

# Лабораторная работа 5.

## Классы для работы с файлами. Сериализация

Определить новые версии классов **Student** (вариант 1), **Magazine** (вариант 2) и **ResearchTeam** (вариант 3). В сформулированных ниже требованиях для этих классов использовано общее обозначение **T**.

В новые версии классов добавить **экземплярные** методы

- **T DeepCopy()** для создания полной копии объекта с использованием сериализации;
- **bool Save(string filename)** для сохранения данных объекта в файле с помощью сериализации;
- **bool Load(string filename)** для инициализации объекта данными из файла с помощью десериализации;
- **bool AddFromConsole()** для добавления в один из списков класса нового элемента, данные для которого вводятся с консоли;

и статические методы

- **static bool Save(string filename, T obj)** для сохранения объекта в файле с помощью сериализации;
- **static bool Load(string filename, T obj)** для восстановления объекта из файла с помощью десериализации.

В экземплярном методе **T DeepCopy()** вызывающий объект сериализуется в поток **MemoryStream**. Метод возвращает восстановленный при десериализации объект, который представляет собой полную копию исходного объекта.

Экземплярный метод **bool Save(string filename)** сериализует все данные вызывающего объекта в файл с именем **filename**. Если файл с именем **filename** существует, приложение его перезаписывает. Если такого файла нет, приложение его создает. Метод возвращает значение **true**, если сериализация завершилась успешно, и значение **false** в противном случае.

Экземплярный метод **bool Load(string filename)** десериализует данные из файла с именем **filename** и использует их для инициализации вызывающего объекта. Метод возвращает значение **true**, если инициализация завершилась успешно. Если полностью выполнить инициализацию объекта не удалось, исходные данные объекта должны остаться без изменения. В этом случае метод возвращает значение **false**.

Статические методы **bool Save(string filename, T obj)** и **bool Load(string filename, T obj)** получают через параметры имя файла и ссылку на объект, для

которого выполняется сериализация или восстановление. Методы возвращают значение true, если сериализация/инициализация завершилась успешно, и значение false в противном случае. Если полностью выполнить инициализацию объекта не удалось, исходные данные объекта должны остаться без изменения.

Во всех реализациях методов сохранения/восстановления данных из файла операции открытия файла, сериализации и десериализации данных должны находиться в блоках try-catch-finally.

В методе **bool AddFromConsole()** для добавления нового элемента в один из списков класса T

- пользователь получает приглашение ввести данные в виде одной строки символов с разделителями; приглашение содержит описание формата строки ввода, в том числе информацию о том, какие символы можно использовать в качестве разделителей;
- выполняется разбор данных; операции преобразования данных, которые могут бросить исключение, должны находиться в блоке try-catch;
- если разбор введенных данных был завершен успешно, в список добавляется новый элемент и метод возвращает значение true; в противном случае пользователь получает сообщение о том, что при вводе были допущены ошибки и возвращаемое значение метода равно false.

В **варианте 1** элементы, данные для которых вводятся с консоли, добавляются в список экзаменов `System.Collections.Generic.List<Exam>`. Вводятся название предмета, оценка и дата экзамена.

В **варианте 2** элементы добавляются в список статей в журнале `System.Collections.Generic.List<Article>`. Вводятся название статьи, данные автора статьи для объекта типа `Person` и рейтинг статьи.

В **варианте 3** элементы добавляются в список публикаций `System.Collections.Generic.List<Paper>`. Вводятся название публикации, данные автора статьи для объекта типа `Person` и дата публикации.

В методе **Main()**

1. Создать объект типа T с непустым списком элементов, для которого предусмотрен ввод данных с консоли. Создать полную копию объекта с помощью метода, использующего сериализацию, и вывести исходный объект и его копию.
2. Предложить пользователю ввести имя файла:

- если файла с введенным именем нет, приложение должно сообщить об этом и создать файл;
- если файл существует, вызвать метод `Load(string filename)` для инициализации объекта `T` данными из файла.

3. Вывести объект `T`.

4. Для этого же объекта `T` сначала вызвать метод `AddFromConsole()`, затем метод `Save(string filename)`. Вывести объект `T`.

5. Вызвать последовательно

- статический метод `Load( string filename, T obj)`, передав как параметры ссылку на тот же самый объект `T` и введенное ранее имя файла;
- метод `AddFromConsole()`;
- статический метод `Save (string filename, T obj)`.

6. Вывести объект `T`.

Приложение должно работать в режиме накопления. Если выбирается один и тот же файл для записи, и пользователь вводит данные без ошибок, при каждом следующем выполнении приложения к списку добавляются два новых элемента. Приложение должно обрабатывать все исключения, которые могут возникнуть из-за ошибок при вводе данных. Независимо от того, корректно были введены данные или при вводе были допущены ошибки, все файловые потоки должны быть закрыты.