

Задача 3. Проверка заполненности данных (атрибуты)

Часто бывает нужным проверить заполняемость обязательных данных в объекте. Для этого, конечно, можно написать отдельный метод, однако, можно пойти и другим путем – декларативным, т.е. через атрибуты. В текущей работе вам потребуется определить 1) собственный атрибут, которым вы будете помечать обязательные для заполнения данные в произвольных типах и 2) метод для проверки заполненности данных объекта.

Определить пользовательский атрибут с именем `MyRequiredAttribute`, которым можно помечать поля и свойства любых типов для проверки их заполненности (строковые данные не должны быть пусты - `string.IsNullOrEmpty`, другие типы не должны содержать умалчиваемое значение - `default(T)`). У атрибута предусмотреть параметр, отвечающий за сообщение, которое будет формироваться для описания ошибок при проверке заполненности данных.

Реализовать метод `string MyValidate()`, расширяющий тип `object` и возвращающий строку с сообщением об ошибке проверки заполненности данных объекта (пустая строка, если нет ошибок).

Создать три произвольных типа: структуру и два класса. Поля и свойства в них разных типов объявить по своему усмотрению, часть из них - обязательную для заполнения - пометить атрибутом `MyRequiredAttribute`.

Создать объекты и протестировать работу метода `MyValidate()`.

Полезная заготовка кода для сравнения значения свойства с умалчиваемым:

```
PropertyInfo prop = ...;
object obj = ...;
if (prop.PropertyType == typeof(string) &&
    string.IsNullOrEmpty((string)t.GetProperty(prop.Name).GetValue(obj))
    ||
    t.GetProperty(prop.Name).GetValue(obj).Equals(GetDefault(prop.PropertyType)))
    return ""
```

Полезная заготовка кода для получения умалчиваемого значения для любого типа:

```
public static object GetDefault(Type type)
{
    if (type.IsValueType)
    {
        return Activator.CreateInstance(type);
    }
    return null;
}
```

Дополнительное задание

Подумайте над тем, что приведенная в этом примере валидация является довольно примитивной. При необходимости реализации нетривиальных проверок можно осуществлять поиск в типе специального пользовательского метода-валидатора (`string AnyName()`), помеченного другим специальным атрибутом, например, `MyCustomValidationAttribute`. При его нахождении – необходимо выполнить его и вернуть полученный от него результат валидации, если метод, помеченный атрибутом не найден – выполнять те же проверки, что и раньше.