Сериализация

№ урока: 8 **Курс:** C# Professional

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Урок посвящен сериализации объектов. Рассматриваются основные способы сериализации, их преимущества и недостатки, особенности и тонкости использования. Кроме того, приводится пример создания собственного способа сериализации в том случае, когда возможностей стандартных методов недостаточно.

Изучив материал данного занятия, учащийся сможет:

- Понимать назначение сериализации.
- Использовать XML сериализацию.
- Использовать Binary сериализацию.
- Использовать SOAP сериализацию.
- Создавать собственные методы сериализации.

Содержание урока

1. Общее понятие сериализации.
2. Процесс сериализации и десериализации. Совместимость версий.
3. Управление сериализацией.
4. Форматы сериализации.
5. XML сериализация.
6. Binary сериализация.
7. SOAP сериализация.
8. Собственные методы сериализации.

Резюме

- Создавая объекты в приложении .NET Framework, вы, возможно, никогда не задумывались о том, как данные хранятся в памяти. Но когда требуется сохранять содержимое объекта в файле, передавать объект другому процессу или по сети, возникает необходимость преобразования его в другой формат. Такое преобразование называется *сериализацией*.
- Сериализация – это процесс преобразования объекта в линейную последовательность байтов, которую можно хранить и передавать. Десериализация – это процесс обратного преобразования созданной последовательности байтов в объект.
- Собственные классы можно будет сериализовать и десериализовать, если добавить к ним атрибут `Serializable`. Этот атрибут можно добавлять к классам, даже если на данный момент сериализация не является необходимой, для того, чтобы добавить ее поддержку на будущее.
- Для отключения сериализации ненужных членов типа, например, полей, значение которых вычисляется в ходе работы программы, нужно использовать атрибут `NonSerialized`. Для автоматической инициализации таких полей после завершения десериализации рекомендуется использовать интерфейс `IDeserializationCallback`. Исполняющая среда будет вызывать метод `OnDeserialization` каждый раз после завершения десериализации класса.
- При попытке десериализовать объект, сереализованный предыдущей версией приложения, могут возникать проблемы с совместимостью. Например, если добавить член к собственному классу и попытаться десериализовать объект, в котором этого члена нет, исполняющая среда сгенерирует исключение.

  Обойти это ограничение можно двумя способами:

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

    - Реализовать собственный метод сериализации.
    - Применить атрибут `OptionalField` к новым членам, способным вызвать проблемы с совместимостью.
- `OptionalField` не влияет на процесс сериализации. Если член не был сериализован, во время десериализации исполняющая среда присвоит ему значение по умолчанию, и не будет генерировать исключение.
- `BinaryFormatter` – находится в пространстве имен `System.Runtime.Serialization.Formatters.Binary` и является наиболее эффективным способом сериализации объектов. Но, к сожалению, он подходит только для .NET приложений. Если десериализацию будет выполнять другая среда, от `BinaryFormatter` придется отказаться.
- `SoapFormatter` – расположен в пространстве имен `System.Runtime.Serialization.Formatters.Soap`, основан на XML, представляет наиболее надежный способ сериализации объектов, которые будут передаваться по сети или считываться приложениями, не использующими .NET Framework.
- `SoapFormatter` не поддерживает совместимость по сериализации между версиями .NET Framework. `BinaryFormatter` обеспечивает совместимость между версиями.
- XML – это стандартный формат текстовых документов, подходящий для хранения информации, к которой будут обращаться приложения. `System.Xml.Serialization` предоставляет методы преобразования объектов, в том числе и собственных классов, в XML-файлы и обратно.
- Преимущества XML сериализации:
    - Больше возможностей взаимодействия.
    - Удобнее работать администраторам.
    - Улучшенная прямая совместимость.
- Ограничения XML сериализации:
    - Можно сериализовать только открытые данные, закрытые данные таким образом не сериализуются.
    - Нельзя сериализовать графы объектов, сериализация применима только к отдельным объектам.
- Чтобы сериализовать класс в формат XML, нужно выполнить следующее:
    - Объявить класс как открытый.
    - Объявить все члены, которые нужно сериализовать, как открытые (`public`).
    - Создать конструктор, не принимающий параметров.
- Для применения XML сериализации класс не обязательно должен иметь атрибут `Serializable`.
- При XML сериализации имена XML-элементов основаны на именах классов и членов, каждый член сериализуется как отдельный элемент XML. Однако, существует набор атрибутов для сериализации в XML, которые обеспечивают соответствие сериализованного класса определенным требованиям XML.
- Собственные методы сериализации позволяют контролировать сериализацию и десериализацию типов. Управляя сериализацией, можно обеспечить совместимость по сериализации, т.е. возможность сериализовать и десериализовать объекты разных версий типа, не изменяя его базовую функциональность.

Закрепление материала

- Что такое сериализация?
- Каким образом класс отмечается как сериализуемый?
- Каково назначение атрибута `NonSerialized`?
- Как обеспечить вызов метода `OnDeserialization` после завершения десериализации?
- Какие проблемы могут возникать при десериализации? Как с ними справиться?
- Расскажите о назначении и применении Binary сериализации.
- Расскажите о назначении и применении SOAP сериализации.
- Расскажите о назначении и применении XML сериализации.

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

Дополнительное задание

Создайте пользовательский тип (например, класс) и выполните сериализацию объекта этого типа, учитывая тот факт, что состояние объекта необходимо будет передать по сети.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные конструкции и понятия, рассмотренные на уроке.

Задание 2
Создайте класс, поддерживающий сериализацию. Выполните сериализацию объекта этого класса в формате XML. Сначала используйте формат по умолчанию, а затем измените его таким образом, чтобы значения полей сохранились в виде атрибутов элементов XML.

Задание 3
Создайте новое приложение, в котором выполните десериализацию объекта из предыдущего примера. Отобразите состояние объекта на экране.

Задание 4
Зайдите на сайт MSDN.
Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: класс `SoapFromatter`
http://msdn.microsoft.com/ru-ru/library/system.runtime.serialization.formatters.soap.soapformatter.aspx

MSDN: класс `BinaryFromatter`
http://msdn.microsoft.com/ru-ru/library/system.runtime.serialization.formatters.binary.binaryformatter.aspx

MSDN: класс `XMLSerializer`
http://msdn.microsoft.com/ru-ru/library/system.xml.serialization.xmlserializer.aspx

MSDN: сериализация
http://msdn.microsoft.com/ru-ru/library/ms233843.aspx

CyberBionic systematics

CyberBionic Systematics ®
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com