

Задание

1. Для изучения .NET Reflection API напишите статический класс **Reflector**, который собирает информацию и будет содержать методы выполняющие исследования класса (принимают в качестве параметра имя класса) и записывающие информацию в файл (формат тестовый, json или xml):

- a. Определение имени сборки, в которой определен класс;
- b. есть ли публичные конструкторы;
- c. извлекает все общедоступные публичные методы класса (возвращает *IEnumerable<string>*);
- d. получает информацию о полях и свойствах класса (возвращает *IEnumerable<string>*);
- e. получает все реализованные классом интерфейсы (возвращает *IEnumerable<string>*);
- f. выводит по имени класса имена методов, которые содержат заданный (пользователем) тип параметра (имя класса передается в качестве аргумента);
- g. метод **Invoke**, который вызывает метод класса, при этом значения для его параметров необходимо 1) прочитать из текстового файла (имя класса и имя метода передаются в качестве аргументов) 2) сгенерировать, используя генератор значений для каждого типа. Параметрами метода **Invoke** должны быть : объект, имя метода, массив параметров.

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/reflection>

Продемонстрируйте работу «Рефлектора» для исследования типов на созданных вами классах не менее двух (предыдущие лабораторные работы) и стандартных классах .Net.

2. Добавьте в **Reflector** обобщенный метод **Create**, который создает объект переданного типа (на основе имеющихся публичных конструкторов) и возвращает его пользователю.

Вопросы

1. Что такое рефлексия?
2. Какие есть способы получения объекта типа Type. Приведите три способа получения типа.
3. Охарактеризуйте классы из пространства имен *System.Reflection*.
4. Как можно использовать класс *System.Type*? Перечислите его свойства и методы.
5. Что такое позднее и раннее связывание?
6. Как динамически загрузить сборку в приложение?
7. Что такое позднее (раннее) связывание?
8. Для чего предназначены *BindingFlags*? Какую комбинацию флагов необходимо использовать, чтобы иметь возможность получать приватные члены класса?