

## Arbeitsblatt: DNET1

Name:

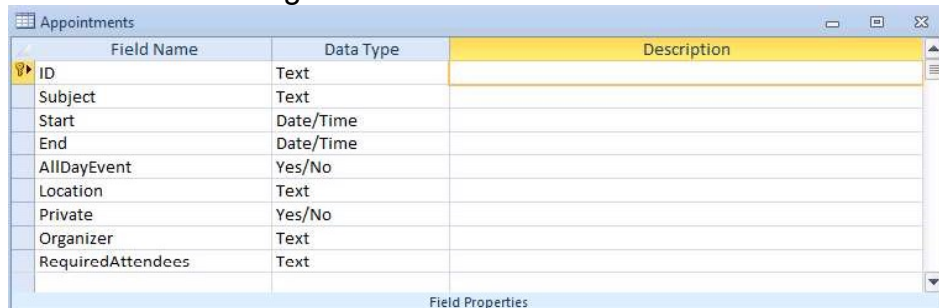
Kurznamen:

### 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!“

#### a) Zugriff Access DB File (optional)

Schreiben Sie eine Kommandozeilenapplikation (**Framework**), um die Daten des mitgelieferten accdb-Files zu lesen und auf die Konsole auszugeben.

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:

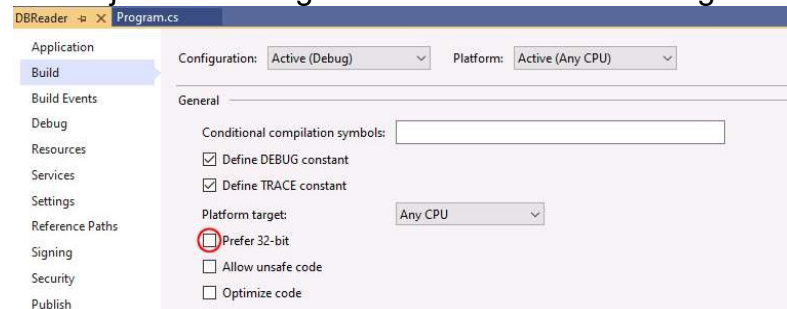
```
using System.Data.OleDb;

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. Laden Sie diesen unter folgendem Link runter:  
<https://www.microsoft.com/en-us/download/details.aspx?id=54920>
- Falls die 64 Bit Version installiert wird, dann muss bei den Projekteinstellungen noch "Prefer 32 Bit" ausgeschaltet werden.



- Falls .NET Core oder Linux verwendet werden soll gehen Sie nach folgender Beschreibung vor.  
<https://medium.com/@daniel.sagita/using-microsoft-access-in-net-core-b419dbeeab1>

## b) Zugriff auf SQLite Datenbank

Unter folgendem Link 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](https://docs.microsoft.com/en-us/dotnet/standard/data/sqlite/#tabpanel_CeZOj-G++Q_visual-studio)

Der Verbindungsaufbau:

```
connection = new SQLiteConnection("Data Source=app2020.db")
```

Lösen Sie die Aufgabe mit einer SQLite Datenbank; die Struktur sieht analog 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** by Microsoft

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

6.0.0

7.0.0



**Microsoft.Data.Sqlite.Core** by Microsoft

Microsoft.Data.Sqlite is a lightweight ADO.NET provider for SQLite. This package does not include a copy of the native SQLite library.

7.0.0



**SQLitePCLRaw.bundle\_winsqlite3** by Eric Sink

This 'batteries-included' bundle brings in SQLitePCLRaw.core and the necessary stuff for certain common use cases. Call SQLitePCL.Batteries.Init(). Policy of this bundle: no SQLite library included, uses winsqlite3.dll from Windows 10.

2.1.3

## Abgabe:

Praktikum: DN9.1

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. Die Buttons `<Load MDB>` und `<Store MDB>` aktivieren die beiden Methoden `LoadTable()` und `StoreTable()`.

## Mittels

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

verbinden Sie beim Laden die `DataGridView` mit dem `DataSet`. In der View können Sie z.B. nach einem Datum filtern.

## Hinweise:

- Nehmen Sie die Vorlagen aus dem Anhang der Folien als Basis.
- 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#";
```

**Hinweise:**

- Verwenden Sie den Zugriff über ODBC  
<https://www.nuget.org/packages/System.Data.Odbc/>
- Das DataSet einfach mittels new erstellen
- Driver für SQLite finden Sie unter <http://www.ch-werner.de/sqliteodbc/>
- Konfigurieren Sie die DB über `odbcad32` Tool (über run)
- oder Verbindungsstring für "DNS Less" Zugriffspfad  
<https://stackoverflow.com/questions/639899/connect-to-sqlite-using-odbc-without-register-database/769274>
- Connection String: `Driver=SQLite3 ODBC Driver; Database=c:\tmp\app2020.db;`

**Abgabe:**

Praktikum: DN9.2

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.