

#12 Рефлексия

Задание:

Note: для тех, кто выполнял повышенное задание в л.р. #8 повторяющиеся пункты можно пропустить и расширить ваш Reflector недостающими методами. (начиная с пункта 6).

1. Ознакомьтесь с концепцией рефлексии в .NET:
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/reflection>
2. Создайте структуру, которая будет хранить информацию о классе:
 - a. Есть ли публичный конструктор? (*bool*)
 - b. Имя типа (*string*)
 - c. Имя сборки, в которой он определен (*string*)
 - d. Имена публичных полей (*IEnumerable<string>*)
 - e. Имена публичных свойств (*IEnumerable<string>*)
 - f. Имена приватных полей (*IEnumerable<string>*)
 - g. Имена приватных свойств (*IEnumerable<string>*)
 - h. Имена публичных методов (*IEnumerable<string>*)
 - i. Имена приватных методов (*IEnumerable<string>*)
3. Добавьте два новых свойства, описывающих любую иную информацию об объекте (на ваш выбор).
4. Создайте статический класс **Reflector**, который будет собирать информацию, используя рефлексия.
5. Реализуйте в классе метод **Analyze**, собирающий информацию об объектах классов (*И СТРУКТУР*) (используя структуру выше) и выводящий ее в json-файл. Метод должен принимать 2 параметра:
 - a. Объект для сбора информации;
 - b. Путь json-файла для сохранения информации.
6. Добавьте в **Reflector** обобщенный метод **Create**, который создает объект переданного типа и возвращает его пользователю.
7. Добавьте в **Reflector** обобщенный метод **Invoke**, который вызывает на переданном объекте указанный метод. При этом:
 - a. Параметры метода:
 - i. Объект
 - ii. Имя метода
 - iii. Массив параметров
 - b. **Invoke** проверяет наличие метода у объекта. При его отсутствии – выбрасывается исключение.

8. Добавьте обобщенную перегрузку метода **Invoke**, которая создает в себе объект переданного типа, используя метод **Create**, и далее повторяет процедуру описанного **Invoke** выше.
9. Добавьте перегрузки для этих двух методов, принимающие дополнительно **BindingFlags** и использующие их при поиске метода в объекте.
10. Используйте *try-catch*.
11. Не допускайте дублирование кода. Если у двух методов есть общее тело, которые незначительно различаются, то такой код можно вынести в отдельный метод.

Вопросы:

1. Что такое рефлексия?
2. Какие методы и классы вы использовали при выполнении работы, которые относятся к рефлексии? Охарактеризуйте их.
3. Перечислите методы и свойства класса *System.Type*.
4. Что такое позднее и раннее связывание?
5. Для чего предназначены **BindingFlags**? Какую комбинацию флагов необходимо использовать, чтобы иметь возможность получать приватные члены класса?