**Задание 1.** Создайте следующий набор типов:

* Атрибут [TrackingEntity], который может применяться к классу или структуре.
* Атрибут [TrackingProperty], который может применяться к полю или свойству и имеет собственное свойство PropertyName.
* Класс Logger с методом Track(), получающим в качестве параметра произвольный объект. Если у этого объекта обнаруживается атрибут [TrackingEntity], то метод записывает в JSON-файл в виде JSON-объекта значения тех экземплярных свойств и полей объекта, которые помечены атрибутом [TrackingProperty]. Значения фиксируются как пары «имя: значение», где «имя» берется из атрибута [TrackingProperty]. Если там имя не задано (пустое), то берётся имя свойства или поля, к которому применен атрибут [TrackingProperty]. Имя самого JSON-файла указывается как параметр конструктора класса Logger.

Разместите созданные типы в отдельной библиотеке классов. В консольном приложении подключите эту библиотеку классов и протестируйте работу типов из неё.

**Задание 2.** *Механизм привязки* позволяет создать объект указанного типа и заполнить его информацией на основе данных строкового словаря. Разработайте и протестируйте метод Bind(), выполняющий привязку данных.

1. Методу Bind() необходимо передать тип, экземпляр которого возвращается методом. Требование к типу – имеет конструктор без параметров.
2. Метод Bind() получает в качестве параметра словарь, где ключами и значениями являются строки. Это набор данных для привязки.
3. Метод Bind() создаёт объект указанного типа и выполняет поиск значений в словаре для экземплярных полей и свойств этого объекта (имя поля или свойства = ключ в словаре). Сделайте поиск регистронезависимым. Для простоты работайте только с полями и свойствами с типами int, double, string.
4. При ошибках привязки исключений не генерировать.
5. Разместите метод Bind() в классе SimpleBinder, реализующем шаблон *Одиночка* (Singleton).