

Praktikum 9

NextChange - ADO.NET

In diesem Praktikum machen Sie Ihre ersten Gehversuche mit ADO.NET und integrieren Kalenderdaten in ein Windows-GUI. Der Einfachheit halber arbeiten Sie mit einer lokalen Access Datenbank.

Verwendete Datenstruktur

Zunächst die Struktur der mitgelieferten Daten:

Appointments : Table		
	Field Name	Data Type
	ID	Text
	Subject	Text
	Start	Date/Time
	End	Date/Time
	AllDayEvent	Yes/No
	Location	Text
	Private	Yes/No
	Organizer	Text
	RequiredAttendees	Text

- Eine Zeile enthält genau ein Appointment.
- Die Zeiten Start und End sind vom Typ DateTime.

Aufgabe 1 - ADO.NET „Hallo, Welt!“

Schreiben Sie eine kleine Kommandozeilenapplikation, um die Daten des mitgelieferten mdb-Files zu lesen und auf die Konsole ausgeben.

Hinweis:

- Die Verbindung sollte über ein Objekt vom Typ `IDbConnection` erfolgen, nicht über den Typ der Implementierung (z.B. `OleDbConnection`). So können Sie später einfacher die Access-Datenbank durch eine andere Datenbank ersetzen.

Konkret:

```
using System.Data.OleDb;
...
IDbConnection conn = new OleDbConnection("provider=Microsoft.Jet.OLEDB.4.0;
data source=m:\private\app97.mdb;");
// oder im fall eines accdb Files
IDbConnection conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=C:\myFolder\app97.accdb;")
conn.Open();
```

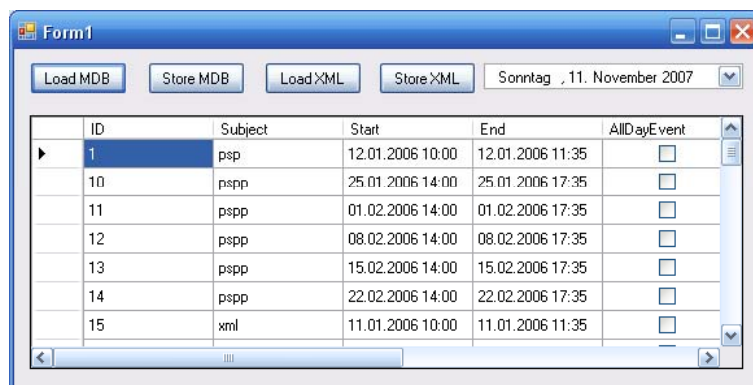
Verwenden Sie eine `IDbCommand` und einen `IDbReader` um Abfragen durchzuführen und das Resultat auszugeben, z.B. die Termine der dnet Vorlesungen.

Aufgabe 2 - Verbindungsloser Zugriff auf eine DB.

In der zweiten Aufgabe soll nicht direkt mit `IDbReader` etc. gearbeitet werden, sondern mit der komfortableren `DataSet`-Klasse.

Erstellen Sie eine `DataSet` und eine `DataGridView` in der Design Ansicht.

Implementieren Sie die beiden Methoden `LoadTable()` und `StoreTable()`, um ein `DataSet` mit den gewünschten Daten zu füllen. Die Buttons <Load MDB> und <Store MDB> aktivieren die beiden Methoden `LoadTable()` und `StoreTable()`.



Mittels

```
DataView dv = dataSet1.Tables["Appointments"].DefaultView;  
dataGridView1.DataSource = dv;
```

verbinden Sie beim Laden die `DataGridView` mit dem `DataSet`.

Erstellen Sie weiter eine Suchmaske, mit der Sie Suchkriterien für die `DataGridView` eingeben können, zum Beispiel das Datum. Übergeben Sie diese Suchkriterien als `RowFilter` dem `DataGridView` mit den entsprechenden Properties. Den Filter-String für ein Datum können Sie folgendem Beispiel entnehmen:

```
string dateDB = dt.Value.ToString("MM.dd.yyyy");  
string filter = "Start > #" + dateDB + " 00:00:00# and Start < #" + dateDB +  
" 23:59:59#";
```

Aufgabe 3 - Verbindungsloser Zugriff auf eine XML-Datei

Die Daten seien nun nicht in einer Datenbank, sondern in der mitgelieferten XML-Datei gespeichert. Es sollen ein verbindungsloser Zugriff auf eine XML-Datei mit Hilfe eines `DataSet`s und die Funktionen *Load XML* und *Store XML* realisiert werden.