

Практическая работа №15

Сериализация и десериализация данных в формате XML

1 Цель работы

1.1 Научиться выполнять сериализацию и десериализацию данных в формате XML в приложениях на C#.

2 Литература

2.1 <https://docs.microsoft.com/ru-ru/dotnet/standard/serialization/examples-of-xml-serialization>

2.2 <https://metanit.com/sharp/tutorial/6.4.php>

3 Задание

3.1 Сериализация объектов класса с полями простых типов данных.

3.1.1 Создать класс для хранения имени и возраста человека.

Для хранения данных в классе вместо полей использовать открытые автореализуемые свойства:

```
public тип ИмяСвойства { get; set; }
```

3.1.2 Создать в основной программ:

- массив объектов созданного класса,

- список объектов созданного класса.

Для инициализации свойств при создании объекта можно использовать следующий код:

```
ИмяКласса переменная = new ИмяКласса
{
    Свойство1 = значение1,
    Свойство2 = значение2
};
```

3.1.3 Используя объект типа XmlSerializer, выполнить сериализацию информации о созданных массиве и списке и сохранить в отдельных файлах формата .xml сериализованные данные.

Результат проверить в текстовом редакторе или браузере.

3.2 Настройка атрибутов сериализации

Указать у свойства **возраст** атрибут сериализации, делающий его XML-атрибутом.

Изменить у свойства **имя** название элемента, используя атрибут сериализации.

Сравнить результаты, полученные с использованием атрибутов сериализации и без них.

3.3 Десериализация объектов класса с полями простых типов данных.

Используя объект типа XmlSerializer, считать данные из сохраненных файлов в отдельные переменные типа var. Результат десериализации проверить в режиме отладки.

Проверить, можно ли считать в переменную типа список переменную типа массив и наоборот.

3.4 Сериализация объектов класса с полями ссылочных типов данных

3.4.1 Создать класс для хранения информации о пользователе. Добавить в класс:

- открытые автореализуемые свойства идентификатор, логин и список комментариев (List<string>),
- открытое поле типа, созданного в п.3.1, для возможности указания имени и возраста пользователя.

3.4.2 Создать в основной программе массив или список объектов созданного класса.

3.4.3 Используя объект типа XmlSerializer, выполнить сериализацию информации о созданном массиве или списке.

Результат проверить в текстовом редакторе или браузере.

3.5 Десериализация объектов класса с полями ссылочных типов данных.

Выполнить десериализацию объектов из п.3.4. Результат десериализации проверить в режиме отладки.

4 Порядок выполнения работы

4.1 Выполнить все задания из п.3 в решении PractWork15. Каждый класс должен быть в отдельном файле. Возможные ошибки требуется обрабатывать. Выполнить форматирование и рефакторинг кода.

4.2 Ответить на контрольные вопросы.

5 Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

6 Контрольные вопросы

6.1 Что такое «XML» и для чего применяется этот формат?

6.2 Какое пространство имен нужно подключить для сериализации данных в формате XML?

6.3 Какой класс позволяет выполнить сериализацию и десериализацию в формате XML?

6.4 Какова общая форма вызова метода сериализации XML?

6.5 Какова общая форма вызова метода десериализации XML?

6.6 Какие атрибуты сериализации можно указать у полей и свойств класса?