

T.C.
MİLLİ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

**GÖRSEL PROGRAMLAMADA VERİ
TABANI**

ANKARA 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iv
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. METİN VE İKİLİK DOSYALAR	3
1.1. Yapı (Structure)	3
1.2. Alan(Field) Kavramı	4
1.3. “SYSTEM.IO” Ad Alanı (name space)	4
1.4. Dosya (File), Klasör (Directory) ve Yol (Path) Deyimleri	4
1.4.1. Dosya(File)	4
1.4.2. Klasör (Directory).....	5
1.4.3. Yol (Path)	5
1.5. “Create Directory, Copy, Move, GetAttributes, SetAtributes, Delete ve Exists” Metotları.....	5
1.5.1. Create Directory	5
1.5.2. Copy.....	5
1.5.3. Move.....	6
1.5.4. GetAttributes	6
1.5.5. SetAttributes	7
1.5.6. Delete.....	7
1.5.7. Exists	7
1.6. Metin Dosyalar ve İkili Dosyalar (Text ve Binary)	8
1.6.1. Metin (Text) Dosyalar	8
1.7. Satır Sonu Bitirme Karakteri.....	8
1.8. Input ve Output Stream Deyimleri.....	8
1.9. FileStream Sınıfı	8
1.10. FileMode	9
1.11. FileAccess	9
1.12. FileShare	10
1.12. EOF, LOF, FileCopy, FileClose, FileLen, FileGet, FileOpen, Kill, Print, Seek ve Write Komutları	11
1.12.1. EOF	11
1.12.2. LOF.....	11
1.12.3. FileCopy	11
1.12.4. FileLen.....	11
1.12.5. FileGet	12
1.12.6. FileOpen	12
1.12.7. Kill.....	13
1.12.8. Print	13
1.12.9. Seek	13
1.12.10. Write	13
UYGULAMA FAALİYETİ	15
ÖLÇME VE DEĞERLENDİRME	16
ÖĞRENME FAALİYETİ-2	17
2.2. HTML, Web Sayfası, UTF-8	17
2.2.1. HTML	17

2.2.2. Web Sayfası.....	18
2.2.3. UTF-8	18
2.3. “Tag, Attribute, Comment ve Element”.....	18
2.3.1. Tag (etiket)	18
2.3.2. Attribute.....	18
2.3.3. Comment	19
2.3.4. Element.....	19
2.4. System. XML Sınıfı	19
2.5. Kök, Ana ve Çocuk Elemanlar	19
2.6. “XML Designer” penceresi.....	20
2.7. “XmlTextWriter, XmlTextReader” Sınıfı ve “Formatting” Özelliği.....	21
2.7.1. XmlTextWriter	21
2.7.2. XmlTextReader	21
2.7.3. Formatting	21
2.7.4. Indentation.....	21
2.7.5. IndentChar	21
2.8. WriteStart Document, Write Comment, WriteAttributeString, WriteEndElement Komutları	21
2.8.1. WriteStartDocument	21
2.8.2. WriteComment	21
UYGULAMA FAALİYETİ	25
ÖLÇME VE DEĞERLENDİRME	26
ÖĞRENME FAALİYETİ-3	27
3. ADO.NET(ACTIVEX DATA OBJECTS) VE OLEDB.....	27
3.1. Data (Veri), Grid (ızgara) Nesneleri	27
3.1.1. Data (Veri).....	27
3.1.2. Grid (Izgara)	27
3.2. ADO.NET ve SQL (Structured Query Language).....	27
3.2.1. ADO.NET.....	27
3.2.2. SQL (Structured Query Language).....	28
3.3. “Select, Insert, Update, Delete” SQL Komutları	28
3.3.1. Select	28
3.3.2. Insert	29
3.3.3. Update.....	30
3.3.4. Delete.....	30
3.4. İlişkisel Veri Tabanı, Tablo, İndeks.....	30
3.5. Microsoft SQL Server 2005 Desktop Engine (MSDE), DBMS	31
3.6. LAN, WAN, İnternet, İntranet Deyimleri.....	31
3.6.1. LAN (Local Area Network).....	31
3.6.2. WAN (Wide Area Network).....	31
3.6.3. İnternet.....	31
3.6.4. İntranet.....	31
3.7. “Dataset, Data Adapter, Data Connection, Data Command” Nesneleri	32
3.7.1. Dataset	32
3.7.2. DataAdapter.....	32
3.7.3. DataConnection	32
3.7.4. DataCommand.....	32

UYGULAMA FAALİYETİ	33
ÖLÇME VE DEĞERLENDİRME	34
ÖĞRENME FAALİYETİ-4	35
4. VERİ BAĞLI NESNELER.....	35
4.1. Data Adapter Configuration Wizard	35
4.2. “Connection” Kavramı.....	36
4.3. Query Builder Penceresi	38
4.4. Dataset, Datagrid Nesneleri	41
4.5. “Command Text” Komutu	42
4.6. ”SelectCommand, Update Command, DeleteCommand, InsertCommand” Komutları	42
4.6.1. SelectCommand.....	42
4.6.2. UpdateCommand.....	43
4.6.3. DeleteCommand	43
4.6.4. InsertCommand	43
4.7. @ Karakteri ile Başlayan Parametreler	43
4.8. Nesnelerin DataBindings Özelliği	43
4.9. “BindingManagerBase” Nesnesi, “Count ve Position” Özellikleri	44
4.9.1. BindingManagerBase	44
4.9.2.Count	44
4.9.3. Position	44
UYGULAMA FAALİYETİ	46
ÖLÇME VE DEĞERLENDİRME	47
MODÜL DEĞERLENDİRME	48
CEVAP ANAHTARLARI	50
ÖNERİLEN KAYNAKLAR.....	51
KAYNAKÇA	52

AÇIKLAMALAR

KOD	482BK0072
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Görsel Programlamada Veri Tabanı
MODÜLÜN TANIMI	Metin dosyaları ve veri tabanı ile ilgili temel becerilerin kazandırıldığı öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Görsel Programlamada Hata Giderme modülünü bitirmiş olmak
YETERLİK	Görsel programlamada veri tabanı ile program yazmak
MODÜLÜN AMACI	Genel Amaç Gerekli ortam sağlandığında, metin, XML ve ADO.NET kullanarak programınıza veri tabanı ekleyebileceksiniz. Amaçlar Bu modül ile gerekli ortam sağlandığında 1. Metin ve ikilik dosyalar ile çalışabileceksiniz. 2. XML dosyalar ile çalışabileceksiniz. 3. ADO.NET (ActiveX Data Objects) ve OLEDB ile çalışabileceksiniz. 4. Veri bağlı nesnelere ve parametrelili sorgular hazırlayabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Görsel programlama için gerekli donanıma ve yazılıma sahip bilgisayar, kâğıt ve kalem hazır bulundurulmalıdır.
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Öğretmen modül sonunda size ölçme aracı (uygulama, soru-cevap) uygulayarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek değerlendirecektir.

GİRİŞ

Sevgili Öğrenciler,

Bilişim teknolojilerindeki uygulamaların tamamına yakını kullanıcı ara yüzüne sahip uygulamalardır. Kullanıcı ara yüzüne sahip uygulamaların kullanımı diğer uygulamalara göre çok daha kolaydır.

Genellikle uygulamadan istenen, bir yerden verileri alıp, onları işledikten sonra ekran ve yazıcıda listeleyip, işlenen bilgileri saklamasıdır. Bu işlemleri yapabilmek için uygulamamızın dosyalara erişip buradan verileri okuyabilmesi, verileri yazabilmesi veya veri tabanına erişip kayıtları listeleyebilmesi, güncelleyebilmesi, ekleyebilmesi ve silebilmesi gerekir. Bu işlemleri yapamayan uygulamalar, her yeniden çalıştığında verileri klavyeden alacak (1000 kişilik bir işletmede çalışanların bilgisi veya bir okuldaki öğrenci ve öğretmenlerin bilgisi), işlemleri yaptıktan sonra bilgileri sadece ekran ve yazıcıdan verecek, uygulama kapatıldığında ne veriler ne de bilgiler kalmayacaktır.

Sonuç olarak modülde anlatılan dosyalar, veri tabanları, XML konuları; kullanıcı ara yüzüne sahip bir uygulamanın vazgeçilmez öğeleridir.

ÖĞRENME FAALİYETİ-1

AMAÇ

Gerekli ortam sağlandığında metin ve ikilik dosyalarda; dosya oluşturarak dosyayı okumak için açacak, içindeki bilgileri listeleyecek, döngü içinde bilgileri yapı içine kaydedecek, yapıdaki bilgileri form üzerindeki uygun nesnelere aktarabileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken araştırmalar şunlardır:

- Dosyaların programda niçin kullanıldığını araştırınız.
- Dosya özelliklerini araştırınız.

Araştırma işlemleri için internet ortamını kullanınız.

1. METİN VE İKİLİK DOSYALAR

1.1. Yapı (Structure)

Structure, birbiri ile ilişkili farklı veri tiplerindeki değişkenlerin tek bir yapı altında tutulması için kullanılır. Structure programcı tarafından tanımlanan bir veri türüdür. Bir structure'ı oluşturan değerler genelde birbiriyle ilişkilidir. Bir structure en az bir alan içermelidir.

Yazacağımız programda bazı bilgilerin dosyaya yazılması gerekecektir. Okul programında öğrenci numarası, adı, soyadı, doğum yeri, doğum tarihi gibi bilgileri saklamamız gerekir. Bu bilgileri dosyaya yazarken, dosyadan okurken, ararken tüm alanlar ayrı, ayrı komutlarla kullanılacaktır. Bu da programı karmaşık hale getirecek, yazmasını zorlaştıracaktır. Structure tanımladığımız zaman ise tüm alanları sadece structure adını kullanarak dosyaya yazabilecek, okuyabilecek veya kontrol edebileceğiz.

Structure tanımının genel yapısı aşağıdaki gibidir.

```
Structure Personel
```

```
    Dim Ad As String
```

```
End Structure
```

Personel tipini oluşturduk.

```
Dim Ogrenci As Personel
```

İfadesi ile Personel türünde Ogrenci adında bir değişken tanımlamış olduk.

```
Ogrenci.Ad = "Ali Çiriş"
```

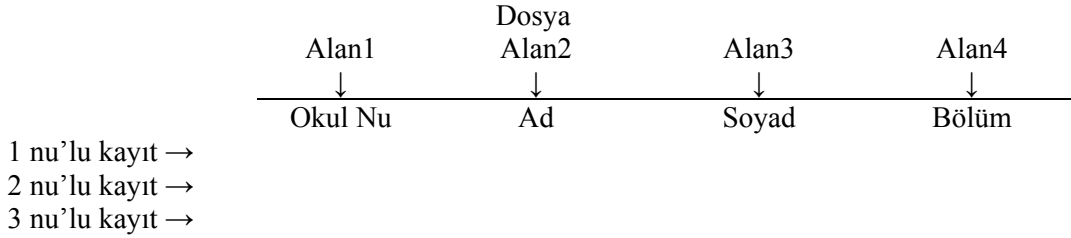
İfadesi ile de Ogrenci değişkeninin Ad alanına "Ali Çiriş" adını atadık.

1.2. Alan(Field) Kavramı

Dosyalar aynı yapıya sahip ve birbirleri ile ilişki içerisinde olan "kayıt"lar (records) topluluğudur. Kayıt ise, yapılan işlemler sırasında ana belleğe yazılan veya ana bellekten okunan veri birimidir. Bir kayıt içindeki ayrı ayrı veri parçalarına da "alan" (field) adı verilir.

Örneğin bir öğrenci bilgileri dosyasını ele alırsak, öğrencinin adı, soyadı, numarası, aldığı dersler ve bu derslerden aldığı notlar ayrı ayrı birer alandır. Bir öğrenciye ait bütün bu alanlar bir araya gelerek öğrencinin kaydını oluşturmakta ve tüm öğrencilere ait kayıtlar da bir araya gelerek öğrenci bilgileri dosyasını oluşturmaktadır

Yukarıdaki örnekte Ad ve DoğumTarihi birer alandır.



1.3. “SYSTEM.IO” Ad Alanı (name space)

Dosyalara ve akışlara (stream) okuma ve yazma için kullanılan sınıfları içerir. İlerideki konularda inceleyeceğimiz komutların birçoğu System.IO ad alanına aittir.

Programımızın General Declararion kısmına aşağıdaki komutu yazarak System.IO ad alanını yüklemiş oluruz.

```
Imports System.IO
```

1.4. Dosya (File), Klasör (Directory) ve Yol (Path) Deyimleri

1.4.1. Dosya(File)

Herhangi bir saklama biriminde bulunan, birbiri ile ilgili bilgilerin bir isim altında toplanmasına denir. Herhangi bir program ile çalışırken, çalıştığımız verileri kaydetmek istediğinizde program sizden bir dosya ismi isteyerek bu verileri daha sonra kullanmak üzere saklayacaktır. Sakladığınız bu veriler çalıştığımız programa göre ses, resim, görüntü, yazı ve benzeri olabilir. Sakladığınız bu veriler çalıştığımız programa göre sizin belirlediğiniz ismin dışında bir de türünü belirten uzantı olacaktır.

1.4.2. Klasör (Directory)

Saklama birimlerinin kapasiteleri arttıkça oluşturulan dosyaların bulunması daha da zorlaşmaktadır. Birbiri ile ilgili dosyaları bir arada saklamak için saklama birimlerinde oluşturulan ortamlara klasör adı verilir.

Örneğin Belgelerim klasöründe belgelerimizi, resimlerimizi, müziğimizi kendilerine ayrılan klasörlerde saklayarak aradığımız dosyaya daha kolay ulaşabiliriz.

1.4.3. Yol (Path)

Saklama birimlerinde var olan dizin adlarıdır. Örnek olarak C:\YEDEK veya A:\OKUL\ATL gösterebiliriz.

1.5. “Create Directory, Copy, Move, GetAttributes, SetAttributes, Delete ve Exists” Metotları

1.5.1. Create Directory

CreateDirectory metodu System.IO.Directory sınıfının bir üyesidir. Belirtilen saklama biriminde yeni bir klasör oluşturur.

```
Directory.CreateDirectory( yol )
```

Örnek:

```
Directory.CreateDirectory( "c:\okul" )  
C sürücüsünde okul adında bir klasör oluşturur.
```

```
Directory.CreateDirectory( "c:\okul\deneme" )  
C sürücüsünde okul ve içerisinde deneme adında iki klasör oluşturur.
```

```
Directory.CreateDirectory( "..\okul" )  
Aktif dizinde okul adında bir klasör oluşturur.
```

1.5.2. Copy

Copy metodu System.IO.File sınıfının bir üyesidir. Dosya kopyalar.

```
File.Copy( kaynak dosya adı , hedef dosya adı )  
Kaynak dosya adı : Kopyalanacak olan dosyanın yolu ve adı  
Hedef dosya adı : Oluşturulacak yeni dosyanın yolu ve adı
```

Örnek:

```
File.Copy( "c:\deneme.txt" , "d:\deneme.txt" )  
C sürücüsündeki deneme.txt adlı dosyayı, D sürücüsüne deneme.txt adıyla kopyalar.
```

```
File.Copy( "c:\okul.dat" , "c:\yedek.dat" )  
C sürücüsündeki okul.dat adlı dosyayı, C: sürücüsüne yedek.dat adıyla kopyalar.
```

1.5.3. Move

Move metodu, System.IO.File ve System.IO.Directory sınıflarının birer üyesidir. File.Move metodu dosya taşırken Directory.Move metodu klasör taşır. Directory.Move metodu kullanılırken kaynak ve hedef klasör aynı sürücüde bulunmalıdır.

```
File.Move(kaynak dosya adı, hedef dosya adı)
```

```
Directory.Move(kaynak dizin adı, hedef dizin adı)
```

```
File.Move("c:\okul.dat", "d:\okul.dat")
```

C sürücüsündeki okul.dat adlı dosyayı, d sürücüsüne okul.dat adıyla taşır.

```
Directory.Move("c:\okul", "c:\yedek\okul2006")
```

C sürücüsündeki okul adlı klasörü, C sürücüsündeki yedek klasörünün altına okul2006 adıyla taşır.

1.5.4. GetAttributes

GetAttributes metodu, System.IO.File sınıfının bir üyesidir. Klasör veya dosyanın özelliklerini gönderir.

```
File.GetAttributes(yol)
```

Özellik	Sayısal	Açıklama
VbNormal	0	Normal
VbReadOnly	1	Yalnız okunabilir
VbHidden	2	Gizli
VbSystem	4	Sistem
VbDirectory	16	Klasör
VbArchive	32	Arşiv

Örnek

```
If File.GetAttributes("c:\okul.dat") And vbHidden Then  
MsgBox("gizli")
```

veya

```
If File.GetAttributes("c:\okul.dat") And 2 Then MsgBox("Gizli")
```

yine aynı örneği

```
Dim a As Byte  
a= File.GetAttributes("c:\okul.dat")  
If a And vbHidden Then MsgBox("Gizli")
```

Şeklinde kullanabiliriz. Örnekte C sürücüsündeki okul.dat dosyasında gizlilik özelliği var ise mesaj kutusunda Gizli yazacaktır.

1.5.5. SetAttributes

SetAttributes metodu, System.IO.File sınıfının bir üyesidir. Klasör veya dosyanın özelliklerini değiştirir.

```
File.SetAttributes( yol ,özellikler)
```

Örnek

```
File.SetAttributes("c:\okul.dat", FileAttributes.Archive)
```

C sürücüsündeki okul.dat adlı dosyaya Arşiv özelliği verilir. Önceden farklı özellikleri varsa onlar kaldırılır.

```
File.SetAttributes(("c:\okul.dat",  
File.GetAttributes("c:\okul.dat") Or FileAttributes.Hidden)
```

C sürücüsündeki okul.dat adlı dosyanın eski özelliklerini kaldırmadan Gizlilik özelliğini ekler.

1.5.6. Delete

Delete metodu, System.IO.File ve System.IO.Directory sınıflarının birer üyesidir. File.Delete metodu dosya silerken Directory.Delete metodu klasör siler. Klasörün silinebilmesi için boş olması ve yazılabilir olması gerekmektedir. Silinen dosyalar geri dönüşüm kutusuna gönderilmez.

```
File.Delete(yol)  
Directory.Delete(yol)
```

Örnek

```
File.Delete("c:\okul.dat")  
C sürücüsündeki okul.dat adlı dosyayı siler.
```

```
Directory.Delete("c:\okul")  
C sürücüsündeki okul adlı klasörü siler.
```

1.5.7. Exists

Exists metodu System.IO.File ve System.IO.Directory sınıflarının birer üyesidir. File.Exists metodu dosyanın mevcut olup olmadığını kontrol ederken Directory.Exists metodu klasörün mevcut olup olmadığını kontrol eder. Geriye True veya False değerini gönderir.

```
File.Exists( yol )  
Directory.Exists( yol )  
  
If File.Exists("c:\okul.dat") Then MsgBox(" Dosya Bulundu")  
  
If Directory.Exists("c:\okul") Then MsgBox("Dosya Bulundu")
```

1.6. Metin Dosyalar ve İkilik Dosyalar (Text ve Binary)

1.6.1. Metin (Text) Dosyalar

Klavyeden basılan her tuş sırasıyla ve enter tusuna bastığımızda ascii kod karşılığı 10 ve 13 (satır sonu karakteri) olan iki karakter dosyaya yazılır. Bu formatta kullanılan dosyalar (mesela txt uzantılı dosyalar) text dosyalardır.

Bilgilerin manyetik veya optik ortama yazılması ve okutulması sıralı bir metin halindedir. Bilgiler manyetik veya optik ortama girildikleri gibi yazdırılır. Bu dosyalar herhangi bir metin editörü ile okunabilir.

1.6.2. İkilik (Binary) Dosyalar

İkon, resim, mp3 vb. tipli dosyalar belli anlamlar taşıyan byte dizileri tutar. Bu dosyaların içinde satır sonu karakterleri yoktur. İşte bu tip dosyalara ise ikilik (binary) dosyalar denilir.

1.7. Satır Sonu Bitirme Karakteri

Metin dosyalarında girilen karakterler formatlı olmadığı için hepsi yan yana çıkacaktır. Bu da dosyaya bir editörle bakıldığı zaman karmaşık bir görüntü verecektir. Aynı zamanda yapılan kayıtların birbirine girmesine neden olacaktır. Oysa her kayıttan sonra satırın bittiğini belirten satır sonu karakteri eklersek kayıtlara ulaşmak daha kolay olacaktır.

1.8. Input ve Output Stream Deyimleri

Tek karakterden geniş veri yapılarına, farklı boyut ve biçimdeki verileri işleyen giriş ve çıkış fonksiyonlarıdır.

1.9. FileStream Sınıfı

Dosya sisteminde dosyadan okur, dosyaya yazar, açar ve dosyaları kapatır. En iyi performans için giriş ve çıkış işlemlerinde önbellek kullanır.

Dim fs As FileStream=New FileStream (adı, modu, erişimi, paylaşımı)

Örnek

```
Dim fs As FileStream = New FileStream ("c:\ornek.txt",  
    FileMode.Open)End Class
```

1.10. FileMode

Dosyanın nasıl açılacağını belirler.

Özellik	Açıklama
Append	Eğer dosya var ise dosyanın sonuna konularak açar, veya yeni dosya oluşturur. FileMode.Append yalnızca FileAccess.Write ile birlikte kullanılabilir.
Create	Yeni bir dosya açılacağını belirler. Eğer dosya var ise üstüne yazılır.
CreateNew	Yeni bir dosya açılacağını belirler. FileIOPermissionAccess.Write gerektirir. Eğer dosya var ise IOException istisnası atılır.
Open	Var olan bir dosyanın açılacağını belirler. Dosyanın yetenekleri FileAccess ile belirlenmiş değerlere göre belirlenir. Eğer dosya yok ise System.IO.FileNotFoundException istisnası atılır.
OpenOrCreate	Eğer dosya var ise açar, yok ise yeni bir dosya oluşturur. Eğer dosya FileAccess.Read erişimiyle açılmışsa okumak için FileIOPermissionAccess.Read gerektirir. Eğer dosya FileAccess.Write erişimi ile açılmışsa yazmak için FileIOPermissionAccess.Write gerektirir. Eğer dosya FileAccess.ReadWrite erişimi ile açılmışsa okumak için FileIOPermissionAccess.Read and yazmak için FileIOPermissionAccess.Write gerektirir. Eğer dosya FileAccess.Append modunda açılmışsa FileIOPermissionAccess.Append gerektirir.
Truncate	Var olan bir dosyanın açılacağını belirtir. Dosya açıldığında içeriği silinerek açılır. FileIOPermissionAccess.Write gerektirir. Truncate ile açılmış dosyadan bilgi okumaya çalışmak istisnaya neden olur.

1.11. FileAccess

Dosyaya okumak, yazmak veya okuma/yazma erişimi için sabitler, tanımlar.


Read	Dosyaya okumak için erişir. Dosyadan veriler okunabilir.
ReadWrite	Dosyaya okuma ve yazma için erişir. Veriler dosyaya yazılıp dosyadan okunabilir.
Write	Dosyaya yazmak için erişir. Dosyaya veriler yazılabilir. Aynı anda okumak da isteniyorsa read/write erişimi kullanılmalıdır.

1.12. FileShare

FileStream nesneleri dışında aynı dosyaya erişimin çeşitli kontrolleri için sabitleri kapsar.

Özellik	Açıklama
Delete	Dosyanın silinmesine izin verir.
None	Değiştirilmiş dosyanın paylaşılmasını reddeder.
Read	Okunmak için dosyanın açılmasına izin verir
ReadWrite	Okumak veya yazmak için dosyanın açılmasına verir.
Write	Yazmak için dosyanın açılmasına izin verir.

Örnek: Şekil.1.1'deki formu oluşturuyoruz.



Şekil 1.1

General Declaration kısmına

Imports System.IO → System.IO ad alanını projemize ekledik.

Yaz butonu klik olayına

```
Dim file As String
```

```
file = "c:\deneme.txt" → Dosya adı için bir değişken
```

```
Dim fs As FileStream = New FileStream(file, FileMode.Append)
```

```
→ fs adında bir filestream, dosya ekleme modunda açıldı.
```

```
Dim w As StreamWriter = New StreamWriter(fs)
```

```
→ Dosyaya yazmak için w adında StreamWriter
```

tanımlandı.

```
w.WriteLine(textBox1.Text & " " & textBox2.Text)
```

```
→ Dosyaya yaz
```

```
w.Close()
```

```
fs.Close()
```


Oku butonu klik olayına

```
Dim file As String
Dim tmp As String
    file = "c:\deneme.txt"
Dim fs As FileStream = New FileStream(file, FileMode.Open)
Dim r As StreamReader = New StreamReader(fs)
    TextBox3.Text = ""
    Do
        tmp = r.ReadLine
        TextBox3.Text = TextBox3.Text & tmp &
Microsoft.VisualBasic.ControlChars.CrLf
    Loop Until tmp Is Nothing → CrLf karakteri her ögenin kendi satırında
    olmasını sağlar.

    r.Close()
    fs.Close()
```

1.12. EOF, LOF, FileCopy, FileClose, FileLen, FileGet, FileOpen, Kill, Print, Seek ve Write Komutları

1.12.1. EOF

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Dosya sonuna gelindiğinde True değeri gönderir. Dosya sonuna gelindiği zaman hatadan kurtarmak için kullanılır.

EOF(Dosya numarası)

1.12.2. LOF

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Açık olan dosyanın büyüklüğünü byte olarak Long veri türünde geri gönderir.

LOF(Dosya numarası)

1.12.3. FileCopy

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Dosya kopyalar.

FileCopy("kaynak dosya", "hedef dosya")

1.12.4. FileLen

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Dosyanın uzunluğunu byte cinsinden belirterek Long veri türünde geri gönderir. Açık olan bir dosyanın o andaki boyutunu öğrenmek için LOF fonksiyonu kullanılmalıdır.

FileLen("Dosya adı")

1.12.5. FileGet

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Disk üzerinde açık olan dosyadaki bilgiyi değişkene kopyalar.

FileGet (Dosya no, değişken, kayıt no)

1.12.6. FileOpen

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Girdi veya çıktı için dosya açar.

FileOpen (Dosya nu, dosya adı, açma modu, erişim, paylaşım, kayıt uzunluğu)

Açma modu (OpenMode); isteğe bağlıdır. Belirtilmediği takdirde Append modu seçilir.

Özellik	Açıklama
Append	Ekleme için dosya açar. Varsayılan.
Binary	İkili erişim için dosya açar.
Input	Yazma erişimi için dosya açar.
Output	Okuma erişimi için dosya açar.
Random	Rastgele erişim için dosya açar.

Erişim (OpenAccess); isteğe bağlıdır. Belirtilmediği takdirde Default modu seçilir.

Özellik	Açıklama
Default	Okuma ve yazma erişimi izinli. Varsayılan.
Read	Okuma erişimi izinli.
ReadWrite	Okuma ve yazma erişimi izinli.
Write	Yazma erişimi izinli.

Paylaşım (OpenShare); isteğe bağlıdır. Belirtilmediği takdirde Default modu seçilir.

Özellik	Açıklama
Default	LockReadWrite. Varsayılan
LockRead	Diğer prosedürler dosyayı okumasını engeller.
LockReadWrite	Diğer prosedürlerin dosyayı okuma ve yazmasını engeller.
LockWrite	Diğer prosedürler dosyaya yazmasını engeller.
Shared	Herhangi bir prosedür dosyaya okuyabilir veya yazabilir.

Kayıt uzunluğu

İsteğe bağlıdır. 32767 (byte)'den küçük veya eşit bir sayıdır. Rastgele erişimli açılmış dosyalar için kayıt uzunluğunu, sıralı erişimli açılmış dosyalar için ise karakter sayısını belirtir.

1.12.7. Kill

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Diskden dosya siler.

Kill(yol)

1.12.8. Print

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Formatlı gösterilmiş bilgiyi sıralı erişimli dosyaya yazar.

Print (dosya numarası, çıkış)

Çıkış; seçime bağlıdır. Sıfır veya daha fazla virgülle ayrılmış deyimi dosyaya yazar.

Ayarlar	Açıklamalar
SPC(n)	n sayısı kadar boşluk ekler. sed to insert space characters in the output, where n is the number of space characters to insert.
TAB(n)	İmleci n sayısı ile belirtilen sütuna taşır. n kullanılmamışsa, imleci sonraki print bölgesinin başlangıcına taşır. N Used to position the insertion point to an absolute column number, where n is the column number. Use TAB with no argument to position the insertion point at the beginning of the next print zone.
Expression	Sayısal veya string ifadeyi yazar.

1.12.9. Seek

Microsoft.VisualBasic.FileSystem sınıfının üyesidir. Dosyada işaretçiyi belirlenen pozisyona konumlandırır, pozisyon değeri girilmediği takdirde konumu geri döndürür.

Seek (Dosya numarası, pozisyon) → Verilen pozisyona konumlanır.

Seek(Dosya numarası) → İşaretçinin konumunu geri döndürür.

1.12.10. Write

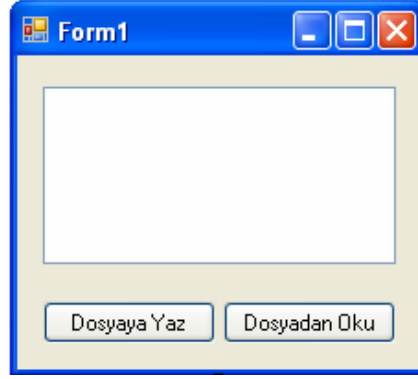
Sıralı dosyalara yazar.

write (dosya numarası, çıkış)

çıkış; seçime bağlıdır. Sıfır veya daha fazla virgülle ayrılmış deyimi dosyaya yazar.

Örnek:

Şekil 1.2'deki formumuza bir textbox (multiline (çok satırlı)), iki buton ekleyelim.



Şekil 1.2

Dosyaya yaz butonunun klik olayına

```
Dim dn As Integer
dn = FreeFile() → Kullanılmayan dosya numarası
FileOpen(dn, "c:\aaa.txt", OpenMode.Append)
PrintLine(dn, TextBox1.Text) → TextBox daki bilgiyi dosyaya yaz
FileClose(dn) → Dosyayı kapat
```

Dosyadan Oku butonunun klik olayına

```
Dim dn As Integer
Dim i As Integer
Dim x As String
dn = FreeFile() → Kullanılmayan dosya numarası
FileOpen(dn, "c:\aaa.txt", OpenMode.Input)
TextBox1.Text = "" → TextBox içeriğini temizle
For i = 1 To LOF(dn)
    → LOF(dn) dn numaralı dosyanın karakter sayısını
    verir. (Text tipi olduğu için)
    x = InputString(dn, 1)
    → dn dosyasından 1 karakter oku
    TextBox1.Text = TextBox1.Text & x
    → okunan x değişkenini TextBox'a ekle
Next
FileClose(dn) → Dosyayı kapat
```

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Farklı alan türlerindeki değişkenleri, yapı olarak bir araya getiriniz.➤ Akım komutları ile bir dosya oluşturunuz.➤ Akım komutu ile bir dosyayı okumak için açınız.➤ Bir metin dosyası içindeki bilgileri listeleyiniz.➤ Döngü içinde bilgileri yapı içine kaydediniz.➤ Yapıdaki bilgileri form üzerindeki uygun nesnelere aktarınız.➤ Dosyayı güncelledikten sonra, dosyayı kapatınız.	<ul style="list-style-type: none">➤ Açılacak dosyanın açılış moduna dikkat ediniz. Verileriniz kaybolabilir.➤ With....End With yapısını kullanınız.

ÖLÇME VE DEĞERLENDİRME

A.OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki cümlelerle boş bırakılan yerlere uygun ifadeleri getiriniz.

1. Birbiri ile ilgili bilgilerin bir isim altında toplanmasınadenir.
2. Birbiri ile ilgili dosyaları saklamak için oluşturulan bölümlere.....denir.
3. Bir kayıt içindeki ayrı ayrı veri parçalarınadenir.
Yol Klasör Dosya Alan Kayıt

Aşağıdaki sorular doğru / yanlış sorularıdır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

4. GetAttributes metodu dosya veya klasörün özelliklerini ayarlamamızı sağlar.()
5. Exist metodu dosya veya klasörün mevcut olup olmadığını kontrol eder.()
6. LOF ile FileLen komutu aynı işi yaparlar.()
7. Aşağıdaki FileMode listelerinde yeni bir dosya oluşturan seçenekleri işaretleyiniz.
 - FileMode.Append
 - FileMode.Create
 - FileMode.CreateNew
 - FileMode.Open
 - FileMode.OpenOrCreate
 - FileMode.Truncate

DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığı sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

ÖĞRENME FAALİYETİ-2

AMAÇ

Gerekli ortam sağlandığında projenize yeni XML dosyayı ekleyecek, veri yapısını oluşturacak, verileri birkaç yolla girebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken araştırmalar şunlardır:

- XML'nin nerelerde ve niçin kullanıldığını, veri tabanından ve internet sayfalarından farklarını araştırınız.
- Araştırma işlemleri için internet ortamını kullanınız.

2. XML DOSYALAR

2.1. XML (Extensible Markup Language) Yapısal Metin Dosyası

XML, HTML gibi bir biçimleme dili değildir. Sadece veriyi kullanım amacına uygun olarak etiketlemek için kullanılır. Zaten isminin açıklımı da genişletilebilir etiketleme dilidir. XML veri tanımlamayı sağlayan ve verilerin alışverişinde kullanılacak standart bir format olarak kullanılan bir dildir. Genişleyebilir, birlikte çalışabilir, veri tanımlayabilir. XML belgelerine kendi etiketlerinizi ekleyebilirsiniz. Bir XML belgesi metin (text) dosya olarak oluşturulan ya da SQL Server gibi çeşitli programlar tarafından oluşturulan bir dosyadır. Temel yapı hiyerarşik etiketlere dayanır. XML, yapısal verileri Web üzerinde kullanmak için geliştirilmiştir. Web sayfalarında biçimler ile veriler aynı koddadır. XML ile verileri biçimlerden ayırarak saklayabiliriz.

Örnek:

```
<eleman>  
<ad>ali</ad>  
<soyad>esmer</soyad>  
</eleman>
```

2.2. HTML, Web Sayfası, UTF-8

2.2.1. HTML

Hypertext Markup Language (HTML) belgelerin birbirlerine nasıl bağlanacaklarını ve belge içindeki metin ve resimlerin nasıl yerleşeceklerini belirleyen ve etiket (tag) denilen kod parçalarından oluşan bir sistemdir. Aslında HTML bir programlama dili değildir. Programlama dili, bir seri prosedür ve açıklamadan oluşur ve genelde dış bir veriye ulaşmayı amaçlar. Bir HTML belgesi ise, başlı başına verinin kendisidir. HTML veriler içine yerleştirilen "tag" parçaları, metnin, dolayısıyla belgenin, okuyan tarayıcı tarafından verileri nasıl işleneceğini belirler.

2.2.2. Web Sayfası

Yazı, resim, ses, film, animasyon gibi pek çok farklı yapıdaki verilere etkileşimli bir şekilde ulaşmamızı sağlayan bir çoklu ortam sistemidir. Bu ortamdaki her veri, başka bir veriyi çağırabilir (link). Link, aynı doküman içinde olabildiği gibi, fiziksel olarak başka bir yerde de (internet üzerindeki başka bir makinede) olabilir. Bütün bu farklı yapıdaki veriler uygun bir standart ile bir arada kullanılıp bir Web tarayıcısında görüntülenebilir. Web sayfaları, başka sayfalara ve değişik türden verilere linkler içermektedir. Buralara fare ile tıklayarak, başka sayfalara, oradan da başka sayfalara geçeriz. Web'in başka bir işlevi de birbirine bağlı bilgisayarlar arasında veri paylaşımı için kuralları olan, iyi bir grafik ara birimli bir işletim sistemi olmasıdır. Web ara yüzü ile, sabit diskimizdeki dosyalar arasında gezinebiliriz. Bir Web sayfasına ulaştığımızda her şey 4 ana fazda gerçekleşir: (1) bağlantı, (2) ne istediğimizin Web servisine iletilmesi, (3) cevap, (4) ilgili sayfaya yapılan bağlantının kesilmesi. Bu ana safhalar, Web üzerinde iletişimin kurallarını tanımlayan bir protokolü oluşturur. Bu protokole de, Hyper Text Transfer Protocol (HTTP) denir.

2.2.3. UTF-8

UTF-8, ASCII kullanımının ötesinde gelişen iletişim kuralları için tavsiye edilen karakter kümesidir. UTF-8 iletişim kuralı, genişletilmiş ASCII karakterleri ve dünyanın birçok yazım sistemini kapsayan 16 bitlik bir Unicode karakter kümesi olan UCS-2 dönüşümü için destek sağlar. UTF-8, karakter verilerinde ASCII veya genişletilmiş ASCII kodlaması kullanılarak elde edilebilenden çok daha geniş kapsamdaki adlara olanak tanımaktadır.

2.3. “Tag, Attribute, Comment ve Element”

2.3.1. Tag (etiket)

En genel anlamı ile “etiket”. XML verileri tanımlamak için kullanılır. Blog veya blogla ilişkili kullanıldığında gönderileri sınıflandırmaya yarayan kategori başlığı veya anahtar kelimeler olarak düşünülür. < ile başlayan > ile biten karakter grubuna açılış tag'i (etiket) denir. Örneğin <ad> bir tag'dir. </ ile başlayan ve > biten grup ise kapanış tag'i dir. Örneğin </ad> bir kapanış tag'idir. XML'de bir açılış tag'i varsa bu tag'in ileride kapanış tag'i olması gerekir.

2.3.2. Attribute

Attribute'ler çalışma ortamına bilgi vermek için kullanılan etiketlerdir. Attribute değerleri daima tırnak içine alınmalıdır. XML elementleri attribute'lara sahip olabilir ve bu attribute'lar ise tıpkı HTML'de olduğu gibi isim/değer(name/value) çiftlerini içerebilir.

```
<?xml version="1.0"?>
<not date="12/10/99">
</not>
```


2.3.3. Comment

XML dokümanlarında comment `<!-- Comments -->` şeklinde verilir. `<!--` ve `-->` arasındaki ifadeleri XML dosyasını işleyen uygulamalar görmez. Bunlar sadece XML yazan kişi veya program tarafından içeriğin daha iyi anlaşılmasını sağlamak için eklenir.

2.3.4. Element

Bir XML dokümanı iç içe konulmuş elementlerden oluşur. En basit element

```
<tag>
  Text
</tag>
```

şeklinde belirtilir. Tag name herhangi bir isim olabilir. Ancak açılan her element mutlaka kapatılmalıdır. Element'ler `<tag>` ile başlar `</tag>` ile kapatılır. XML büyük küçük harfe duyarlıdır. yani 'tag', 'Tag' ve 'TAG' ifadeleri farklı kabul edilir.

2.4. System. XML Sınıfı

Çeşitli XML işlemleri için temel destek sağlar.

2.5. Kök, Ana ve Çocuk Elemanlar

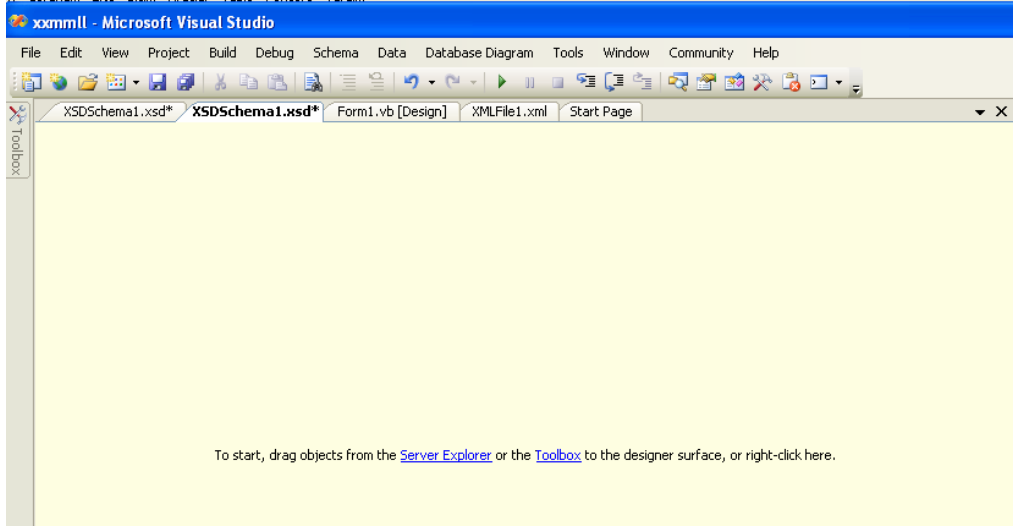
XML dokümanları kendini tarif eden bir yazıma sahiptir. Dokümanın ilk satırı XML bildirimini (XML declaration) olarak isimlendirilir. Bu kısımda XML dokümanının versiyon bilgisi tanımlanır. Elemanlar belirli bir düzen çerçevesinde iç içe geçmiş durumdadır. XML yazımında tüm XML dokümanlarının bir kök elemanı olması gerekmektedir. XML dokümanları içerisinde ilk tag kök tag (root tag) olarak isimlendirilir:

Bahsedilen bu kök elemanın altındaki tüm elemanlar ana ve çocuk olarak adlandırılır. Bu ana ve çocuk elemanlar ise daha önceden belirtilmiş olan yazım kurallarına uymak zorundadır. Yani elementlerin iç içe geçme durumları belirli bir hiyerarşiye göre olmaktadır.

```
<kök>
  <ana>
    <çocuk>....</çocuk>
  </ana>
  .....
</kök>
```

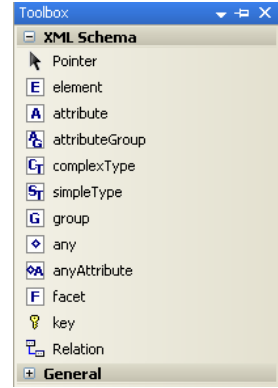
2.6. “XML Designer” penceresi

Yeni XML şemaları hazırlamak ve var olan XML şemalarını değiştirmek, tablolar arasında ilişkiler oluşturmak ve düzenlemek, birincil anahtar oluşturmak ve düzenlemek için kullanılır.



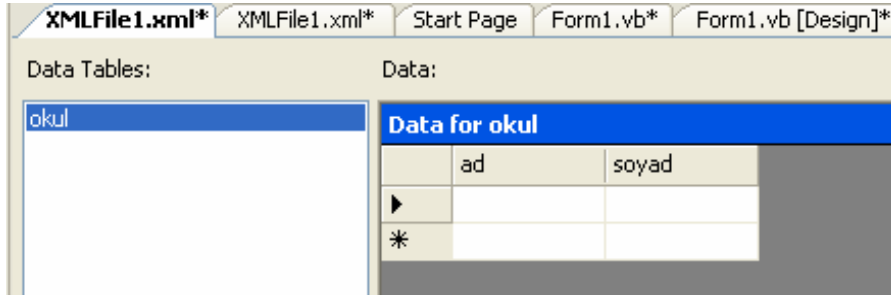
Şekil 2.1: XMLDesigner

XML designer çalıştırmak için projemize sağ tıklayarak add→new item tıklanarak xml files ve xml schema seçilerek projemize eklenir. Server explorer penceresinden veya şekil 2.2’deki araç kutusundan objeleri XML designer yüzeyine sürükleyerek veya sağ tıklayarak çalışmaya başlayabiliriz. XML seçili iken sağ tıklayarak designer penceresinden kod penceresine veya kod penceresinden designer penceresine view code veya view designer seçilerek geçilebilir.



Şekil 2.2

XMLFile seçili iken kök eleman girildikten sonra istediğimiz elemanları belirleyip, sağ tıklayarak View Code seçeneğinden şekil 2.3’teki gibi kayıtları hesap tablosu şeklinde girebiliriz.



Şekil 2.3

2.7. “XmlTextWriter, XmlTextReader” Sınıfı ve “Formatting” Özelliği

2.7.1. XmlTextWriter

XML belgeleri yazmak için XmlTextWriter sınıfı kullanılır. XmlTextWriter kullanarak XML belgesi yukarıdan aşağıya doğru yazılır.

2.7.2. XmlTextReader

XML belgeleri okumak için XmlTextReader sınıfı kullanılır.

2.7.3. Formatting

Çıkış formatının nasıl olacağını belirtir. Intender veya none seçilir. Intendet seçildiğinde girintili none seçildiğinde ise girintisiz yazılır.

2.7.4. Indentation

Formatting özelliğinde Formating. Indented ayarlı iken her bir hiyerarşiye kaç tane IndentChars'da belirtilen karakterden yazılacağını ayarlar veya alır.

2.7.5. IndentChar

Formmating özelliğinde Formating.Indented ayarlı iken girinti için hangi karakterin kullanılacağını ayarlar veya alır.

2.8. WriteStart Document, Write Comment, WriteAttributeString, WriteEndElement Komutları

2.8.1. WriteStartDocument

Bu metod ile XML dosyası yazmaya başlıyoruz. XML dökümanları aslında text dokümanlardır ve bir başlık kaydı içerir. Bu dokümanın XML verisi ve kodlama bilgisini taşıyan satırdır. Her XML dokümanda ilk satır olarak yazılır.

2.8.2. WriteComment

Açıklama satırı yazmamızı sağlar. Bu satırlar işleme alınmaz, sadece kullanıcıya veya programcıya bilgi vermek amacıyla yazılır. <!--.....--> arasında bulunur.

2.8.3. WriteAttributeString

Yeni bir eleman ve değerini bildiriyoruz.

2.8.4. WriteEndElement

XML dosyasını sonlandırıyoruz.

2.9. Read, Close, ReadStartElement, ReadEndElement Komutları

2.9.1.Read

XML dosyasından bilgi okur.

2.9.2. Close

XML dosyasının kapatılacağını bildirir.

2.9.3. ReadStartElement

Okunmaya başlanacak olan XML dosyasının kök elemanını bildirir.

Örnek:

Yapacağımız örnek ile XML dosyasını oluşturacak form üzerindeki textboxlara girilen eleman ve değerleri XML dosyasına yazarak tekrar geri okuyacağız. Form üzerine eleman ve değer için iki tane textbox, okumak için bir tane çok satırlı textbox, yazmak ve okumak için iki tane buton ekleyelim.

```
Imports System.Xml          → System.Xml sınıfını tanıtır.
```

```
Public Class Form1
```

```
Public Class ornek
```

```
Public eleman As String
```

Form üzerindeki eleman ve değer

```
Public deger As String
```

bilgisini saklayacak sınıf (ornek)

```
End Class
```

```
Public Class ornekkolleksiyon
```

```
Inherits System.Collections.CollectionBase
```

Form üzerindeki eleman ve

```
Public Sub add(ByVal aornek As ornek)
```

değer bilgisini saklayacak

```
List.Add(aornek)
```

kolleksiyon

```
End Sub
```

```
End Class
```

```
Dim theornek As New ornekkolleksiyon()
```

→ ornekkolleksiyon sınıfını temel alan bir nesne (theornek)

```
Private Sub Button1_Click(ByVal sender As System.Object, _
```

```
ByVal e As System.EventArgs) Handles Ekle.Click
```

→Form üzerinde textboxlara girilen eleman ve değer değişkenlere aktarılarak koleksiyona eklenir.

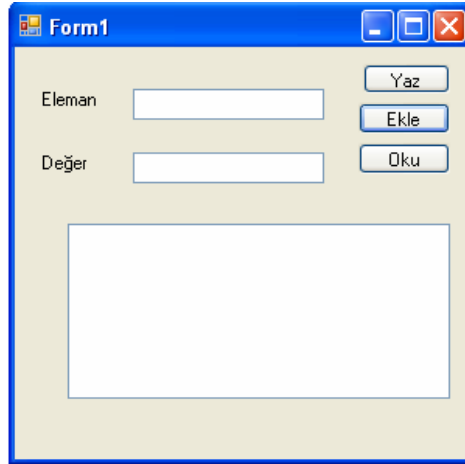
```
Dim aornek As New ornek()
```

```
aornek.eleman = txtelement.Text
```

```
aornek.deger = txtvalue.Text
```

```
theornek.add(aornek)
```

```
End Sub
```



Şekil 2.4

```
Private Sub Button3_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Yaz.Click
    → Yaz butonu click olayı
    Dim aornek As New ornek() →ornek sınıfında tanımlanmış aornek nesnesi
    Dim xmlyaz As New XmlTextWriter("ornek.xml", Nothing)
    → XmlTextWriter tanımı
```

Eğer burada olduğu gibi yol belirtilmezse projenin bin klasörüne oluşturur. Kodlamaya nothing diyerek kodlamanın UTF-8 olarak yazılmasını sağlar. Eğer dosya önceden var ise üzerine yazılır.

```
With xmlyaz → Her satırda nesnenin adını tekrar yazmamak için kullanılır.
.Formatting = Formatting.Indented → XML dosyasının girintili olması için
.Indentation = 3 → Girinti karakterinden 3 tane kullanılacak
.IndentChar = " " → Girinti karakteri olarak boşluk kullanılacak
.WriteStartDocument() → XML dosyasına yazmaya başlıyoruz.
.WriteComment("Bu satır açıklama satırındır") → Açıklama satırı
.
    → Kök eleman "kitap"
```

```
For Each aornek In theornek
    → theornek koleksiyonundaki her bir aornek nesnesi için; yani form üzerinde girilen
ve ekle denilen her bir eleman ve değer için
```

```
.WriteElementString(aornek.eleman, aornek.deger) →Değer taşıyan elementi
yazıyoruz.
Next
```

```
.WriteEndElement() → WriteStartElement("kitap") ile başladığımız kök elemanı
bitiriyoruz.
.Close() →İşlemi sonlandırıyoruz.
```

```

End With

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles oku.Click → oku butonu click olayı
Dim xmloku As New XmlTextReader("ornek.xml") → XmlTextReader tanımı
    xmloku.ReadStartElement("kitap") → Kök elemanın "kitap" olduğu belirtildi.

Do While xmloku.Read() → okunan bilgi kitap oluncaya kadar hepsini oku.
If xmloku.Name = "kitap" Then Exit Do → Eğer kök eleman okunursa (kitap) çık.

TextBox1.Text = TextBox1.Text & _
    Microsoft.VisualBasic.ControlChars.CrLf & _
    xmloku.ReadOuterXml
    Loop
    xmloku.Close() İşlemi bitir.
End Sub
End Class

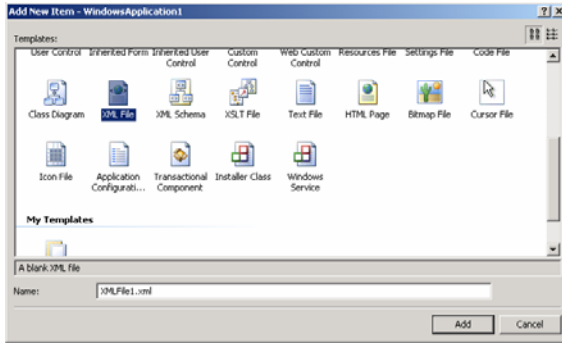
```

ReadOuterXml ile XML dosyası içindeki eleman ve değerleri çok satırlı bir textbox içerisine okuyabiliriz. CrLf karakteri her öğenin kendi satırında olmasını sağlar.

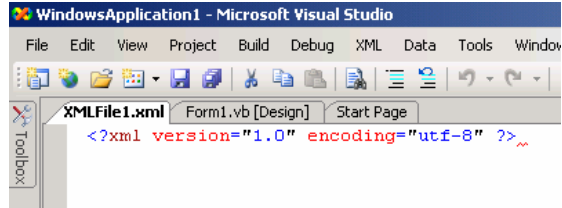
UYGULAMA FAALİYETİ

İşlem Basamakları

- “Project*Add New Item...*XML File” ile projeye yeni bir XML dosya ekleyiniz.



- Açılan sayfada istenen yapıda veri oluşturarak veri ekleyiniz.



- “Data View” komutu ile başka bir yöntemle veri giriniz.
- Program komutları ile veri okuyunuz.
- Program komutları ile veri girişi yapınız.

Öneriler

- Büyük küçük harf duyarlı olduğunu unutmayınız.
- Etiketlerin sıralamasında dikkat ediniz
- XML kod sayfasında sağ tıklayarak ve “xml view data grid” seçilerek grid yapıya geçilir.

ÖLÇME VE DEĞERLENDİRME

A.OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki sorular doğru / yanlış sorularıdır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. XML bir biçimleme dilidir. ()
2. XML ile HTML dillerinin tek benzer yanları (tag) etiketleridir. ()
3. XML dosyasını hazırlamak için SQL Server gerekmektedir. ()
4. XML dosyaları büyük küçük harf duyarlıdır. ()
5. <!--.....--> arasına yazılan ifadeler XML dosyası başlık kısmıdır ve XML dosyasının kullanacağı iletişim kuralını belirler. ()
6. Açılan her tag (etiket) kapanmak zorundadır. ()

DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığı sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

ÖĞRENME FAALİYETİ-3

AMAÇ

Gerekli ortam sağlandığında veri tabanına tablolar oluşturacak, bu tablolardan verileri seçecek ve üzerlerinde değişiklikler yapabileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken araştırmalar şunlardır:

- Veri tabanı nerelerde, niçin kullanılır? Araştırınız.
- Dosya özelliklerini araştırınız.

Araştırma işlemleri için internet ortamını ve veri tabanı kullanılan okul, hastane ve bankaları ziyaret ediniz.

3. ADO.NET(ACTIVEX DATA OBJECTS) VE OLEDB

3.1. Data (Veri), Grid (ızgara) Nesneleri

3.1.1. Data (Veri)

İşlenmemiş bilgi veya bilginin ham halidir. Bilgi ise, en basit anlamda verinin işlenmiş şeklidir. Bilgi ile veri arasındaki farkı şu örnekle açıklayabiliriz. Doğum tarihi ham veridir. Bugünün tarihinden doğum tarihini çıkararak bugünkü yaşı elde ederiz. Yaş ise bilgidir.

3.1.2. Grid (Izgara)

Grid, Access veya Excel gibi bir programda görebileceğimiz bir biçimde form üzerinde satır ve sütunlardan oluşan, herhangi bir türde sekmeli veri görüntülemek için kullanılabilen bir nesnedir.

3.2. ADO.NET ve SQL (Structured Query Language)

3.2.1. ADO.NET

ADO (ActiveX Data Object), Microsoft'un tüm Windows tabanlı programlama dillerinde programcıların veri tabanı işlemlerini kolay, hızlı ve güvenli bir şekilde yapması için oluşturduğu bir ara yüzdür. ADO.NET ise ADO'ya göre daha geniş veri erişim şekilleri sunmaktadır. Ayrıca ADO.NET bağlantısız mimari ile çalışır. ADO'da bağlantı uygulama açık kaldığı sürece açık kalırdı. ADO.NET'te ise bağlantı kurulur, tablolar DataSet'e alındıktan sonra bağlantı kesilir. ADO.NET özellikle internet kullanımı için tasarlanmıştır. XML ADO.NET'in iç veri biçimi olmuştur.

3.2.2. SQL (Structured Query Language)

SQL (Structured Query Language), bir veri tabanı sorgu dilidir. Program geliştiricileri ve veri tabanı kullanıcıları, bir veri tabanına veri eklerken, silerken, güncellerken veya sorgularken bu dili kullanırlar.

3.3. “Select, Insert, Update, Delete” SQL Komutları

3.3.1. Select

Belirtilen tablodan veri seçmek için kullanılır.

SELECT alan adları FROM tablo adı
Tablo1

ad	soyad	adres	sehir
Halil	Gündüz	Bahçelievler 23	Ankara
Ali	Çiriş	Harıkcı 15	Adıyaman
Ayşe	Çetinkaya	Nailbey 12	Elazığ
Mehmet	Çiçek	Demetevler 34	Ankara

SELECT *FROM tablo1
Tablo1 adlı tablonun tüm alanlarını ve kayıtlarını listeler.

SELECT ad, soyad FROM tablo1

ad	soyad
Ali	Çiriş
Ayşe	Çetinkaya
Halil	Gündüz
Mehmet	Çiçek

Operatör	Açıklama
=	Eşittir
<>	Eşit değil
>	Büyük
<	Küçük
>=	Büyük eşit
<=	Küçük eşit
BETWEEN	Arasında
LIKE	Örnek ara

SELECT *FROM tablo1 WHERE sehir ='ankara'
Tablo1'de Sehir alanı Ankara olan kayıtları listeler

ad	soyad	Adres	sehir
Halil	Gündüz	Bahçelievler 23	Ankara
Mehmet	Çiçek	Demetevler 34	Ankara

SELECT * FROM tablo1 WHERE ad LIKE 'a*'
Tablo1’de ad alanı “a” harfi ile başlayan kayıtları listeler.

ad	soyad	Adres	sehir
Ali	Çiriş	Harıkçı 15	Adıyaman
Ayşe	Çetinkaya	Nailbey 12	Elazığ

SELECT DISTINCT sehir FROM tablo1

Aynı adla kayıtlı iki veya daha fazla şehir varsa bunlardan bir tanesini seçer ve listeler.
Örneğin tablo1’de şehir adı Ankara olan 3 kayıt varsa sadece bir tane Ankara listeler.

sehir
Adıyaman
Elazığ
Ankara

SELECT ad, soyad FROM tablo1 ORDER BY ad

Tablo1’deki kayıtların ad, soyad alanlarını seçer ve ad alanına göre küçükten büyüğe listeler.

ad	soyad
Ali	Çiriş
Ayşe	Çetinkaya
Halil	Gündüz
Mehmet	Çiçek

3.3.2. Insert

Tabloya yeni kayıt ekler.
INSERT INTO tablo1 VALUES ('Mehmet', 'Yıldırım', 'Harhar 25', 'Adıyaman')
INSERT INTO tablo1 (ad, soyad) VALUES ('Kamil', 'Kadioğlu')

ad	soyad	Adres	sehir
Halil	Gündüz	Bahçelievler 23	Ankara
Ali	Çiriş	Harıkçı 15	Adıyaman
Ayşe	Çetinkaya	Nailbey 12	Elazığ
Mehmet	Çiçek	Demetevler 34	Ankara
Mehmet	Yıldırım	Harhar 25	Adıyaman
Kamil	Kadioğlu		

3.3.3. Update

Tablodaki veriyi deęiřtirir.

```
UPDATE tablo1 SET adres = 'Çiftlik 34', sehir = 'Samsun'  
WHERE ad = 'Ali' AND soyad='Çiriř'
```

ad	soyad	Adres	sehir
Halil	Gündüz	Bahçelievler 23	Ankara
Ali	Çiriř	Çiftlik 34	Samsun
Ayře	Çetinkaya	Nailbey 12	Elazığ
Mehmet	Çiçek	Demetevler 34	Ankara
Mehmet	Yıldırım	Harhar 25	Adıyaman
Kamil	Kadıođlu		

3.3.4. Delete

Tablodan kayıt siler.

```
DELETE FROM tablo1 WHERE soyad = 'Gündüz'
```

ad	soyad	Adres	sehir
Ali	Çiriř	Çiftlik 34	Samsun
Ayře	Çetinkaya	Nailbey 12	Elazığ
Mehmet	Çiçek	Demetevler 34	Ankara
Mehmet	Yıldırım	Harhar 25	Adıyaman
Kamil	Kadıođlu		

3.4. İliřkisel Veri Tabanı, Tablo, İndeks

3.4.1. İliřkisel Veri Tabanı

Veri tabanı tek bir tabloya sığdırılmalıdır. Bu, birkaç kayıtta ortak olan herhangi bir bilginin her kayıtta tekrarlanacağı anlamına gelir. İliřkisel veri tabanında ise birçok farklı tablo kullanılır ve tablolar arasında iliřkiler oluşturulur. Bir iliřki, bir tabloya, başka bir tablodaki kaydı bađlanmamızı sađlar. Bu řekilde veriler daha az yer kaplar ve güncelleme kolaylařır.

3.4.2. Tablo

Bir veri tabanında verilerin saklandıđı kısımlardır. Bir veri tabanında bir veya daha fazla tablo olabilir. Örneđin “okul” isimli veri tabanında “personel”, “öđrenci” ve “sınıflar” isimli 3 adet tablo olabilir. Her sütün, bir bilgi kategorisi olan bir alanı her satır ise bir öđe için bilgi saklayan kayıttan oluşur.

3.5. Microsoft SQL Server 2005 Desktop Engine (MSDE), DBMS

Microsoft SQL Server 2005 Desktop Engine (MSDE), herhangi bir grafik ara yüze sahip olmamasına karşın sunduğu SQL Server 2005 veri tabanı motoru (engine) ve bu sayede sağladığı ölçeklenebilir ve güvenli veri iletimi sayesinde programların dağıtım aşamasında kullanılmaktadır. DBMS (veritabanı yönetim sistemi - database management system) ise bir veri tabanı oluşturup üzerinde çeşitli işlemler yapılmasını sağlayan programlar topluluğudur.

3.6. LAN, WAN, İnternet, İnternet Deyimleri

3.6.1. LAN (Local Area Network)

Yerel ağ anlamına gelen LAN, bilgisayarların fiziksel olarak birbirlerine bağlanmasına denir. Örneğin bir okulda laboratuvarlardaki bilgisayarların birbirlerine olan bağlantıları ve laboratuvarlar arasındaki bağlantıyı verebiliriz. Yerel ağ çok uzun mesafeli olmaz.

3.6.2. WAN (Wide Area Network)

Geniş ağ anlamına gelen WAN, bilgisayarların fiziksel olarak bağlanamayacakları bilgisayarlara telefon hatları, uydu aracılığıyla bağlanmasına denir. WAN birçok LAN'ın kendi arasında birleşimidir.

3.6.3. İnternet

İnternet, birçok bilgisayar sisteminin birbirine bağlı olduğu, dünya çapında yaygın olan ve sürekli büyüyen bir iletişim ağıdır. İnternet, insanların her geçen gün gittikçe artan "üretilen bilgiyi saklama/paylaşma ve ona kolayca ulaşma" istekleri sonrasında ortaya çıkmış bir teknolojidir. Bu teknoloji yardımıyla pek çok alandaki bilgilere insanlar kolay, ucuz, hızlı ve güvenli bir şekilde erişebilmektedir.

3.6.4. İnternet

İnternet (iç ağ), sadece belirli bir kuruluş içindeki bilgisayarları, yerel ağları (LAN) ve geniş alan ağlarını (WAN) birbirine bağlayan, çoğunlukla TCP/IP tabanlı bir ağıdır. İnternet'in daha özel bir halidir. İnternet'ler gateway'ler ile diğer networklere bağlanabilir. Temel oluşturulma amaçları, kuruluş bünyesinde bilgileri ve bilgi işlem kapasitesini paylaşmaktır. İnternet'ler, şirketler içi telekonferans uygulamalarında ve farklı birimlerdeki kişilerin bir araya gelebildiği iş gruplarının oluşturulmasında da kullanılır.

İnternet'ler üzerinden HTTP, FTP vb. pek çok protokol uygulamaları çalıştırılabilir. Günümüzde, İnternet'ler içinde, Web erişimi ile kaynakların kullanımı oldukça yaygındır. Bazı şirketlerdeki intranet'lerden, "Firewall" sistemleri üzerinden (bazı emniyet tedbirleri ile), internet çıkışı da yapılmaktadır. Bu sayede, her iki yönde de ileti trafiği kontrol edilebilmekte ve güvenlik sağlanmaktadır.

3.7. “Dataset, Data Adapter, Data Connection, Data Command” Nesneleri

3.7.1. Dataset

ADO.NET bağlantısız veri saklamak için DataSet kullanır. Bir Dataset birden fazla tabloyu tamponunda saklayabilir. Datasetin tamponunda yer alan tablolar arasında ilişki kurabilir ve istediğiniz veriyi elde edebilirsiniz. Bir Dataset bir veri tabanının küçük bir kopyası gibi çalışır. Dataset uyarlayıcısı veri güncellemek ve almak için gerekli metotlara sahiptir.

Datasetler XML ile tam olarak uyumludur. Dataset içindeki bir bilgi bir bileşenden başka bir bileşene aktarılacaksa o zaman XML kullanılır.

Dataset bir ya da daha fazla veri tablosudur. Veri bağlantı katmanı veriyi veri tabanından çeker ve veri seti oluşturmak için kullanır.

3.7.2. DataAdapter

DataAdapter, sahip oldukları veri sağlayıcılarının izin verdiği veri kaynakları ile sistem belleği üzerinde yer alan bağlantısız katman nesneleri arasındaki veri alışverişinin kolay, güçlü ve verimli bir şekilde sağlanmasından sorumludur.

DataAdapter sınıflarının; veri kaynağından verileri alıp bağlantısız katman nesneleri olan DataSet ve DataTable nesnelere doldurulmasından ve ayrıca, bağlantısız katman nesnelere taşıdığı verilerdeki değişikliklerin de veri kaynağına yansıtılmasından sorumlu olduğunu söyleyebiliriz.

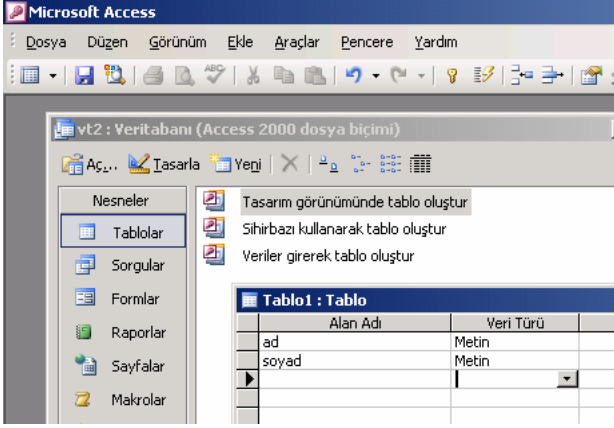
3.7.3. DataConnection

DataAdapter ile veri sağlayıcılarının arasındaki bağlantıyı sağlar. Bağlantı OLEDB, ODBC, SQL Server, Oracle veya diğer veri kaynakları olabilir. Bağlantı istenildiği zaman açılıp kapanabilir.

3.7.4. DataCommand

Veri kaynağına SQL deyimlerini uygular.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>➤ Bir veri tabanında saklanan tablolar oluşturunuz.</p>  <p>➤ “Select” komutu ile tablodan verileri alınız.</p> <p>➤ Veri üzerinde değişiklikler yapınız (Insert, Delete veya Update)</p> <p>➤ Veritabanının işlemleri bitince bağlantıyı kapatınız.</p>	<p>➤ Alan tiplerini uygun seçiniz.</p> <p>➤ Select komutuyla beraber Where, Order By, And, Or deyimlerini kullanarak seçimi özelleştiriniz.</p>

ÖLÇME VE DEĞERLENDİRME

A. OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki sorulardan; sonunda parantez olanlar doğru / yanlış sorularıdır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. Bilgi ile veri aynı kavramlardır. ()
2. Veri tabanındaki kayıtlar tablolarda saklanır. ()
3. SELECT * FROM tablo komut satırı tablo tablosundaki tüm kayıt ve alanlarını seçer. ()

Ad	Soyad	Şehir
AHMET	SALTUK	BİNGÖL
MURAT	ŞENTÜRK	KONYA
HASAN	ERDOĞAN	İSTANBUL

Aşağıdaki ifadelerin seçeceği bilgileri yazınız.

4. SELECT ad FROM tablo WHERE şehir='İSTANBUL'
5. SELECT ad FROM tablo WHERE ad LIKE 'a*'

DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığı sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

ÖĞRENME FAALİYETİ-4

AMAÇ

Gerekli ortam sağlandığında Data Adapter Configuration Wizard kullanarak veri tabanlarına daha hızlı bir biçimde ulaşacak, verileri DataGrid kontrolleriyle izleyip güncelleyebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken araştırmalar şunlardır:

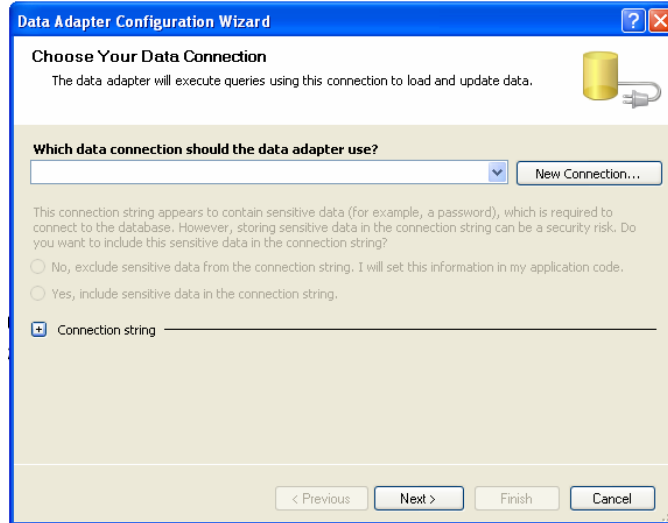
- Veri bağlı nesnelerin kullanım yerlerini araştırınız.
- Araştırma işlemleri için internet ortamını kullanınız.

4. VERİ BAĞLI NESNELER

4.1. Data Adapter Configuration Wizard

Data Adapter Configuration Wizard projemize Data Adapter eklemek için hazırlanmış bir sihirbazdır. Data Adapter Configuration Wizard çalıştırmak için formumuzun dizayn ekranında iken Toolbox'dan DataAdapter seçilerek form üzerine taşınır. Eğer Toolbox'ta DataAdapter nesnesi yok ise Toolbox'a sağ tıklanır. Choose Items tıklanarak .NET Framework Components'ten Data Adapter bulunarak seçilir.


Data Adapter form üzerine taşındığında Data Adapter Wizard ekrana gelecektir (Şekil4.1).

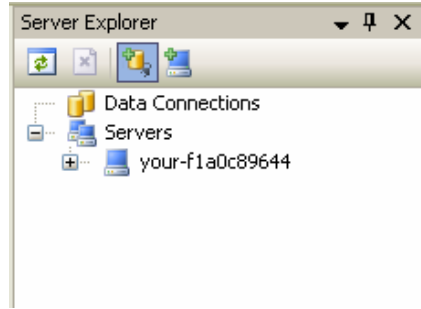


Şekil 4.1: Data Adapter Configuration Wizard

4.2. “Connection” Kavramı

Bağlanılacak olan veri kaynağını seçmek için kullanılır.

Şekil 4.1’deki Data Adapter Configuration Wizard penceresinde New Connection butonuna tıklayarak veya Data Adapter Configuration Wizard çalıştırmadan şekil 4.2’deki Server Explorer’da Connect to Database  butonu kullanılarak



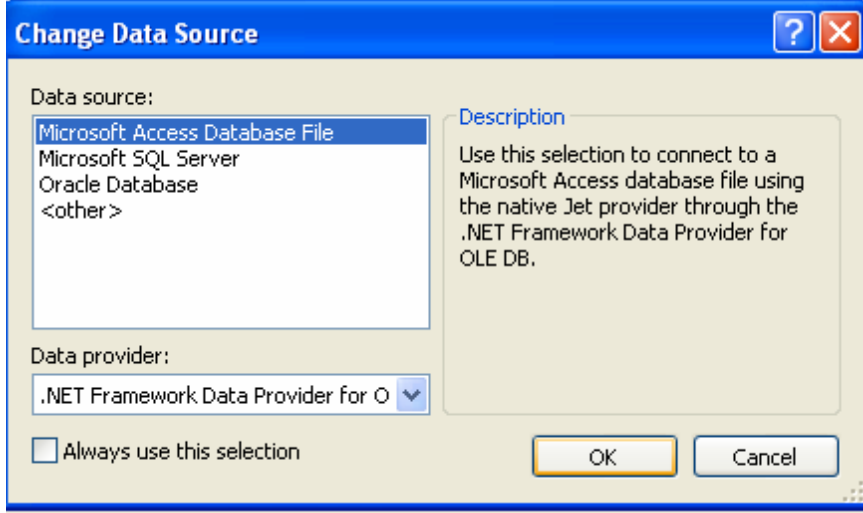
Şekil 4.2. Server Explorer

şekil 4.3’teki Add Connection penceresine ulaşabiliriz.



Şekil 4.3 AddConnection

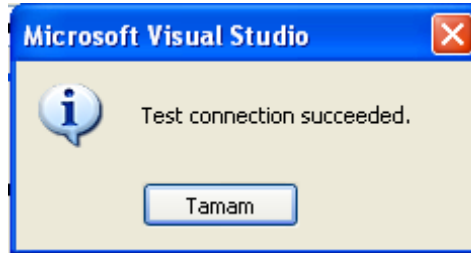
Burada Data Source (veri kaynağı) değiştirebiliriz. Microsoft Access dosyası, Microsoft SQL Server veya Oracle veri tabanı veya farklı bir kaynak seçilebilir.



Şekil 4.4: Change Data Source

Veri kaynağımızı seçip OK butonuna tıklayarak Add Connection penceresine döneriz. Browse butonuna tıklayarak veri tabanı dosyamızı seçeriz.

Test Connection butonuna tıklayarak bağlantımızın başarılı veya başarısız olduğunu kontrol ederiz. OK butonu ile tekrar Data Adapter Configuration Wizard penceresine döneriz.

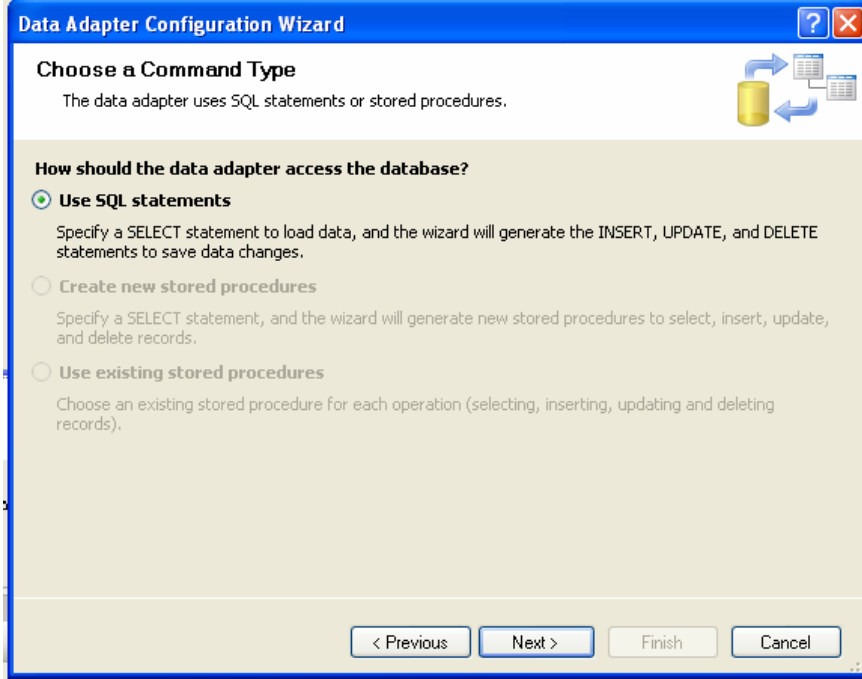


Şekil 4.5: Bağlantı testi başarılı

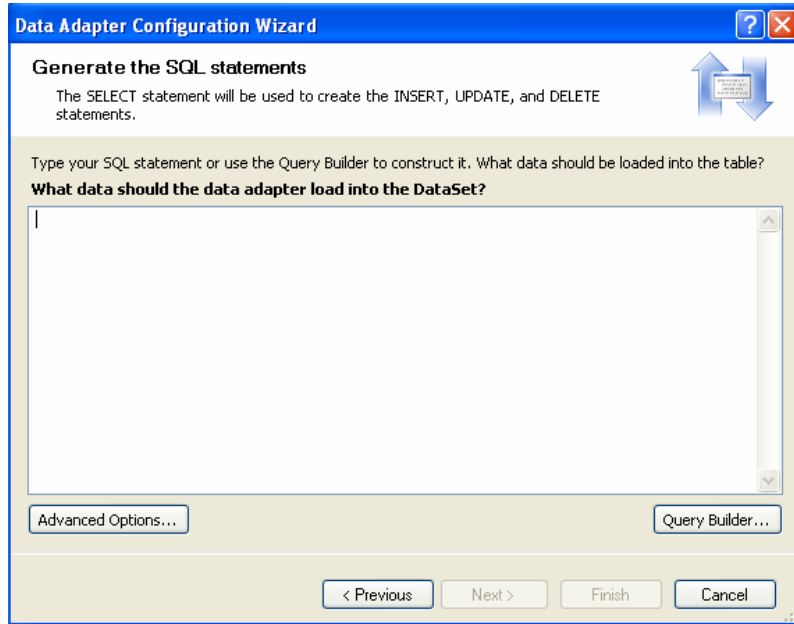
4.3. Query Builder Penceresi

SQL ifadelerini Query Builder kullanarak rahatça yazabiliriz.

Data Adapter Configuration Wizard penceresinde Use SQL statements seçili iken Next butonu tıklanır.

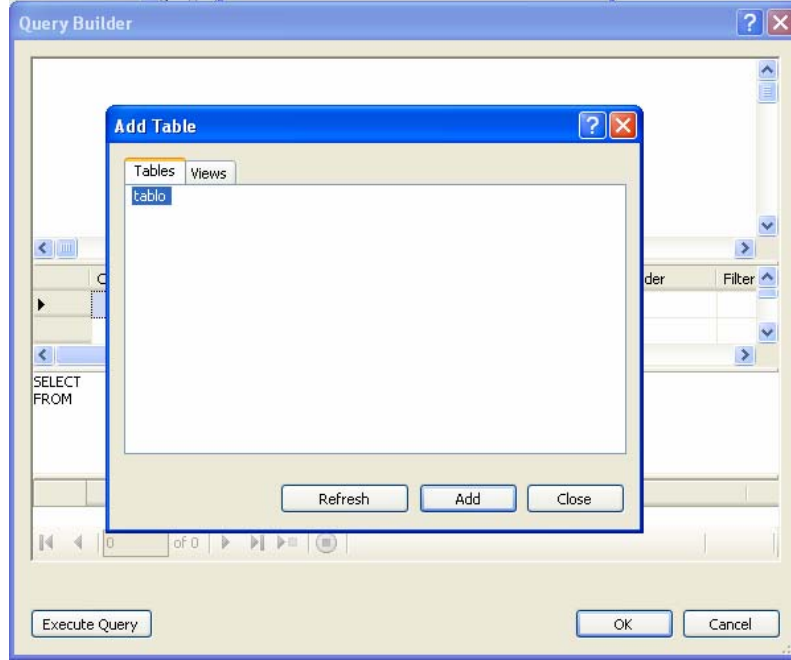


Şekil 4.6



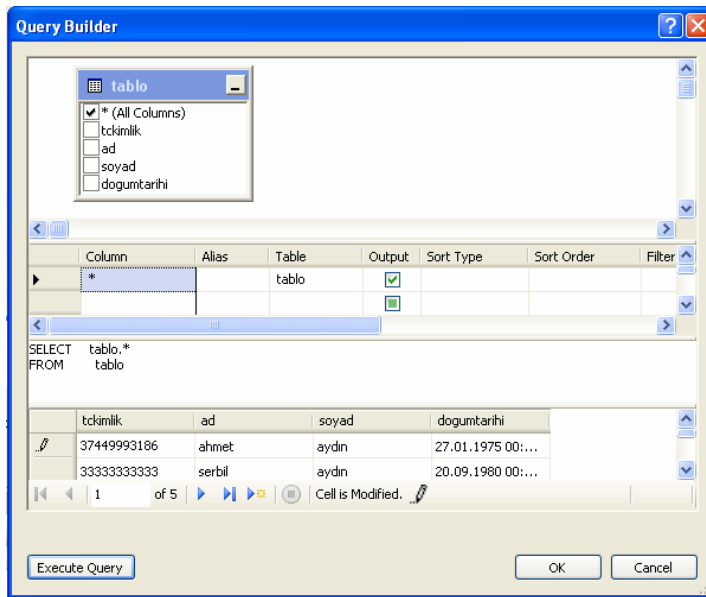
Şekil 4.7

Şekil 4.7 de Query Builder... tıklanıldığında



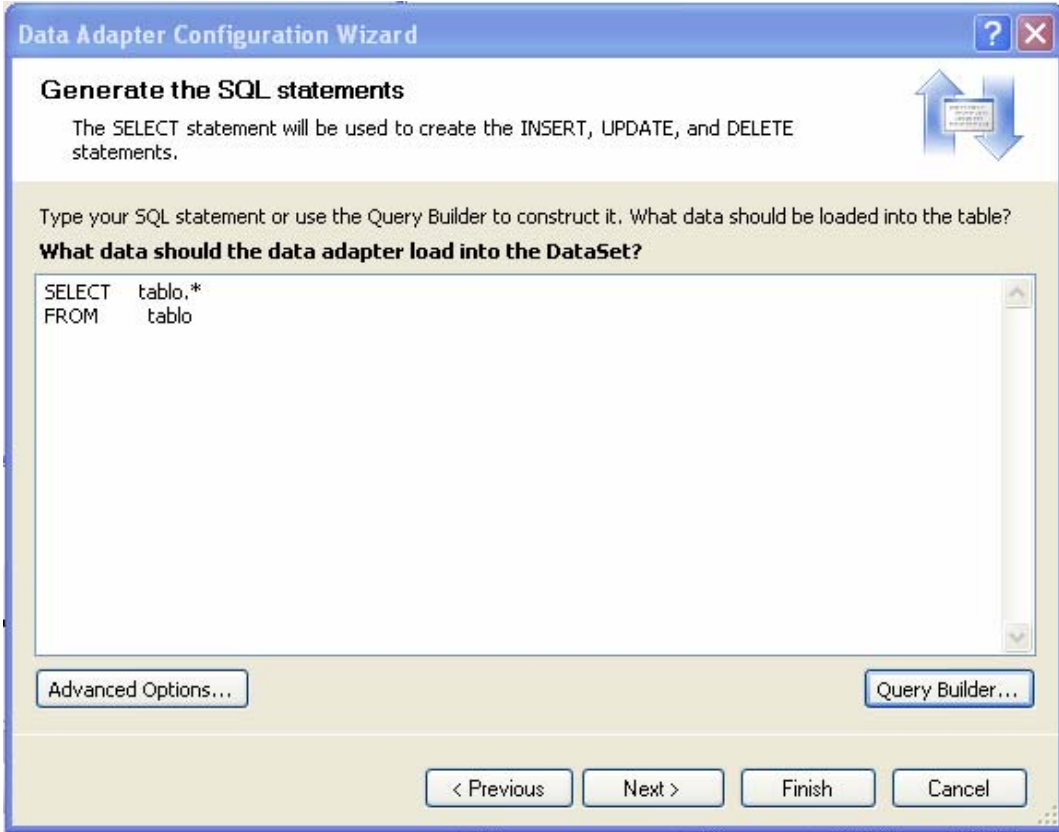
Şekil 4.8: Query Builder

Query Builder penceresi ve üzerinde veri tabanı dosyamızdaki tabloların listelendiği Add Table penceresi gelecektir. İstenen tablolar seçilip Add butonu tıklanarak tabloları Query Builder penceresine yerleştirmiş oluruz. Close butonuyla Add Table penceresi kapatılarak Query Builder penceresine dönülür.



Şekil 4.9

Tablodan hangi alanlar listelenecek ise seçilir, Execute Query ile sorgu uygulanarak tablo içindeki veriler listelenir. OK butonu ile tekrar Data Adapter Configuration Wizard penceresine dönülür.

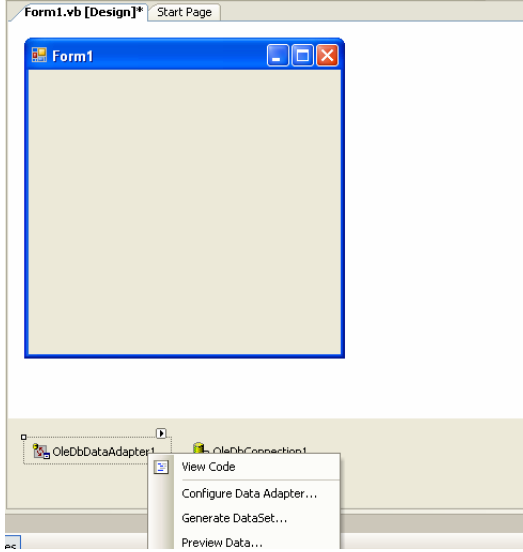


Şekil 4.10

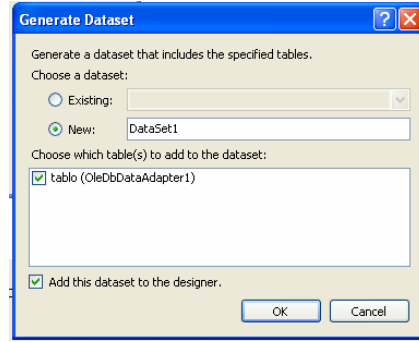
Finish butonu ile sihirbaz kapatılır. Bu işlem sonunda DataAdapter ve Connection bileşenini oluşturmuş olduk.

4.4. Dataset, Datagrid Nesneleri

DataSet oluşturmak için Component Tray'da Data Adapter sağ tıklanarak Generate DataSet... seçilir (Şekil.4.11).



Şekil 4.11



Şekil 4.12

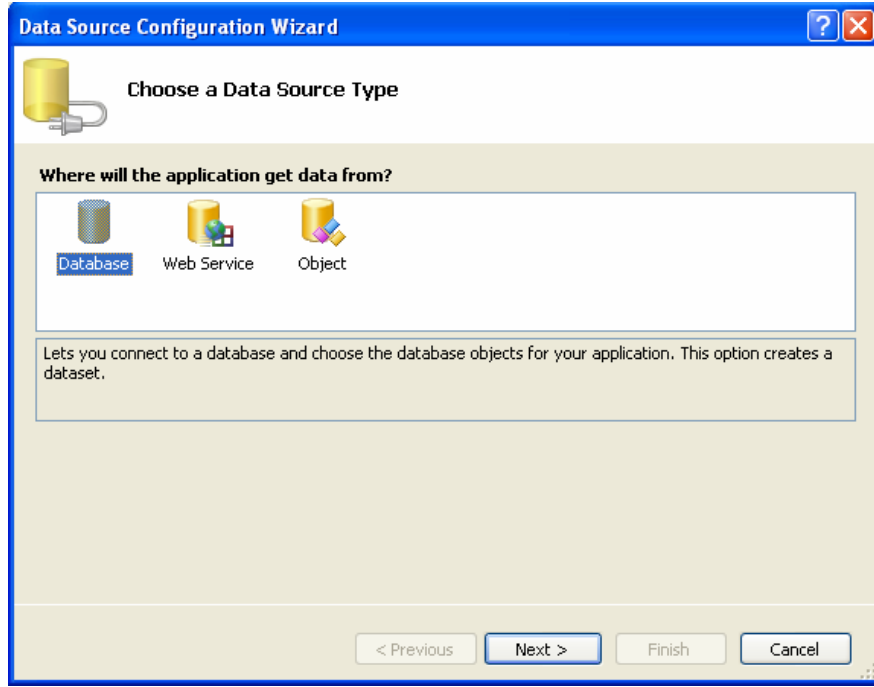
Generate Dataset penceresinde (Şekil.4.12) Datasete istenen isim verilip OK butonuna basıldığında yeni Dataset oluşur.

ToolBox'tan DataGrid seçilerek Form üzerine eklenir. Properties penceresinde Data Source DataSet seçilir. Forma çift tıklanarak Form Load olayına

OleDbDataAdapter1.Fill (Dataset1) yazılarak Data Adapter verilerle doldurulur. Artık veriler Datagrid üzerinde izlenir.

DataGrid üzerinde verilerle işlem yapabilmek için ikinci bir yol;

DataGridView forma eklenir. Açılan Data Grid View Task'tan Add Project Data Source tıklanarak Data Source Configuration Wizard çalıştırılır. Database seçilerek Next tıklanıldığında Data Adapter Wizard da olduğu gibi Data Connection penceresi gelir. Add Connection veya önceden var olan bir connection seçilerek tekrar Next tıklanır. Veri tabanında kullanılacak nesnelere kısmında Tables (Tablolar) seçilerek Finish butonuyla işlem tamamlanır.



Şekil 4.13

4.5. “Command Text” Komutu

Command sınıfı sunucuya gönderilecek deyimleri barındırır. CommandText komutu SQL ifadesini, prosedür adını veya tablo adını alır.

4.6.”SelectCommand, Update Command, DeleteCommand, InsertCommand” Komutları

4.6.1. SelectCommand

Data Source (veri kaynağı) verileri okumak için kullanılır.

```
Dim DS As DataSet
Dim adapter As OleDbDataAdapter = New OleDbDataAdapter()
Dim command As OleDbCommand
Dim connection As New
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=c:\aa.mdb")
command = New OleDbCommand("SELECT * FROM tablo ", connection)
adapter.SelectCommand = command
DS = New DataSet
adapter.Fill(DS, "tablo")
DataGrid1.DataSource = DS.Tables("tablo").DefaultView
```


4.6.2. UpdateCommand

Data Source (veri kaynağı) verileri güncellemek için kullanılır.

4.6.3. DeleteCommand

Data Source (veri kaynağı) verileri silmek için kullanılır.

4.6.4. InsertCommand

Data Source (veri kaynağı) verileri okumak için kullanılır.

4.7. @ Karakteri ile Başlayan Parametreler

Elbetteki, DbConnection nesnesi oluşturulurken yaşanan problemin benzeri burada da söz konusudur. Bu kez Command nesnelerinin parametre isimlendirmeleri bir veri sağlayıcıdan ötekine farklılık göstermektedir. Örneğin SqlCommand nesnesinde kullanılan parametreler @ ile başlarken, OleDbCommand nesnelerinde sorgu cümleciğindeki parametreler ? ile tanımlanmak zorundadır.

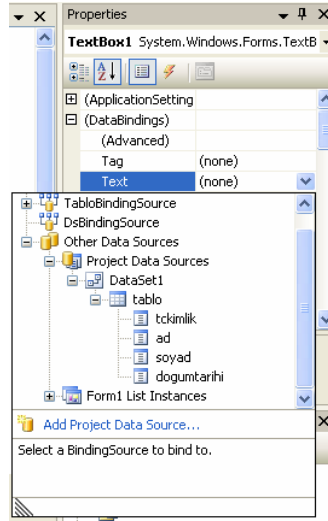
4.8. Nesnelerin DataBindings Özelliği

Verileri DataSet'e aktardıktan sonra Form üzerindeki kontrollerde bu verileri görüntülemek için iki temel yöntem vardır. Simple Data Binding yönteminde yerel veri kaynağındaki bir alanı, bir nesnenin istenilen özelliğine bağlar. Yerel veri kaynağındaki alanlardan birini kontrolün istenilen özelliğine bağlamak için kontrolün DataBindings koleksiyonunu kullanırız.

```
TextBox1.DataBindings.Add("Text", ds, "tablo.ad")
```

TextBox1 nesnesinin Text özelliğine "ds" DataSetinin "tablo" tablosunun "ad" alanını ekledik.

Aynı işlemi TextBox1 nesnesinin properties penceresinden DataBindings, Text ve veri kaynağımızdan istediğimiz alanı seçerek ekleyebiliriz.



4.9. “BindingManagerBase” Nesnesi, “Count ve Position” Özellikleri

4.9.1. BindingManagerBase

Windows formu üzerindeki veri sıçrama kontrollerinin, veri kaynağı sıçramalarına senkronizasyonunu sağlayan Binding neslerini yönetir. Başka bir deyişle form üzerinde butonlarla veri kaynağımız üzerinde gezinmemizi sağlar.

```
Dim bmb As BindingManagerBase
bmb = Me.BindingContext(DataSet adı, "Tablo adı")
```

şeklinde tanımlanır.

4.9.2.Count

Kayıt sayısını geri döndürür.

4.9.3. Position

0'dan başlamak üzere kayıt numarasını ayarlar veya geri döndürür.

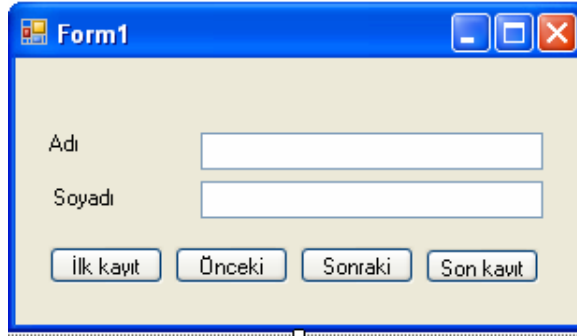
bmb.Position = 0	→ İlk kayda git
bmb.Position = n	→ n numaralı kayda git
bmb.Position += 1	→ sonraki kayda git
bmb.Position -= 1	→ önceki kayda git
bmb.Position = bmb.Count-1	→son kayda git

Yukarıdaki yazımlarda $x+=1 \rightarrow x=x+1$, $x-=1 \rightarrow x=x-1$ dir.

bmb.Count-1 ise kayıt numaraları 0'dan başladığı için son kayıt numarası kayıt sayısından bir eksiktir.

Örnek

Veri tabanımızdan ad ve soyad bilgilerini TextBoxlarda görüntüleyerek Butonlarla gezinti yapacağız. Bunun için aşağıdaki formu oluşturarak veri tabanı bağlantısını (DataAdapter, DataSet, DataConnection) oluşturunuz.



Form Load olayına

```
OleDbDataAdapter1.Fill(ds)
    TextBox1.DataBindings.Add("Text", ds, "tablo.ad")
    TextBox2.DataBindings.Add("Text", ds, "tablo.soyad")
```

İlk kayıt butonu klik olayına

```
Dim bmb As BindingManagerBase = Me.BindingContext(ds, "tablo")
    bmb.Position = 0
```

Önceki klik olayına

```
Dim bmb As BindingManagerBase = Me.BindingContext(ds, "tablo")
    bmb.Position -= 1
```

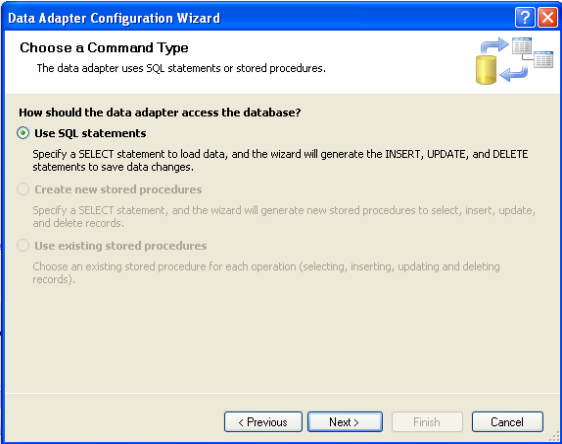
Sonraki klik olayına

```
Dim bmb As BindingManagerBase = Me.BindingContext(ds, "tablo")
    bmb.Position += 1
```

Son kayıt klik olayına

```
Dim bmb As BindingManagerBase = Me.BindingContext(ds, "tablo")
    bmb.Position = bmb.Count - 1
```

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>➤ “Data Adapter Configuration Wizard” çalıştırarak hızlı bir şekilde bir veri tabanına bağlantı yapınız.</p>  <p>➤ DataSet nesnesi ile tablodan verileri alınız (Data*Generate Dataset)</p> <p>➤ DataAdapter’ın “Fill” metodu ile verileri doldurunuz.</p> <p>➤ DataGrid kontrolüne tablo bilgilerini bağlayınız.</p> <p>➤ Metin kutuları, açılır kutu ve düğmeler ile veri tabanı içinde işlemler yapınız.</p>	<p>➤ Data Adapter Configuration Wizard sonunda oluşan öğeleri inceleyiniz.</p> <p>➤ DataSet’i kodla veya Adapter üzerine sağ tıklayarak oluşturabilirsiniz.</p> <p>➤ Fill metodunu Form Load olayına yazabilirsiniz.</p> <p>➤ DataGridView kontrolüne tablo bilgilerini kodla, properties penceresinden ve DataGridView Task penceresinden bağlayabiliriz.</p>

ÖLÇME VE DEĞERLENDİRME

A. OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki sorulardan; sonunda parantez olanlar doğru / yanlış sorularıdır. Verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. DataSet veri tabanı ile sürekli bağlantı halindedir. ()
2. Data Adapter verilerin DataSete aktarılmasını sağlar. ()
3. DataGrid veritabanındaki bilgilerin listelenebildiği bir ızgaradır. ()
4. CommandText SQL komutlarının işletildiği komuttur. ()
5. Dim bmb As BindingManagerBase = Me.BindingContext(ds, "tablo")
bmb.Position = bmb.Count - 1
komutu son kayıttan bir önceki kayda gider. ()

DEĞERLENDİRME

Sorulara verdiğiniz yanıtları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığı sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

MODÜL DEĞERLENDİRME

Modülün Adı	Görsel Programlamada Veri Tabanı	Öğrencinin Adı...:		
Amaç	Gerekli ortam sağlandığında, metin, XML ve ADO.NET kullanarak programınıza veritabanı ekleyebileceksiniz.	Soyadı:		
		Sınıfı :		
		Nu....:		
<p>AÇIKLAMA: Aşağıda listelenen davranışların her birini öğrencide gözlemleyemediyse (0), zayıf nitelikli gözlemlediyseniz (1), orta düzeyde gözlemlediyseniz (2) ve iyi nitelikte gözlemlediyseniz (3) rakamının altındaki ilgili kutucuğa X işareti koyunuz.</p>					
GÖZLENECEK DAVRANIŞLAR		0 (kötü)	1 (zayıf)	2 (orta)	3 (iyi)
METİN VE İKİLİK DOSYALAR					
1. Yapı – structure					
2. Alan – field kavramı					
3. “System.IO” sınıfı					
4. Dosya, klasör, yol (path) deyimleri					
5. “CreateDirectory, Copy, Move, GetAttributes, SetAttributes, Delete ve Exists” metotları					
6. Metin dosyalar ve ikilik dosyalar (text ve binary)					
7. Satır sonu bitirme karakteri					
8. Input ve output stream deyimleri					
9. “FileStream” sınıfı					
10. “FileAccess, FileMode, FileShare” listeleri					
11. “DirectoryNotFound, FileNotFound, ve EndOfStream” komutları					
12. “EOF, LOF, FileCopy, FileClose, FileLen, FileGet, FileOpen, Kill, Print, Seek ve Write” komutları					
XML DOSYALAR					
1. XML (Extensible Markup Language) yapısal metin dosyası					
2. HTML, web sayfası, UTF-8					
3. “Tag, attribute, comment ve element”					
4. “System.XML” sınıfı					
5. Kök, ana ve çocuk elemanlar					
6. “XML Designer” penceresi					
7. “XmlTextWriter, XmlTextReader” sınıfı ve “Formatting” özelliği					
8. “WriteStartDocument, WriteComment, WriteAttributeString, WriteEndElement” komutları					
ADO.NET (ACTIVE X DATA OBJECTS) VE OLEDB					
1. Data - veri, grid – ızgara nesneleri					
2. ADO.NET ve SQL (Structured Query Language)					

3. "Select, Insert, Update, Delete" SQL komutları				
4. İlişkisel veritabanı, tablo, indeks				
5. Microsoft SQL Server 2000 Desktop Engine (MSDE), DBMS				
6. LAN, WAN, internet, intranet deyimleri				
7. "Dataset, Data Adapter, Data Command, Data Connection" nesneleri				
VERİ BAĞLI NESNELER				
1. "Database Adapter Configuration Wizard"				
2. "Connection" kavramı				
3. "Data Link Properties – Veri Bağlantısı Özellikleri" penceresi				
4. "Query Builder" penceresi				
5. "Dataset, DataGrid" nesnesi				
6. "CommandText" komutu				
7. "SelectCommand, UpdateCommand, DeleteCommand, InsertCommand" komutları				
8. @ karakteri ile başlayan parametreler				
9. Nesnelerin "DataBindings" özelliği				
10. "BindingManagerBase" nesnesi, "Count ve Position" özellikleri				
TOPLAM PUAN				

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır. Öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	Dosya
2	Klasör
3	Alan
4	Yanlış
5	Doğru
6	Yanlış
7	Append Create CreateNew OpenOrCreate

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Yanlış
4	Doğru
5	Yanlış
6	Doğru

ÖĞRENME FAALİYETİ-3 CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Doğru
4	HASAN
5	AHMET

ÖĞRENME FAALİYETİ-4 CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Doğru
4	Doğru
5	Yanlış

ÖNERİLEN KAYNAKLAR

- DAVIS Harold, **Elektrik Visual Basic.NET Programlama Kılavuzu**.Sybex, Alfa, İstanbul, 2002.
- PALA Zeydin, **Visual Basic.Net**, Türkmen Kitabevi, İstanbul, 2003.

KAYNAKÇA

- www.godoro.com
- www.classdotnet.com
- www.zvon.org
- <http://ironbark.bendigo.latrobe.edu.au>
- <http://msdn2.microsoft.com/en-us/default.aspx>