# lib.dll

lib.dll – библиотека, используемая в космосе для решения несложных математических и не только проблем.

## Классы и методы

### Статический класс Plugins

**Описание:** предназначен для создания объектов IPlugin

|  |  |
| --- | --- |
| **Свойства и методы** | **Значение** |
| *static int PluginsCount { get; }* | Вернёт количество существующих плагинов |
| *static string[] GetPluginNames{ get; }* | Вернёт имена всевозможных плагинов |
| *static IPlugin GetPlugin(string pluginName)* | Вернёт плагин, указанный в pluginName |

### Интерфейс IPlugin

**Описание:** Абстрактный класс Plugin реализует данный интерфейс.

|  |  |
| --- | --- |
| **Свойства и методы** | **Значение** |
| *string PluginName { get; }* | Имя плагина |
| *string Version { get; }* | Версия плагина |
| *System.Drawing.Image Image { get; }* | Картинка плагина |
| *string Description { get; }* | Описание плагина |
| *int Run (int input1, int input2)* | Действие плагина |

### Абстрактный класс Plugin

**Описание:** реализует интерфейс IPlugin, а также свойство Image

**Реализация абстрактного класса:**

*public abstract string PluginName { get; }*

*public abstract string Version { get; }*

*public Image Image { get; }*

*public virtual string Description => "Plugin Description : ";*

*public abstract int Run(int input1, int input2)*

### Плагин AddPlugin

**Описание**: сложит два числа, указанные в аргументах метода Run

|  |  |
| --- | --- |
| **Методы** | **Значение** |
| *public override int Run(int input1, int input2)* | Складывает значения input1 и input2. Выдаст ошибку переполнения в случае переполнения |

**Пример:**

1. IPlugin Plugin = Plugins.GetPlugin("AddPlugin");
2. int value1 = 3;
3. int value2 = 3;
4. int result = Plugin.Run(value1, value2);
5. // result = 6;

### Плагин MultiplyPlugin

**Описание**: умножит два числа, указанные в аргументах метода Run

|  |  |
| --- | --- |
| **Методы** | **Значение** |
| *public override int Run(int input1, int input2)* | Умножит значения input1 и input2. Выдаст ошибку переполнения в случае переполнения |

**Пример:**

1. IPlugin Plugin = Plugins.GetPlugin("MultiplyPlugin");
2. int value1 = 3;
3. int value2 = 3;
4. int result = Plugin.Run(value1, value2);
5. // result = 9;

### Плагин DividePlugin

**Описание**: разделит два числа, указанные в аргументах метода Run

|  |  |
| --- | --- |
| **Методы** | **Значение** |
| *public override int Run(int input1, int input2)* | Разделит значение input1 на input2. Выдаст ошибку потери точности в случае не целого деления |

**Пример:**

1. IPlugin Plugin = Plugins.GetPlugin("DividePlugin");
2. int value1 = 3;
3. int value2 = 3;
4. int result = Plugin.Run(value1, value2);
5. // result = 1;

### Плагин DivideWithRoundPlugin

**Описание**: разделит два числа, указанные в аргументах метода Run и применит округление

|  |  |
| --- | --- |
| **Методы** | **Значение** |
| *public override int Run(int input1, int input2)* | Разделит значение input1 на input2. Не выдаст ошибку потери точности в случае не целого деления |

**Пример:**

1. IPlugin Plugin = Plugins.GetPlugin("DivideWithRoundPlugin");
2. int value1 = 3;
3. int value2 = 3;
4. int result = Plugin.Run(value1, value2);
5. // result = 1;

### Плагин PowPlugin

**Описание**: возведёт число в степень

|  |  |
| --- | --- |
| **Методы** | **Значение** |
| *public override int Run(int input1, int input2)* | Возведёт значение input1 в степень input2. Выдаст ошибку если значение input2 будет меньше 0. |

**Пример:**

1. IPlugin Plugin = Plugins.GetPlugin("PowPlugin");
2. int value1 = 3;
3. int value2 = 3;
4. int result = Plugin.Run(value1, value2);
5. // result = 27;

### Плагин UltimateAnswerOfLifeAndUniverseAndEverythingPlugin

**Описание**: поможет экипажу найти ответ на главный вопрос жизни, вселенной и всего такого.

|  |  |
| --- | --- |
| **Методы** | **Значение** |
| *public override int Run(int input1, int input2)* | Ответ может всех огорчить. Вернёт значение 42 |

**Пример:**

1. IPlugin Plugin = Plugins.GetPlugin("UltimateAnswerOfLifeAndUniverseAndEverythingPlugin");
2. int value1 = 3; int value2 = 3;
3. int result = Plugin.Run(value1, value2);
4. // result = 42;