



## Hedefler

Bu üniteyi çalıştıktan sonra;



Sequential (Sıralı) Erişimli Dosyaları öğrenmek,



Random (Rasgele) Erişimli Dosyaları öğrenmek,

### *İçindekiler*

- Dosya İşlemleri,
- Sequential (Sıralı) Erişimli Dosyalar,
- Random (Rasgele) Erişimli Dosyalar.

## DOSYA İŞLEMLERİ

Bilgisayarda girilen bilgileri kaybolmayacak ve gerektiğinde tekrar kullanılabilir şekilde disket veya harddiske kaydetmek için dosyalardan yararlanılır. Sınırlı sayıdaki bilgilerin disk ortamında saklanması, bu bilgilere erişilmesi, üzerinde değişiklik yapılması gibi işlemler için veri tabanı oluşturmak yerine dosyalama işlemleri kullanılabilir. Özellikle son yıllarda Access ve SQL Server veritabanı programlarının yoğun şekilde kullanımının yaygınlaşmasıyla, Visual Basic' te dosya uygulamalarının alanı oldukça kısıtlandı. Özellikle Visual Basic.NET' te ADO.NET' in (İlerleyen bölümlerde anlatılacaktır) gelişmiş özellikleri, programcıları hızla bu yöne çekmektedir.

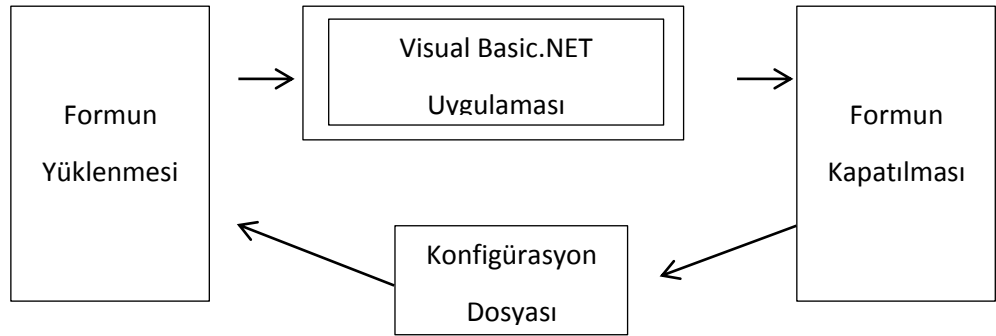
Yine de özellikle konfigürasyon dosyalarında, bilgilerin Excel, Access gibi programlara aktarılmasında veya kısıtlı sayıda kayıttan oluşan basit dosya uygulamalarında Visual Basic' te hala dosyalar kullanılmaktadır.

Visual Basic'de dosyalar, **Sequential** (Sıralı), **Random** (Rasgele) ve **Binary** (ikili) dosyalar olmak üzere üçe ayrılır. Visual Basic 6' dan kalma bir sınıf olmasına rağmen bu sınıflar .NET' te de kullanılmaktadır. Bu bölümde daha çok kullanılan Sıralı ve Rasgele erişimli dosyaları incelenecektir.

### SEQUENTIAL (SIRALI) ERİŞİMLİ DOSYALAR

Sıralı erişimli dosyalarda her bir kayıt, disk üzerine birbiri ardına kaydedilmektedir. İstenilen herhangi bir kayda ulaşmak için ondan önceki kayıtlar okunması gereklidir. Bu dosyalar özellikle konfigürasyon dosyalarının oluşturulması ve okunması için yaygın olarak kullanılmaktadır. Tüm Windows uygulamaları konfigürasyon dosyalarını kullanır. Örneğin Microsoft Word, sayfa ayarlarını, font ayarlarını, hangi araç çubuklarının görüntüleneceğini ve benzeri bilgileri konfigürasyon

dosyasında (genellikle uzantısı **.ini** dir) saklar. Bunu ayar bilgilerini Word programından çıkarken, ilgili konfigürasyon dosyasına yazarak sağlar. Daha sonra Word programı çalıştırıldığında önce bu dosya okunur ve buradaki bilgilere göre Word ekrana gelir. Benzer şekilde biz de kendi geliştireceğimiz uygulamada, hangi menü seçeneklerinin seçilmiş olduğu, seçilmiş renkler, yazı tipleri gibi bilgileri konfigürasyon dosyasına kaydederiz. Program çalıştığında ilk önce bu dosya okunarak, programımızın ön ayarları yapılır.



Birçok Windows uygulaması dosyaları Export etme özelliğine sahiptir. Örneğin Microsoft Excel'in bir tablosu bir harici dosyaya kaydedilebilir. Genellikle kullanılan yöntem, değişkenlerin virgüllerle ayrıldığı bir sıralı dosya (Comma Separated Variables - **CSV** ASCII dosya) dır. Visual Basic NET, başka bir Windows uygulamasından gelen bu dosyayı okuyup (Sıralı Dosya erişimini kullanarak), bu değişkenlerle (bilgilerle) gerekli işlemleri yapabilir.

Sıralı Dosya erişimiyle çok kullanılan bir başka uygulama da) özellikle Text Editöründe yazılan bilgilerin kaydedilmesi ve okutulmasıdır.

### Sıralı Erişimli Dosyalarda Kullanılan Fonksiyonlar

#### FileOpen (Dosya Açma)

Bu komut, dosyayı açmak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**FileOpen (Dosya No, Dosya Adı, Dosya Açma Modu, Dosya Erişim Biçimi)**

**Dosya No** : Açılacak dosyanın numarasını belirleyen bir tamsayıdır.

**Dosya Adı** : Açılacak dosyanın diskteki adı.

**Dosya Açma Modu** : Bu mod, dosyanın hangi amaç için açıldığını belirler.

**Dosya Açma Modu** için **OpenMode** metodu kullanılır. 5 farklı mod vardır.

**OpenMode.Output** modu, dosyayı ilk defa yazma amacıyla oluşturmak için kullanılır. Daha önceden bu dosya açılmış ve içinde kayıt varsa, bu modla

açıldığı zaman içindekiler silinecektir. **OpenMode.Append** modu, dosyaya kayıt eklemek, düzenlemek için kullanılır. **OpenMode.Input** modu ise

dosyadan kayıt okumak için kullanılır. **OpenMode.Binary** modu, dosyanın binary erişim modunda açılması için kullanılır. **OpenMode.Random**

dosyanın rastgele erişimli modda açılması için kullanılır.

**Dosya Erişim Biçimi :** *OpenAccess* metodu kullanılır. 4 farklı değer alabilir. *OpenAccess.Default*, varsayılan moddur. Dosyanın okuma ve yazma modunda açılması için kullanılır. *OpenAccess.Read*, dosyanın okuma modunda açılmasını sağlar. *OpenAccess.Write*, dosyanın yazma modunda açılmasını sağlar. *OpenAccess.ReadWrite*, dosyanın okuma ve yazma modunda açılmasını sağlar.

**Örnek :** `FileOpen (1, "C:\STOK\STOK.DAT", OpenMode.Input)`

**FileClose (Dosya Kapama) :**

Bu komut, açık olan dosyayı kapamak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**FileClose (Dosya No)**

**Write / WriteLine (Dosyaya Yazdırma) :**

Bu komut, açık olan dosyaya bilgiyi yazdırmak için kullanılır. Genellikle **Input** fonksiyonu ile dosyadan okunan bilgiler, **Write** fonksiyonu ile yazdırılırlar. Deyimin genel ifadesi aşağıdaki şekildedir:

**Write / WriteLine (Dosya No, Değişkenler\_listesi)**

**Dosya No :** Açılmış dosyanın numarasını belirleyen bir tamsayıdır. Dosyanın **Output** veya **Append** modunda açılmış olması gerekir.

**Write** fonksiyonu, değişkenlerin değerlerini dosyaya (,) ile ayırarak yazar. **WriteLine** ise aynı şekilde yazdıktan sonra, bir sonraki satıra (satır sonunda Enter'a basılmış gibi) geçer. **Boolean** değerler **True** veya **False** şeklinde yazılır. **Date** ifadeleri ise #2002-08-24# şeklinde yazılır.

**Örnek :** `Dim A As Integer`

`Dim B As Boolean`

`Dim C As String`

`.....`

`FileOpen (1, "c:\stok\stok.dat", OpenMode.Output)`

`Write (1, A, B, C)`

`FileClose(1)`

**Print / PrintLine (Dosyaya Yazdırma) :**

Bu komut, aynı **Write** gibi açık olan dosyaya bilgiyi yazdırmak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**Print / PrintLine (Dosya No, Değişkenler\_listesi)**

**Dosya No :** Açılmış dosyanın numarasını belirleyen bir tamsayıdır. Dosyanın **Output** veya **Append** modunda açılmış olması gerekir.

**Print** fonksiyonu, değişkenlerin değerlerini dosyaya boşluk bırakarak yazar. **PrintLine** ise aynı şekilde yazdıktan sonra, bir sonraki satıra (satır sonunda Enter'a basılmış gibi) geçer.

**Input (Dosyadan Okuma) :**

Bu komut, açık olan dosyadan bilgileri okumak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**Input (Dosya No, Değişken)**

Her **Input** fonksiyonunda dosyadan yalnızca bir değişkenin bilgisi okunur. 50 değişken varsa, 50 tane **Input** deyimi kullanılmalıdır. Dosyadan değişkenler okunurken, yazıldığı sırada işlem gerçekleştirilmelidir.

**Örnek :** Dim A As Integer

Dim B As Boolean

Dim C As String

.....

FileOpen (1, "c:\stok\stok.dat", OpenMode.Input)

Input(1, A)

Input(1,B)

Input(1,C)

FileClose(1)

**LineInput (Dosyadan Satır Okuma) :**

Bu komut, açık olan dosyadan bilgileri satır satır okumak için kullanılır. Genellikle **Print** deyimiyle yazılmış bilgileri okumak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**LineInput (Dosya No)**

**Dosya No :** Açılmış dosyanın numarasını belirleyen bir tamsayıdır. Dosyanın **Input** modunda açılmış olması gerekir.

Her **LineInput** fonksiyonunda dosyadan bir satırlık bilgi okunur ve bu bilgi string bir değişkene aktarılır. Dosya sonuna varıncaya kadar (bu **EOF** ile kontrol edilir) satır okuma işlemi devam eder.

**Örnek : Dim Satir As String**

.....

**FileOpen (1, "c:\stok\stok.dat", OpenMode.Input)**

**While Not Eof(1)**

**Satir = LineInput(1)**

....

**End While**

**FileClose(1)**

**EOF (End of File – Dosya Sonu) :**

Bu fonksiyon ile dosya sonuna ulaşıp ulaşılmadığı kontrol edilir. Deyimin genel ifadesi aşağıdaki şekildedir:

**EOF (Dosya No)**

**Dosya No :** Açılmış dosyanın numarasını belirleyen bir tamsayıdır.

**KILL Deyimi :**

Dosya silmek için kullanılır.

**KILL ("Dosya Adı")**

**RENAME Deyimi :**

Dosya adını değiştirmek için kullanılan bir komuttur.

**RENAME ("Eski Dosya Adı", "Yeni Dosya Adı")**

**RANDOM (RASGELE) ERİŞİMLİ DOSYALAR**

Bu dosyalarda her kaydın bir kayıt numarası bulunur. Bu kayıt numarası kullanılarak, istenilen kayda doğrudan doğruya ulaşılabilir. Sıralı dosyalarda olduğu gibi, herhangi bir kayda ulaşmak için kendinden önceki tüm kayıtları okumaya gerek yoktur.

Rasgele erişimli dosyalarda, hem okuma hem de yazma işlemi birlikte yapılabilmektedir.

Rasgele erişim li dosyalarda sabit uzunluklu kayıtlar kullanılır. Bir kaydı oluşturan bilgiler bir kayıt ismi altında, **STRUCTURE** tanımlama bloğunun altında tanımlanır. Bu deyimın genel yazılış ifadesi ;

Structure Kayıt İsmi

Değişken1 AS Değişken Tipi 1

Değişken2 AS Değişken Tipi 2

Değişken3 AS Değişken Tipi 3

Değişken4 AS Değişken Tipi 4

.....

End Structure



Burada Değişken1, Değişken2, ... kullanılacak değişkenlerin ismidir. Değişken tipi 1, Değişken tipi 2, ... değişkenlerin tipini göstermektedir. **String** veri tiplerini sabit uzunlukta tanımlamak için **VbfixedString** ifadesinden yararlanılır.

```
Structure Memur
    <VbfixedString (25)» Public AdSoy As String
    <VbfixedString (20)» Public Gorev As String
    Public Cins As Boolean
    Public DogTar As Date
    Public HizYil As Byte
    Public Medeni As Boolean
    Public Cocuk As Byte
    Public Boy As Single
    Public Ucret As Long
    Public Kontrol As Boolean
End Structure
```

**Structure** ile tanımlanan kaydın bir değişkene **Dim** ile atanma işlemi gerçekleştirilir.

**Örnek :** Dim Per As Memur

Artık program içinde tüm işlemlerde **Per** kayıt değişkeni kullanılacaktır. Kayıt içindeki herhangi bir değişken kayıt değişkeni ile birlikte tanımlanır (Örneğin Per.Ucret).

**FileOpen (Dosya Açma)**

Bu komut, dosyayı açmak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**FileOpen (Dosya No, Dosya Adı, Dosya Açma Modu, , ,Kayıt Uzunluğu)**

**Dosya No** : Açılacak dosyanın numarasını belirleyen bir tamsayıdır.

**Dosya Adı** : Açılacak dosyanın diskteki adı.

**Dosya Açma Modu** : Bu mod, dosyanın hangi amaç için açıldığını belirler. **Random** dosyalarda hem okuma hem de yazma için **OpenMode.Random** kullanılır.

**Kayıt Uzunluğu** : Dosyadaki bir kaydın uzunluğu yazılmalıdır. **LEN** fonksiyonundan yararlanarak kayıt uzunluğu bulunabilir.

**Örnek** : Dim Uzunluk As Long

.....

Uzunluk = Len(Memur)

.....

FileOpen (1, "C:\personel\personel.dat", OpenMode.Random, ,  
,Uzunluk)

.....

**FileClose (Dosya Kapama) :**

Bu komut, açık olan dosyayı kapamak için kullanılır. Deyimin genel ifadesi aşağıdaki şekildedir:

**FileClose (Dosya No)**

**Dosya No** : Kapanacak dosyanın numarasını belirleyen bir tamsayıdır. Açılan dosyanın numarası ile aynı olmak zorundadır.

**Örnek** : FileClose (1)

**FilePut (Dosyaya Yazdırma) :**

Random dosyaya bilgi kaydetmek için kullanılır. Genel ifadesi ;

**FilePut (Dosya No, Kayıt Değişkeni, Kayıt No)**

**FileGet (Dosyadan Okuma) :**

Random dosyadan bilgi okumak için kullanılır. Genel ifadesi ;

**FileGet (Dosya No, Kayıt Değişkeni, Kayıt No)**

**LOF (Dosya Uzunluğu) :**

Random dosyalarda, dosyanın toplam karakter sayısını bulmak için kullanılır. Genel ifadesi :

**LOF (Dosya No)**

**Seek (Bulunan Kayıt Pozisyonu) :**

Random dosyalarda, kayıt göstergesinin hangi kayıta bulunduğunu göstermek veya istenilen kayıt pozisyonuna gitmek için kullanılır. Genel ifadesi :

**Seek (Dosya No) '**  Hangi kayıta olduğunu belirler.

**Seek (Dosya No, Kayıt No) '**  Okuyucu/Yazıcı pozisyonu istenilen kayıta getirir.

**Loc (Bulunulan Kayıt Pozisyonu) :**

Random dosyalarda, kayıt göstergesinin hangi kayıta bulunduğunu gösterir. Genel ifadesi :

**Loc (Dosya No)** 'Hangi kayıta olduğunu belirler.

**EOF (End of File – Dosya Sonu) :**

Bu fonksiyon ile dosya sonuna ulaşıp ulaşılmadığı kontrol edilir. Deyimin genel ifadesi aşağıdaki şekildedir:

**EOF (Dosya No)**