

## Sequential-Access ve Random-Access dosya işlemleri için örnek programlar

Tüm Kayıtlar				
4	Fatma	Çetin	345	567
8	Kaan	Sevecen	456	897

```
using System;
// Class Library projesi içinde ÖğrenciKayit ve ÖğrenciKayitRandom sınıfları
namespace ÖğrenciLibrary
{
    // Sıralı erişimli dosya kayıt sınıfı
    [Serializable]
    public class ÖğrenciKayit
    {
        public string numara;
        public string ad;
        public string soyad;
        public string telefon;
        public int kredi;

        // default constructor
        public ÖğrenciKayit() : this( "", "", "", "", 0 )
        {
        }
        // overload constructor
        public ÖğrenciKayit(string numara, string ad, string soyad, string telefon, int kredi)
        {
            this.numara = numara;
            this.ad = ad;
            this.soyad = soyad;
            this.telefon = telefon;
            this.kredi = kredi;
        } // end constructor
    } // ÖğrenciKayit class sonu

    // Rastgele erişimli dosya kayıt sınıfı
    public class ÖğrenciKayitRandom
    {
        private int sıra;
        private char[ ] numara = new char[10];
        private char[ ] ad = new char[15];
        private char[ ] soyad = new char[15];
    }
}
```

```

private char[ ] telefon = new char[15];
private int kredi;

public const int KAYITBOYUTU = 4 + 2 * ( 10 + 15 + 15 + 15 ) + 4; // 118 byte

// default constructor
public OgrenciKayitRandom() : this( 0, null, null, null, null, 0 )
{
}

// overload constructor
public OgrenciKayitRandom( int sira, char[ ] numara, char[ ] ad, char[ ] soyad,
char[ ] telefon, int kredi)
{
    this.sira = sira;
    this.numara = numara;
    this.ad = ad;
    this.soyad = soyad;
    this.telefon = telefon;
    this.kredi = kredi;
} // end constructor

public int Sira
{
    get
    {
        return sira;
    }
    set
    {
        sira = value;
    }
}

```

```

public string Numara
{
    get
    {
        return new string(numara);
    }
    set
    {
        string Str = value;
        if (value.Length >= 10)
            Str = value.Substring( 0, 10 );
        else
            Str = value + new string(' ', 10 - value.Length);
        numara = Str.ToCharArray();
    }
}

public string Ad
{
    get
    {
        return new string(ad);
    }
    set
    {
        string Str = value;
        if (value.Length >= 15)
            Str = value.Substring( 0, 15 );
        else
            Str = value + new string(' ', 15 - value.Length);
        ad = Str.ToCharArray();
    }
}

```

```

public string Soyad
{
    get
    {
        return new string(soyad);
    }
    set
    {
        string Str = value;
        if (value.Length >= 15)
            Str = value.Substring( 0, 15 );
        else
            Str = value + new string(' ', 15 - value.Length);
        soyad = Str.ToCharArray();
    }
}

public string Telefon
{
    get
    {
        return new string(telefon);
    }
    set
    {
        string Str = value;
        if (value.Length >= 15)
            Str = value.Substring( 0, 15 );
        else
            Str = value + new string(' ', 15 - value.Length);
        telefon = Str.ToCharArray();
    }
}

```

```

public int Kredi
{
    get
    {
        return kredi;
    }
    set
    {
        kredi = value;
    }
} // Property sonu
} // OgrenciKayitRandom class sonu
} // OgrenciLibrary namespace sonu

```



## Sequential-Access Dosya

MinimizeBox = false, MaksimizeBox = false

The screenshot shows a Windows application window titled "Öğrenci İşlemleri". The window has a title bar with standard Windows controls (minimize, maximize, close). The main content area is divided into several sections:

- Form Fields:** A series of labels and text boxes for entering student data: Numara (1), Ad (Kemal), Soyad (AYDIN), Tel (333 33 33), Kredi (1000). There are also buttons for "Programdan Çıkış" and "Tümünü Oku".
- File Operations:** A section with buttons for "Yeni Dosya Oluştur", "Dosya Aç", "Yeni Kayıt", "Sonraki Kaydı Oku >>", "Dosya Kapat", and "Tümünü Oku".
- List Box:** A list box titled "Tüm Kayıtlar" containing a table of student records.

Tüm Kayıtlar					
4	Fatma	Çetin	345	567	
8	Kaan	Sevecen	456	897	

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;

using ÖğrenciLibrary;

namespace ÖğrenciIslemleri
{
    /// <summary>
    /// Summary description for Form1.
    /// </summary>
    public class FormIslemler : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.TextBox tBSoyad;
        private System.Windows.Forms.TextBox tBAd;
        private System.Windows.Forms.TextBox tBNumara;
        private System.Windows.Forms.TextBox tBKredi;
        private System.Windows.Forms.TextBox tBTel;
        private System.Windows.Forms.ListBox lBTum;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Button bYeniDosya;
        private System.Windows.Forms.Button bYeniKayit;
        private System.Windows.Forms.Button bDosyaKapat;
    }
}
```

```

private System.Windows.Forms.Button bSonrakiKayit;
private System.Windows.Forms.Button bDosyaAc;
private System.Windows.Forms.Button bCikis;
private System.Windows.Forms.Button bTumu;
private System.ComponentModel.Container components = null;
/// <summary>
/// Required designer variable.
/// </summary>

// Genel tanımlamalar
private FileStream kaynakDosya;
string dosyaAdi;
OgrenciKayit ogrenciKayit = new OgrenciKayit();
DialogResult sonuc;

// Sequential-Access dosya için
private BinaryFormatter formatlayici = new BinaryFormatter();

public FormIslemler()
{
    //
    // Required for Windows Form Designer support
    //
    InitializeComponent();

    //
    // TODO: Add any constructor code after InitializeComponent call
    //
}

/// <summary>
/// Clean up any resources being used.
/// </summary>

```

```

protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
        base.Dispose( disposing );
    }
}

// Visual Studio .NET generated code

[STAThread]
static void Main()
{
    Application.Run(new FormIslemler());
}

private void bYeniDosya_Click(object sender, System.EventArgs e)
{
    SaveFileDialog dosyaSecici = new SaveFileDialog();
    sonuc = dosyaSecici.ShowDialog();

    if (sonuc != DialogResult.Cancel)
    {
        dosyaAdi = dosyaSecici.FileName;
        kaynakDosya = new FileStream( dosyaAdi,
            FileMode.OpenOrCreate, FileAccess.Write );
    }
}

```

```

private void bDosyaAc_Click(object sender, System.EventArgs e)
{
    OpenFileDialog dosyaSecici = new OpenFileDialog();
    sonuc = dosyaSecici.ShowDialog();

    if (sonuc != DialogResult.Cancel)
    {
        dosyaAdi = dosyaSecici.FileName;
        if (kaynakDosya != null) kaynakDosya.Close();

        kaynakDosya = new FileStream( dosyaAdi,
            FileMode.Open, FileAccess.ReadWrite );

        foreach (Control cl in Controls)
            if (cl is TextBox) cl.Text = "";

        bSonrakiKayit_Click(bSonrakiKayit, System.EventArgs.Empty);
    }
}

private void bYeniKayit_Click(object sender, System.EventArgs e)
{
    ogrenciKayit.numara = tBNumara.Text;
    ogrenciKayit.ad = tBAd.Text;
    ogrenciKayit.soyad = tBSoyad.Text;
    ogrenciKayit.telefon = tBTel.Text;
    ogrenciKayit.kredi = Int32.Parse(tBKredi.Text);

    formatlayici.Serialize( kaynakDosya, ogrenciKayit );
}

```

```

private void bSonrakiKayit_Click(object sender, System.EventArgs e)
{
    if (kaynakDosya == null || !kaynakDosya.CanRead)
        MessageBox.Show("Açık Dosya Yok !", "Hata",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        try
        {
            ogrenciKayit = (OgrenciKayit)formatlayici.Deserialize(kaynakDosya);
            tBNumara.Text = ogrenciKayit.numara;
            tBAd.Text = ogrenciKayit.ad;
            tBSoyad.Text = ogrenciKayit.soyad;
            tBTel.Text = ogrenciKayit.telefon;
            tBKredi.Text = ogrenciKayit.kredi.ToString();
        }
        catch (SerializationException)
        {
            MessageBox.Show("Dosya Sonu !", "Uyarı",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void bDosyaKapat_Click(object sender, System.EventArgs e)
{
    kaynakDosya.Close();
}

private void bTumu_Click(object sender, System.EventArgs e)
{
    bool dosyaSonu = false;

    if (kaynakDosya != null) kaynakDosya.Close();
    OpenFileDialog dosyaSecici = new OpenFileDialog();
    sonuc = dosyaSecici.ShowDialog();
}

```

```

if (sonuc != DialogResult.Cancel)
{
    dosyaAdi = dosyaSecici.FileName;
    kaynakDosya = new FileStream( dosyaAdi,
        FileMode.Open, FileAccess.ReadWrite );
    lBTum.Items.Clear();

    while (!dosyaSonu)
    {
        try
        {
            ogrenciKayit = (OgrenciKayit) formatlayici.Deserialize(kaynakDosya);
            lBTum.Items.Add(ogrenciKayit.numara + "\t" +
                ogrenciKayit.ad + "\t" +
                ogrenciKayit.soyad + "\t" +
                ogrenciKayit.telefon + "\t" +
                ogrenciKayit.kredi.ToString());
        }
        catch (SerializationException)
        {
            MessageBox.Show("Tüm Kayıtlar Okundu !", "Bilgi",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            dosyaSonu = true;
        }
    }
}

private void bCikis_Click(object sender, System.EventArgs e)
{
    Application.Exit();
}
}
}

```

## Random-Access Dosya

MinimizeBox = false, MaksimizeBox = false

Sıra	Numara	Ad	Soyad	Tel	Kredi	bTum
0						456
1	456	m.ali	akcayol	456	234	789
2	456	ahmet	kerim			456
3		kemal	sel			0
4						0
5						0
6						0
7						0
8						0
9						0
10						0
11						0
12						0
13						0
14						0
15						0

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using System.IO;

using OgresciLibrary;

namespace OgresciIslemleri
{
    /// <summary>
    /// Summary description for Form1.
    /// </summary>

    public class FormIslemler : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.TextBox tBSoyad;
        private System.Windows.Forms.TextBox tBAd;
        private System.Windows.Forms.TextBox tBNumara;
        private System.Windows.Forms.TextBox tBKredi;
        private System.Windows.Forms.TextBox tBTel;
        private System.Windows.Forms.ListBox lBTum;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.TextBox tBSira;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Button bCikis;
        private System.Windows.Forms.Button bSonrakiKayit;

```

```

        private System.Windows.Forms.Button bDosyaKapat;
        private System.Windows.Forms.Button bYeniKayit;
        private System.Windows.Forms.Button bTumunuOku;
        private System.Windows.Forms.Button bYeniDosya;
        private System.Windows.Forms.Button bDosyaAc;

        private System.ComponentModel.Container components = null;
        /// <summary>
        /// Required designer variable.
        /// </summary>

        // Genel tanımlamalar
        private FileStream kaynakDosya;
        string dosyaAdi;
        private int aktifKayitSira;

        OgresciKayitRandom ogresciKayitRandom = new OgresciKayitRandom();
        DialogResult sonuc;

        // Random-Access dosya için
        private BinaryWriter randomYazici;
        private BinaryReader randomOkuyucu;

        public FormIslemler()
        {
            //
            // Required for Windows Form Designer support
            //
            InitializeComponent();
            //
            // TODO: Add any constructor code after InitializeComponent call
            //
        }

```



```

/// <summary>
/// Clean up any resources being used.
/// </summary>
protected override void Dispose( bool disposing )
{
    if( disposing )
    {
        if (components != null)
        {
            components.Dispose();
        }
        base.Dispose( disposing );
    }
}

// Visual Studio .NET generated code

[STAThread]
static void Main()
{
    Application.Run(new FormIslemler());
}

private void bYeniDosya_Click(object sender, System.EventArgs e)
{
    SaveFileDialog dosyaSecici = new SaveFileDialog();
    sonuc = dosyaSecici.ShowDialog();

    if (sonuc != DialogResult.Cancel)
    {
        dosyaAdi = dosyaSecici.FileName;
        kaynakDosya = new FileStream( dosyaAdi,
            FileMode.Create, FileAccess.Write );
    }
}

```

```

// Başlangıçta 100 kayıt daha sonra artırılabilir
kaynakDosya.SetLength(100 * OgrenciKayitRandom.KAYITBOYUTU);
randomYazici = new BinaryWriter(kaynakDosya);
}
}

private void bDosyaAc_Click(object sender, System.EventArgs e)
{
    OpenFileDialog dosyaSecici = new OpenFileDialog();
    sonuc = dosyaSecici.ShowDialog();

    if (sonuc != DialogResult.Cancel)
    {
        dosyaAdi = dosyaSecici.FileName;
        if (kaynakDosya != null) kaynakDosya.Close();
        kaynakDosya = new FileStream( dosyaAdi,
            FileMode.Open, FileAccess.ReadWrite );

        randomOkuyucu = new BinaryReader( kaynakDosya );
        randomYazici = new BinaryWriter(kaynakDosya);

        foreach (Control cl in Controls)
            if (cl is TextBox) cl.Text = "";
        aktifKayitSira = 0;
        bSonrakiKayit_Click(bSonrakiKayit, System.EventArgs.Empty);
    }
}

private void bYeniKayit_Click(object sender, System.EventArgs e)
{
    ogrenciKayitRandom.Sira = Int32.Parse(tBSira.Text);
    ogrenciKayitRandom.Numara = tBNumara.Text;
    ogrenciKayitRandom.Ad = tBAd.Text;
    ogrenciKayitRandom.Soyad = tBSoyad.Text;
    ogrenciKayitRandom.Telefon = tBTel.Text;
}

```

```

ogrenciKayitRandom.Kredi = Int32.Parse(tBKredi.Text);
kaynakDosya.Position = ogrenciKayitRandom.Sira * OgrrenciKayitRandom.KAYITBOYUTU;
randomYazici.Write(ogrenciKayitRandom.Sira);
randomYazici.Write(ogrenciKayitRandom.Numara);
randomYazici.Write(ogrenciKayitRandom.Ad);
randomYazici.Write(ogrenciKayitRandom.Soyad);
randomYazici.Write(ogrenciKayitRandom.Telefon);
randomYazici.Write(ogrenciKayitRandom.Kredi);
}

private void bSonrakiKayit_Click(object sender, System.EventArgs e)
{
    if (kaynakDosya == null || !kaynakDosya.CanRead)
        MessageBox.Show("Açık Dosya Yok !", "Hata",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        try
        {
            kaynakDosya.Position = aktifKayitSira * OgrrenciKayitRandom.KAYITBOYUTU;
            aktifKayitSira++;
            tBSira.Text = randomOkuyucu.ReadInt32().ToString();
            tBNumara.Text = randomOkuyucu.ReadString();
            tBAd.Text = randomOkuyucu.ReadString();
            tBSoyad.Text = randomOkuyucu.ReadString();
            tBTel.Text = randomOkuyucu.ReadString();
            tBKredi.Text = randomOkuyucu.ReadInt32().ToString();
        }
        catch (IOException)
        {
            MessageBox.Show("Dosya Sonu !", "Uyarı",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```

private void bDosyaKapat_Click(object sender, System.EventArgs e)
{
    if (kaynakDosya != null) kaynakDosya.Close();
    if (randomYazici != null) randomYazici.Close();
    if (randomOkuyucu != null) randomOkuyucu.Close();
}

private void bTumunuOku_Click(object sender, System.EventArgs e)
{
    bool dosyaSonu = false;

    if (kaynakDosya != null) kaynakDosya.Close();
    OpenFileDialog dosyaSecici = new OpenFileDialog();
    sonuc = dosyaSecici.ShowDialog();
    if (sonuc != DialogResult.Cancel)
    {
        dosyaAdi = dosyaSecici.FileName;
        kaynakDosya = new FileStream(dosyaAdi,
            FileMode.Open, FileAccess.ReadWrite);
        randomOkuyucu = new BinaryReader(kaynakDosya);

        aktifKayitSira = 0;
        lBTum.Items.Clear();

        while (!dosyaSonu)
        {
            try
            {
                kaynakDosya.Position = aktifKayitSira * OgrrenciKayitRandom.KAYITBOYUTU;
                aktifKayitSira++;

                ogrenciKayitRandom.Sira = randomOkuyucu.ReadInt32();
                ogrenciKayitRandom.Numara = randomOkuyucu.ReadString();
                ogrenciKayitRandom.Ad = randomOkuyucu.ReadString();
                ogrenciKayitRandom.Soyad = randomOkuyucu.ReadString();
            }
            catch { }
        }
    }
}

```

```
ogrenciKayitRandom.Telefon = randomOkuyucu.ReadString();
ogrenciKayitRandom.Kredi = randomOkuyucu.ReadInt32();

lBTum.Items.Add(ogrenciKayitRandom.Sira.ToString() + "\t" +
ogrenciKayitRandom.Numara + "\t" +
ogrenciKayitRandom.Ad + "\t" +
ogrenciKayitRandom.Soyad + "\t" +
ogrenciKayitRandom.Telefon+ "\t" +
ogrenciKayitRandom.Kredi.ToString());
}
catch (IOException)
{
    MessageBox.Show("Tüm Kayıtlar Okundu !", "Bilgi",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    dosyaSonu = true;
}
}
}

private void bCikis_Click(object sender, System.EventArgs e)
{
    Application.Exit();
}
}
```