Программа курса
«Технология доступа к базам данных ADO.NET»

Технологія доступу до баз даних ADO.NET
ADO.Net: Database Access Technology

Для групп стационара. Версия 3.0.3
Объём курса: 32 пары.

Цель курса
Обучить слушателя разработке Windows-приложений с использованием платформы Microsoft.NET, языка программирования C# и технологии доступа к данным ADO.NET. Исследовать механизмы доступа к данным для использования в рамках Windows и Web приложений.

По окончании курса слушатель будет:
- уметь создавать Windows приложения с доступом к источникам данных;
- разбираться в технологиях доступа к данным;
- уметь выбирать правильный механизм доступа к источнику данных;
- уметь соединяться с базой данных, добавлять, удалять, обновлять данные;
- вызывать хранимые процедуры и передавать параметры;
- использовать механизм транзакций;
- уметь работать в присоединенном и отсоединенном режиме;
- уметь использовать механизмы LINQ.

По окончании данного курса студент сдаёт все практические задания курса. На основании всех сданных заданий выставляется оценка по предмету.

Перед началом данного предмета необходимо предоставить студентам доступ к следующим курсам Microsoft Imagine Academy:
- Using Data in Software Applications.

Тематический план

Модуль 1

Введение в ADO.NET

1. Что такое ADO.NET?
2. Исторический экскурс в технологии доступа к данным:
   - ODBC;
   - DAO;
   - OLEDB;
   - ADO.
3. Сравнительный анализ технологий доступа к данным.
4. Сравнительный анализ понятий драйвер и провайдер.
5. Пространства ADO.NET:
   - System.Data;
   - System.Data.Common;
   - System.Data.OleDb;
   - System.Data.SqlClient;
   - System.Data.Sql;
   - System.Data.Odbc;
   - System.Data.OracleClient;
   - Другие пространства.
6. Модели работы ADO.NET:
   - присоединенный режим;
   - отсоединенный режим.
7. Концепция интерфейсов и базовых классов ADO.NET:
   - интерфейс IDbConnection;
   - интерфейс IDbCommand;
   - интерфейсы IDataReader, IDataRecord;
   - интерфейсы IDataAdapter, IDbDataAdapter;
   - интерфейсы IDataParameter, IDbDataParameter;
   - интерфейс IDbTransaction;
   - классы DbConnection, DbCommand, DbDataReader, DbDataAdapter, DbParameter, DbTransaction.
8. Обзорный пример использования ADO.NET для доступа к источнику данных.

Модуль 2

Присоединенный режим

1. Класс DbConnection:
   - цели и задачи класса DbConnection и его потомков;
   - анализ методов и свойств;
   - пример соединения с источником данных.
2. Класс DbCommand:
   - цели и задачи класса DbCommand и его потомков;
   - анализ методов и свойств;
   - пример посылки запроса.
3. Класс DbDataReader:
   - цели и задачи класса DbDataReader и его потомков;
   - анализ методов и свойств;
   - пример получения данных.
4. Примеры вставки, обновления, удаления данных.
5. Использование параметров:
   - класс DbParameter:
     - цели и задачи класса DbParameter и его потомков;
     - анализ методов и свойств;
     - пример передачи параметров.
   - использование и вызов хранимых процедур.
6. Использование транзакций:
   - класс DbTransaction:
     - цели и задачи класса DbTransaction и его потомков;
     - анализ методов и свойств;
     - пример работы с транзакциями.
   - использование механизма транзакций аппарата СУБД.

Модуль 3

Фабрика провайдеров, асинхронный режим доступа, конфигурационные файлы

1. Фабрика провайдеров ADO.NET:
   - концепция фабрики провайдеров;
   - класс DbProviderFactories:
     - цели и задачи класса DbProviderFactories и его потомков;
     - анализ методов класса;
     - пример использования получения конкретной фабрики.

- фабрики, специфичные для провайдера:
  - цели и задачи класса DbProviderFactory и его потомков;
  - анализ методов и свойств;
  - пример использования конкретной фабрики.

2. Асинхронные механизмы доступа к данным:
   - что такое асинхронность?
   - зачем нужна асинхронность?
   - анализ методов для асинхронного механизма доступа к данным;
   - примеры использования асинхронного доступа.
3. Использование конфигурационных файлов:
   - что такое конфигурационный файл?
   - цели и задачи конфигурационных файлов;
   - типы конфигурационных файлов;
   - способы доступа к конфигурационным файлам;
   - примеры использования конфигурационных файлов при доступе к источникам данных.

Модуль 4

Отсоединенный режим

1. Что такое отсоединенный режим?
2. Концепция использования отсоединенного режима.
3. Что такое DataSet?
   - цели и задачи класса DataSet;
   - анализ методов и свойств;
   - пример использования DataSet.
4. Класс DataTable:
   - цели и задачи класса DataTable;
   - анализ методов и свойств;
   - пример использования DataTable.
5. Класс DataRow:
   - цели и задачи класса DataRow;
   - анализ методов и свойств;
   - пример использования DataRow.
6. Класс DataColumn:
   - цели и задачи класса DataColumn;
   - анализ методов и свойств;
   - пример использования DataColumn.
7. Класс DbDataAdapter:
   - цели и задачи класса DbDataAdapter;

- анализ методов и свойств;
- пример использования DbDataAdapter.

8. Класс SqlCommandBuilder:
   - Цели и задачи класса SqlCommandBuilder;
   - Анализ методов и свойств;
   - Пример использования SqlCommandBuilder.

Модуль 5

Введение в LINQ

1. Что такое LINQ?
2. Цели и задачи LINQ.
3. Понятие запроса:
   - запрос в LINQ;
   - синтаксис запроса;
   - исполнение запроса;
   - сортировка;
   - ключевые слова let и into;
   - группировка;
   - вложенные запросы;
   - объединения (join).
4. Использование LINQ и коллекций.
5. Использование LINQ и XML.
6. Использование LINQ и SQL.
7. Примеры использования.

Модуль 6

Введение в Entity Framework

1. Что такое Entity Framework?
2. Цели и задачи Entity Framework.
3. Понятие модели.
4. Понятие сгенерированного кода.
5. Обзор различных подходов при работе с EF:
   - DB First;
   - Model First;
   - Code First.
6. Использование DB First:
   - почему стоит использовать DB First?
   - создание модели;
   - изменение модели.

7. Использование Model First:
   - почему стоит использовать Model First?
   - создание модели;
   - изменение модели.
8. Использование Code First:
   - почему стоит использовать Code First?
   - создание модели;
   - понятие domain classes;
   - изменение модели.
9. Примеры использования.