

Arbeitsblatt: DNET1

Name:

Kurznamen:

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 Access Datenbank.

Verwendete Datenstruktur

Die Struktur der mitgelieferten Daten:

Field Name	Data Type	Description
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 accdb-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:

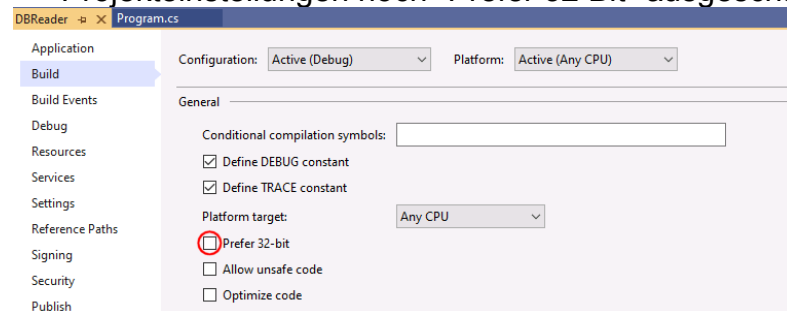
```
using System.Data.OleDb;

IDbConnection conn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=C:\myFolder\app2020.accdb;")
conn.Open();
```

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

Hinweise

- Die Architektur (32/64 Bit) des Datenbanktreibers muss mit derjenigen des installierten Offices übereinstimmen.
- Falls die 64 Bit Version installiert wird, dann muss bei den Projekteinstellungen noch "Prefer 32 Bit" ausgeschaltet sein.



Abgabe:

Praktikum: DT9.1

Datei: ReadAccess.cs

Aufgabe 2 - Zugriff auf SQLite Datenbank

Unter folgender Urls finden Sie Informationen, wie Sie eine SQLite Datenbank zugreifen können;

https://docs.microsoft.com/en-us/dotnet/standard/data/sqlite/#tabpanel_CeZOj-G++Q_visual-studio

Lösen Sie die Aufgabe eins nochmals mit einer SQLite Datenbank; die Struktur sieht wie folgt aus.

Name	Type	Schema
Tables (1)		
Appointments		CREATE TABLE `Appointmen
ID	TEXT	"ID" TEXT
Subject	TEXT	"Subject" TEXT
Start	TEXT	"Start" TEXT
End	TEXT	"End" TEXT
AllDayEvent	TEXT	"AllDayEvent" TEXT
Location	TEXT	"Location" TEXT
Private	TEXT	"Private" TEXT
Organizer	TEXT	"Organizer" TEXT
RequiredAttendees	TEXT	"RequiredAttendees" TEXT

Das notwendige Nuget Paket finden Sie hier

<https://www.nuget.org/packages/Microsoft.Data.Sqlite>



Microsoft.Data.Sqlite is a lightweight ADO.NET provider for SQLite.

Commonly Used Types:

Microsoft.Data.Sqlite.SqliteCommand
Microsoft.Data.Sqlite.SqliteConnection
Microsoft.Data.Sqlite.SqliteConnectionStringBuilder
Microsoft.Data.Sqlite.SqliteDataReader
Microsoft.Data.Sqlite.SqliteException
Microsoft.Data.Sqlite.SqliteFactory
Microsoft.Data.Sqlite.SqliteParameter
Microsoft.Data.Sqlite.SqliteTransaction

Abgabe:

Praktikum: DT9.2

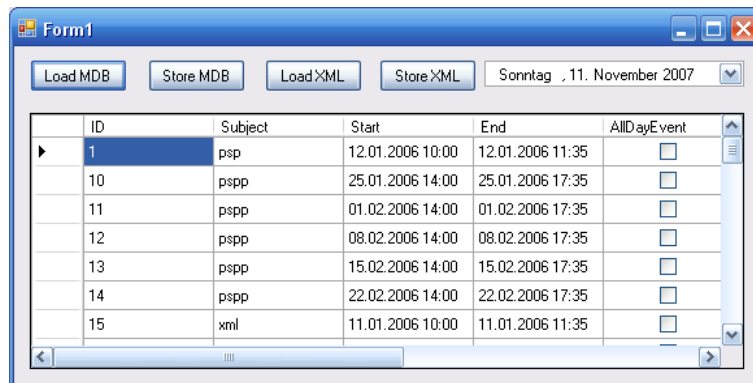
Datei: ReadSQLite.cs

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. Nehmen Sie die Vorlagen aus dem Anhang der Vorlesung als Basis. 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.

Hinweis:

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#";
```

Abgabe:

Praktikum: DT9.3

Datei: Forms1.cs

Aufgabe 4 - Verbindungsloser Zugriff auf eine XML-Datei (optional)

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.