

Учебен план

1. Въведение в приложното програмиране
2. Въведение в платформата .NET
3. Среда за разработка Visual Studio .NET
4. Език за програмиране C#
5. Изграждане на Windows приложения
6. Използване на бази от данни

Приложно програмиране

Използване на бази от данни

Технология

- ◆ ADO.NET – Active Data Objects
- ◆ включва различни драйвери и набори от класове
- ◆ класовете са в namespace `System.Data.*`

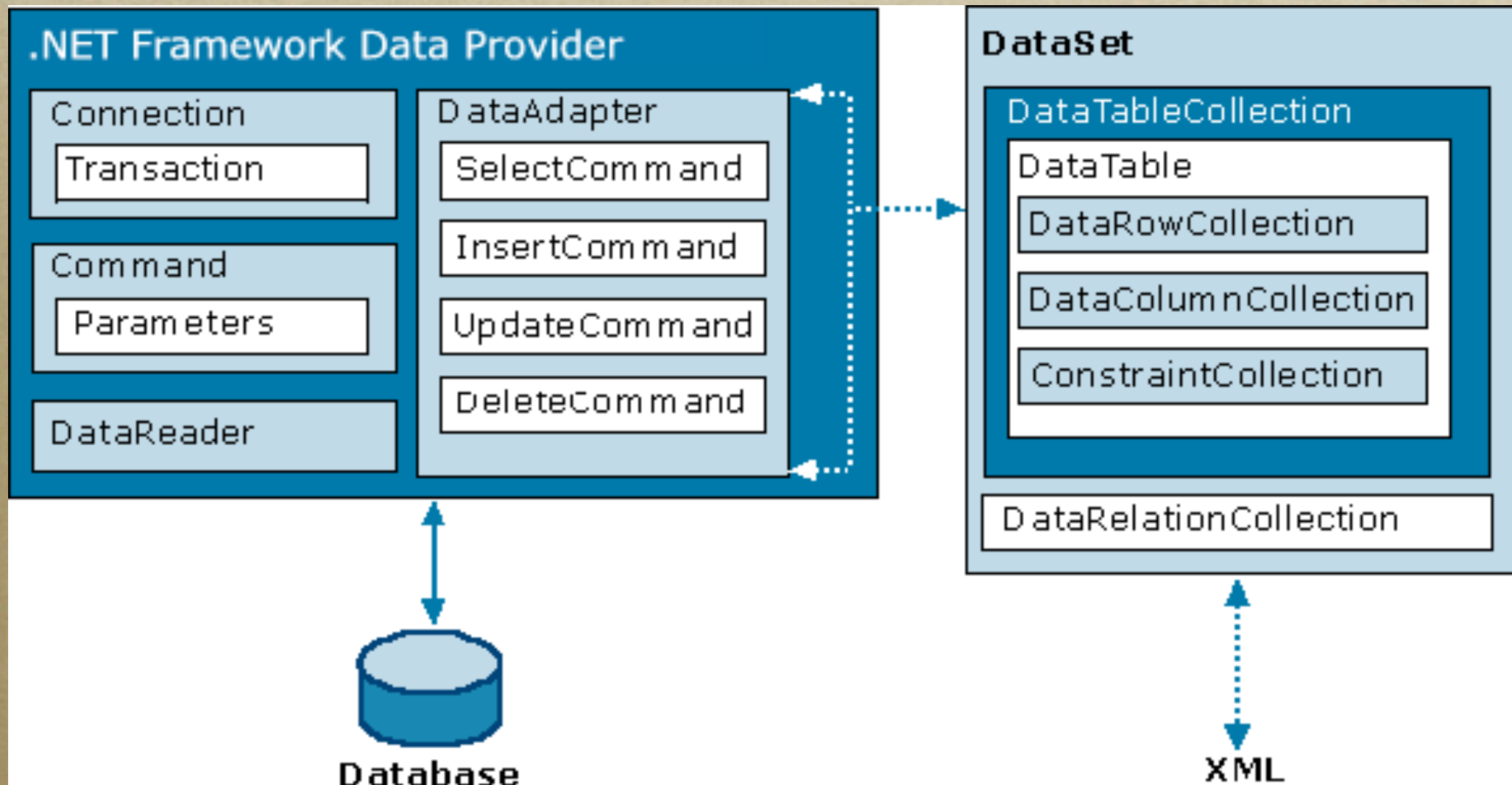
Начини на работа

- ◆ визуално–ориентиран
- ◆ "ръчен", с код
- ◆ комбиниран

Основни понятия (класове)

- ◆ Data Source
- ◆ Data Provider
- ◆ Connection
- ◆ Command
- ◆ Data Reader
- ◆ Data Adapter
- ◆ Data Set

Класове/Компоненти



Data Source

- ◆ ИЗТОЧНИКЪТ на данни
- ◆ конкретна БД, поддържана чрез някой от "адаптерите"
- ◆ може да бъде файл или сървър
- ◆ може да бъде локален или по мрежата

Data Provider

- ◆ осигурява достъп до конкретен вид данни
- ◆ може да се разглежда като аналог на драйвер

Видове data provider(s)

- ◆ SQL Server
- ◆ Oracle (поддръжката може да бъде премахната)
- ◆ ODBC
- ◆ OLE DB (поддръжката може да бъде премахната)

SQL Server

- ◆ оптимизиран достъп до SQL Server, да се избира именно този вариант за тази СУБД
- ◆ поддържа се версия 7.0 и по-нови
- ◆ namespace System.Data.SqlClient

Oracle

- ◆ оптимизиран достъп до Oracle, да се избира именно този вариант за тази СУБД
- ◆ поддържа се версия 8.1.7 и по-нови
- ◆ namespace
System.Data.OracleClient

ODBC

- ◆ достъп до произволна база от данни, за която има ODBC драйвер
- ◆ достъпът не е толкова добре оптимизиран; да не се използва за SQL Server или Oracle
- ◆ namespace System.Data.Odbc

OLE DB

- ◆ достъп до произволна база от данни, за която има OLE DB Provider
- ◆ достъпът не е добре оптимизиран, стара технология, за съвместимост
- ◆ за Access, SQL Server версия 6.5 или по-стари
- ◆ namespace System.Data.OleDb

Connection

- ◆ дефинира основните параметри на връзката с конкретен източник
- ◆ име/адрес/път на БД
- ◆ начин на достъп
- ◆ име и парола ако е необходимо

Command

- ◆ определена команда/заявка към БД
- ◆ може да бъде напр. SQL код/ заявка

Data Reader

- ◆ най-простият вид "четец" на данни
- ◆ осъществява последователен еднопосочен (само напред) достъп до записите

Data Adapter

- ◆ осъществява достъп до част от данните на източника (напр. една таблица)
- ◆ осъществява връзката (двупосочна) с Data Set

Data Set

- ◆ представя пред програмиста и/или контролите набор от данни, получени от БД посредством Data Adapter
- ◆ ако данните се актуализират в програмата, реалното актуализиране в БД се извършва отново посредством Data Adapter
- ◆ може да се ползва и с други източници, напр. XML файл

"Ръчен" програмен достъп

- ◆ примери за SQL Server, namespace `System.Data.SqlClient`
- ◆ клас `SqlConnection` – създава връзка със сървъра
- ◆ клас `SqlCommand` – формира SQL заявка
- ◆ клас `SqlDataReader` – получава резултат от `SELECT` заявка

Клас SqlConnection

- ◆ конструктор `SqlConnection(string ConnectionString)`
- ◆ `ConnectionString` е и свойство, което може да се укаже/промени допълнително

ConnectionString – някои опции

- ◆ `Server="server";` (свойство `DataSource`)
- ◆ `Database="database";` (свойство `Database`)
- ◆ `Integrated Security="true";/"false";`
– Windows имена/пароли или се дават директно в стринга
- ◆ `User Id = "user";`
- ◆ `Password = "password";`

SqlConnection – някои методи

- ◆ Open – за отваряне, трябва да е зададен `ConnectionString`
- ◆ Close – затваряне
- ◆ `BeginTransaction` – започва/създава транзакция

Отваряне на SqlConnection

```
SqlConnection mySqlCon = new  
    SqlConnection("server=localhost; database=stud;  
    integrated security=true");  
  
mySqlCon.Open();  
  
...  
  
mySqlCon.Close();
```


Транзакции

- ◆ операциите, извършени по време на транзакцията или се прилагат или се отменят като цяло
- ◆ клас `SqlTransaction`, връща се от `SqlConnection.BeginTransaction`
- ◆ `Commit` – прилага операциите (транзакцията)
- ◆ `Rollback` – отменя операциите (транзакциите)

Транзакция – пример

```
SqlConnection mySqlConnection = new  
    SqlConnection("server=localhost; database=stud;  
    integrated security=true");
```

```
mySqlConnection.Open();
```

```
SqlTransaction myTrans;  
myTrans = mySqlConnection.BeginTransaction();
```

```
...
```

```
myTrans.Commit(); или myTrans.Rollback();
```

```
mySqlConnection.Close();
```


Клас SqlDataAdapter

- ◆ свързващото звено между SqlConnection и DataSet (който може да се използва ръчно или да се избира за източник на данни за контроли)
- ◆ може да се извършват двупосочни операции – получаване на данни от него и изпращане на данни към него

SqlDataAdapter – някои свойства/методи

- ◆ конструктор `SqlDataAdapter(string SelectCommand, SqlConnection connection)`
- ◆ `Fill(DataSet ds)`
- ◆ `Update(DataSet ds)`

Клас DataSet

- ◆ представяне в паметта на данни от база от данни или файл
- ◆ може да се използва програмно или като източник на данни за контроли по формите
- ◆ има колекция Tables от тип DataTable

Клас DataTable

- ◆ представя таблица от база от данни или файл
- ◆ има колекция Columns
- ◆ има колекция Rows от тип DataRow
- ◆ метод NewRow