

#6 Структуры, перечисления.

Задание:

1. К предыдущей лабораторной работе (#5) добавьте перечисление.
2. Ознакомьтесь с методами класса **Enum**, в особенности с теми, которые служат для извлечения/преобразования значений перечисления:
<https://docs.microsoft.com/en-us/dotnet/api/system.enum?view=netframework-4.8>
3. Преобразуйте класс-коллекцию из л.р. #5 в *generic*-класс на основе выбранного вами базового класса или интерфейса.
4. Преобразуйте ваш класс-коллекцию в т.н. *класс-контейнер*, который:
 - a. Хранит данные;
 - b. Управляет данными (добавление, удаление);
 - c. Осуществляет проверку валидности принимаемых значений.
5. Создайте т.н. *класс-контроллер*, который наследует от *класса-контейнера* существующую функциональность и расширяет ее:
 - a. Переопределяет 2 любых оператора (на ваш выбор как сами операторы, так и то, что они будут делать);
 - b. Содержит методы поиска **FirstOrDefault** и **FindAll**, которые принимают предикат и возвращают:
 - i. **FirstOrDefault** – первый элемент коллекции, удовлетворяющий условию и **null**, если таковых нет;
 - ii. **FindAll** – все элементы коллекции, удовлетворяющие данному условию.
 - c. Реализует любой из стандартных интерфейсов: **IEnumerable**, **IDictionary**, **IList**, **ICollection**, **IQueryable** (можно найти любой свой из стандартной библиотеки)
Note: некоторые интерфейсы являются *generic*. В этом случае параметризовать их следует типом вашего класса (не коллекции, не контейнера)
6. Добавьте классу-контроллеру структуру, содержащую имя автора программы и текущую дату. Инициализируйте объект структуры вместе с созданием экземпляра класса-контроллера.

Повышенный уровень:

1. Добавьте в класс-контроллер метод, считывающий построчно текстовый файл, в котором хранятся данные вашего класса и инициализирует таким образом коллекцию.
2. Ознакомьтесь с библиотекой **Newtonsoft**:
<https://www.newtonsoft.com/json>
3. Реализуйте еще один метод, который будет считывать данные из **json**-файла и инициализировать коллекцию. Пример работы показан **здесь** (кликабельно)

Вопросы:

1. Что такое структура? В чем различия между классом и структурой?
2. Может ли структура содержать вложенные члены (классы, структуры)?
3. Что такое перечисления?
4. Что такое полиморфизм? Какие существуют виды полиморфизма? Какие из них реализованы/поддерживаются .NET?
5. Расскажите про стандартный интерфейс, который вы использовали. Будьте готовы к вопросам по другим стандартным интерфейсам: [ref](#) (кликабельно)

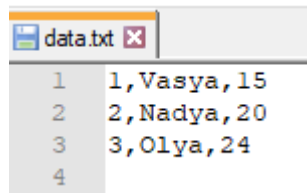
Повышенный уровень:

1. Что такое **JSON**? В чем заключаются «плюсы» **JSON** перед обычным текстовым форматом?
2. Для чего служит библиотека **Newtonsoft**? Расскажите про методы библиотеки, которые вы использовали в работе.

Пример объявления класса:

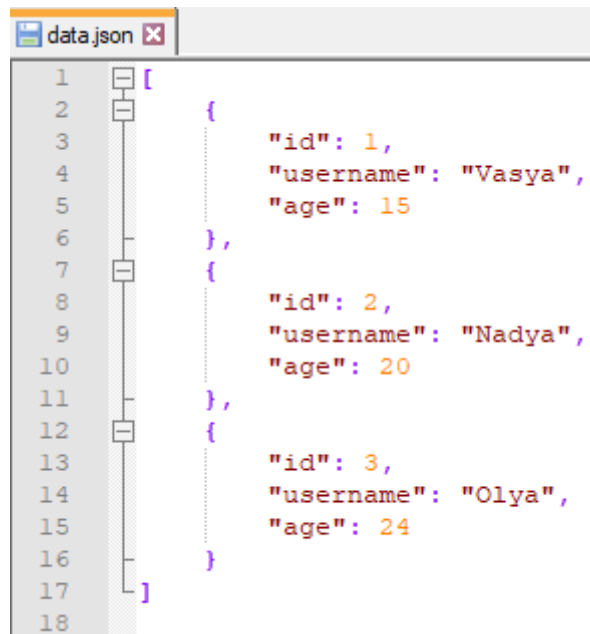
```
1 reference
class User
{
    1 reference
    public int Id { get; set; }
    1 reference
    public string Username { get; set; }
    1 reference
    public int Age { get; set; }
}
```

Пример текстового файла с данными (разделитель может быть любым):



id	username	age
1	Vasya	15
2	Nadya	20
3	Olya	24

Пример json-файла с данными:



```
[
  {
    "id": 1,
    "username": "Vasya",
    "age": 15
  },
  {
    "id": 2,
    "username": "Nadya",
    "age": 20
  },
  {
    "id": 3,
    "username": "Olya",
    "age": 24
  }
]
```