

Сравнение объектов

Написать реализацию интерфейса `IComparer` для глубокого сравнения объектов:

```
interface IComparer
{
    IEnumerable<Difference> Compare<T>(T First, T Second);
}
```

Класс должен принимать объекты неизвестного заранее типа `T` и выдавать список различающихся публичных, доступных для чтения свойств. Про тип `T` известно, что его свойства могут быть:

1. Простых типов (`string`, `int`, `double`)
2. Объектами типов, для которых выполняются те же условия, которые выполняются для типа `T`.

Класс `Difference` предлагается подобрать самостоятельно с учётом того, что он должен содержать путь до различающегося свойства и значения объектов `First` и `Second`.

Например для классов

```
class Person
{
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public Address Address { get; set; }
}

class Address
{
    public string City { get; set; }
    public string Street { get; set; }
    public int House { get; set; }
}
```

при вызове

```
var first =
    new Person
    {
        FirstName = "Иван",
        LastName = "Иванов",
        Address =
            new Address
```

```

        {
            City = "Екатеринбург",
            Street = "Ленина",
            House = 1
        }
    }

var first =
    new Person
    {
        FirstName = "Иван",
        LastName = "Сидоров",
        Address =
            new Address
            {
                City = "Екатеринбург",
                Street = "Малышева",
                House = 4
            }
    }

var differences = comparer.Compare<Person>(first, second);

```

результат должен быть следующим:

Путь	First	Second
LastName	Иванов	Сидоров
Address.Street	Ленина	Малышева
Address.House	1	4

Так же нужно составить модульные тесты для проверки реализации.