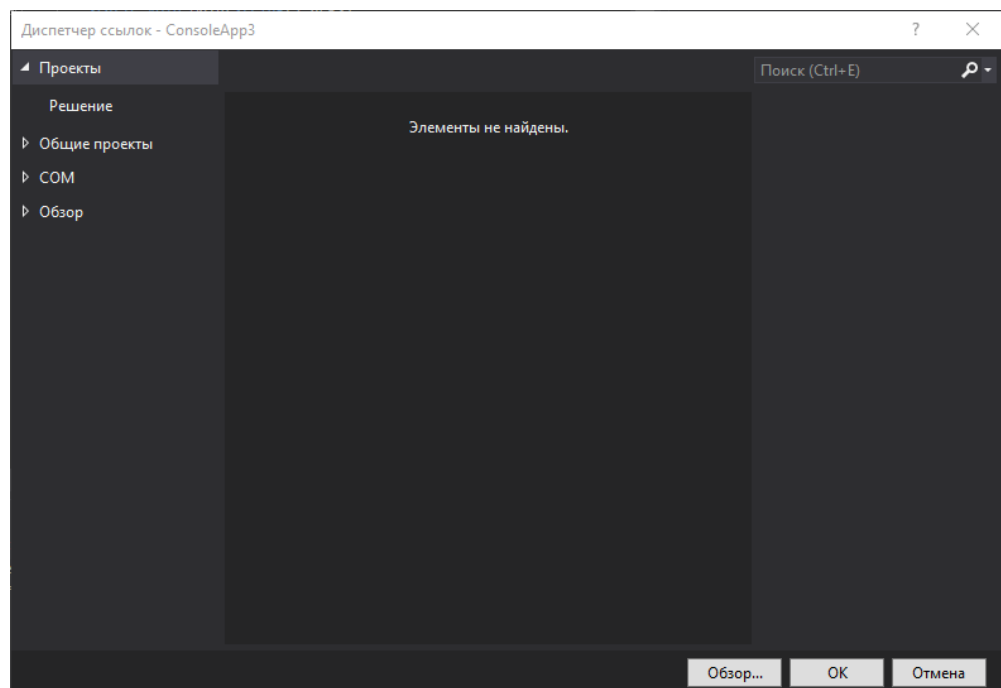


Рисунок 1 – Диспетчер ссылок

- 2) Нажмите кнопку обзор и найдите установленный проект и перейдите в нем по пути: bin\Debug\netcoreapp3.1. Выберите файл lib.dll и нажмите «Добавить» и затем кнопку «Ок» в Диспетчере ссылок.
- 3) В зависимостях должна появиться ссылка на сборку:

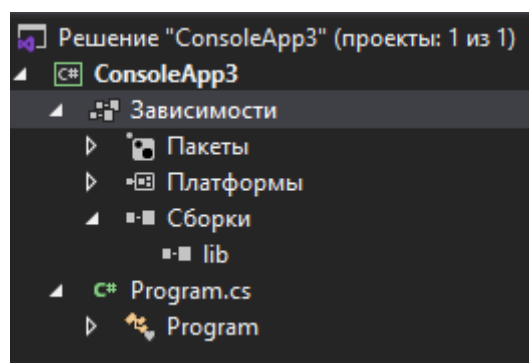


Рисунок 2 – Библиотека lib.dll в проекте

- 4) Теперь необходимо установить два NuGet пакета. Для этого открываем консоль диспетчера пакетов VS 2019 и прописываем следующие команды:

А) Установка System.Drawing.Common*:

```
PM> Install-Package System.Drawing.Common
Идет восстановление пакетов для C:\Users\Даниил\source\repos\ConsoleApp3
\ConsoleApp3.csproj...
Установка пакета NuGet System.Drawing.Common 4.7.0.
Фиксация восстановления...
Запись файла ресурсов на диск. Путь: C:\Users\Даниил\source\repos\ConsoleApp3\obj
\project.assets.json
Восстановлен C:\Users\Даниил\source\repos\ConsoleApp3\ConsoleApp3.csproj (за 20 ms).
"Microsoft.NETCore.Platforms 3.1.0" успешно установлено в ConsoleApp3
"Microsoft.Win32.SystemEvents 4.7.0" успешно установлено в ConsoleApp3
"System.Drawing.Common 4.7.0" успешно установлено в ConsoleApp3
Выполнение действий NuGet заняло 654 ms
Прошло времени: 00:00:01.3047335
```

Рисунок 3 – Установка пакета System.Drawing.Common

* С недавнего времени Image находится в сборке System.Drawing.Common, поэтому может потребоваться установка данного пакета.

Б) Установка System.Resources.Extensions**:

```
PM> Install-Package System.Resources.Extensions
Идет восстановление пакетов для C:\Users\Даниил\source\repos\ConsoleApp3
\ConsoleApp3.csproj...
Установка пакета NuGet System.Resources.Extensions 4.7.1.
Фиксация восстановления...
Запись файла ресурсов на диск. Путь: C:\Users\Даниил\source\repos\ConsoleApp3\obj
\project.assets.json
Восстановлен C:\Users\Даниил\source\repos\ConsoleApp3\ConsoleApp3.csproj (за 18 ms).
"System.Resources.Extensions 4.7.1" успешно установлено в ConsoleApp3
Выполнение действий NuGet заняло 296 ms
Прошло времени: 00:00:00.5807470
```

Рисунок 4 – Установка пакета System.Resources.Extensions

** Эта пакет необходим, т.к. в файлах ресурсов библиотека хранятся изображения.

- 5) Подключите вверху вашего рабочего файла пространство имен ds.test.impl

```

using System;
using ds.test.impl;

namespace Example
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}

```

Рисунок 5 – Подключение пространства имен ds.temp.impl

- 6) Поздравляю! Вы настроили библиотеку и можете приступать к ее использованию!

3. Структура библиотеки

Для использования Вам доступны статический класс Plugins и интерфейс IPlugin. Также в программу были добавлены свои исключения LogPluginException и NullPluginException.

- 1) Статический класс Plugins:

Таблица 1. Описание методов и свойств Plugins

Метод/Свойство	Описание
Свойство int PluginsCount	Возвращает количество, имеющих в системе плагинов
Свойство string[] GetPluginNames	Возвращает строковый массив с именами всех доступных плагинов.
Метод IPlugin GetPlugin(stringName)	Возвращает экземпляр IPlugin по заданному имени плагин. Если плагина с имеющимся видом не существует, возвращает NullPluginException

- 2) Интерфейс IPlugin:

Таблица 2. Описание методов и свойств IPlugin

Метод/Свойство	Описание
Свойство string PluginName	Имя используемого плагина
Свойство String Version	Текущую версия используемого плагина

Свойство Image Image	Картинка используемого плагина
Свойство string Description	Описание плагина (полезно для использования, если непонятно, что выполняет какой-либо плагин)
Метод int Run(int input1, int input2)	Возвращает целочисленное значение, являющееся результатом выполненного действия для 2 входных переменных
Метод double Run(double input1, double input2)	Возвращает целочисленное значение, являющееся результатом выполненного действия для 2 входных переменных

3) Исключение NullPluginException:

Данное исключение может появляться при попытке вызова функции Plugins.GetPlugin(string pluginName) в случае, если не существует плагина с именем pluginName.

4) Исключение LogPluginException:

Исключение позволяет отловить ошибку некорректно введенных данных input1 и input2 при попытке вызова функции Run(input1, input2).

4. Пример программы

Данная программа демонстрирует основной функционал. Данный проект вы можете найти в репозитории на github в папке Example.

Обозреватель объектов Program.cs

Example Example.Program Main(string[] args)

```
1 using System;
2 using ds.test.impl;
3
4 namespace Example
5 {
6     class Program
7     {
8         static void Main(string[] args)
9         {
10             // Получаем строковый массив всех имен плагинов
11             string[] pluginNames = Plugins.GetPluginNames;
12
13             for(int i = 0; i < pluginNames.Length; i++)
14             {
15                 // Получаем плагин с определенным именем
16                 var plugin = Plugins.GetPlugin(pluginNames[i]);
17                 // Выводим в консоль описание и результат метода Run
18                 // для каждого плагина
19                 Console.WriteLine(plugin.Description);
20                 Console.WriteLine(plugin.Run(2, 2));
21             }
22
23             try
24             {
25                 // Пытаемся получить несуществующий плагин
26                 var plugin = Plugins.GetPlugin("qwe");
27                 Console.WriteLine(plugin.Description);
28             }
29             catch(NullPluginException ex)
30             {
31                 Console.WriteLine(ex.Message);
32             }
33         }
34     }
35 }
36
```

Рисунок 6 – Код программы

