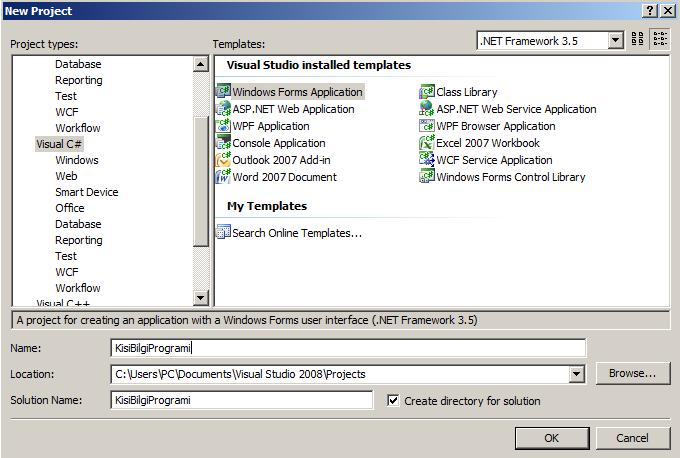
[C# ile List Kullanımı](http://www.batuhanduzgun.net/post/2010/08/12/C-ile-List-Kullanc4b1mc4b1.aspx)

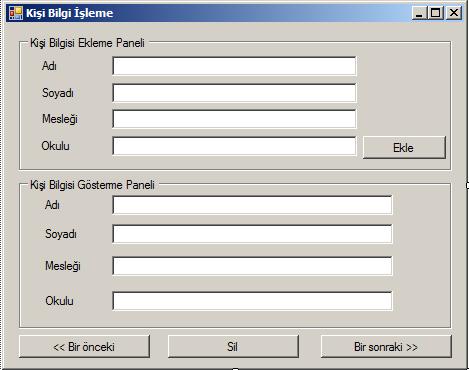
Projemiz kişiye ait bir takım bilgileri alıp,göstermeye yarayacak,ayrıca ilgili kaydı silmenizi veya yeni bir kayıt eklemenizi sağlar.Bunun için bir sınıf tasarladık,bu sınıfımız form üzerindeki textboxlardan gelen verileri bizim oluşturduğumuz sınıfımızın içindeki parametrelere atayacak.Daha sonra bilgileri yüklediğimiz sınıfa ait nesneyi List yapısına ekleyeceğiz veya sileceğiz.

Şimdi ilk iş olarak yeni bir FormApplication açalım.Aşağıdaki şekildeki gibi :



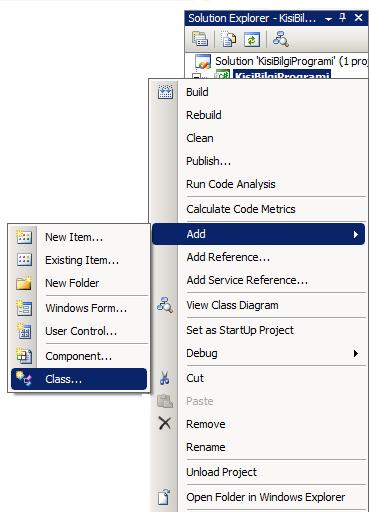
Projemizi oluşturduktan sonra artık görsel olarak formumuzu düzenlemeye sıra geldi.Form üzerine iki adet **GroupBox**,sekiz adet **TextBox** ve dört adet **Buton**yerleştirdik.

Görsel olarak aşağıdaki şekilde düzenledim.



Görsel olarak tasarımımızı yaptıktan sonra Form üzerinde son işlememiz yerleştiğimiz TextBox ve Buton nesnelerine uygun isimler vermek.Unutmayalım iyi yazılımcılar bu tür isimlendirme standartlarına uyarlar.Bu kodlama yaparken hangi görsel bileşenin ne işe yaradığını bize hatırlamada yardımcı olur.Ben sırasıyla şu şekilde verdim isimleri; TextBoxlar için : **txt\_kaydet\_adi,txt\_kaydet\_soyadi,txt\_kaydet\_Meslek ve txt\_kaydet\_Okulu** şeklindedir,Butonlar için : **btn\_Ekle,btn\_bir\_onceki,btn\_sil**ve **btn\_bir\_sonraki** şeklindedir.Sizler kendinize göre farklı isimlerde verebilirsiniz.

Evet görsel tasarım kısmını hazırladık.Artık bilgilerimizi depolayacak sınıfımızı projemize ekleyelim.Aşağıdaki şekilde bir projeye bir C# Class'ı nasıl eklenir gösterilmiştir.



Ardından gelen pencerede sizden sınıfın isminin ne olması gerektiğini soracaktır.Ben**"Kisi"** ismini verdim.Ardından açılan boş kod sayfasına kodlarımızı eklemeye başlayalım.

**"Kisi"**  adlı sınıfımızın kodlarını yazarken dört adet değişken kullandık.Bunlar private değişkenlerdi.Biz bu private değişkenlere dolaylı yoldan erişim için,**"get-set"** bloklarıyla dışarıdan erişime açtık.Bu bize uygulamayı geliştirirken kod güvenliğini arttırmayı sağladı.**"Kisi"** adlı sınıfımıza ait kodlar aşağıdaki gibidir.

class Kisi

{

        privatestring Kisi\_Adi;

        public string KISI\_ADI

        {

            get{ return Kisi\_Adi; }

            set{ Kisi\_Adi = value; }

        }

        privatestring Kisi\_Soyadi;

        public string KISI\_SOYADI

        {

            get{ return Kisi\_Soyadi; }

            set{ Kisi\_Soyadi = value; }

        }

        privatestring Kisi\_Meslegi;

        public string KISI\_MESLEGI

        {

            get{ return Kisi\_Meslegi; }

            set{ Kisi\_Meslegi = value; }

        }

        privatestring Kisi\_Okulu;

        public string KISI\_OKULU

        {

            get{ return Kisi\_Okulu; }

            set{ Kisi\_Okulu = value; }

        }

 }

 Görüldüğü üzere private olan,yani dışardan erişime kapalı olan değişkelerimizi **"get-set"**  bloklarıyla dolaylı yoldan erişime açmış oluyoruz.**"set"** bloğu ilgili değişkene değer atar.Mesela **"Kisi\_Okulu"** değişkenine **"Sakarya Üniversitesi"** bilgisini atarsam,**"set"**bloğu çalışır.Aksi durumda ben "**Kisi\_Okulu"** adlı değişkendeki veriyi görmek istersem,**"get"** bloğu çalışır.Çalışma prensibi bu şekildedir.

 Şimdi formumuza çift tıklayıp formumuza ait olan kod sayfasına gelelim.Bu kod sayfasına bir başlık dosyası eklememiz gerekiyor.Bu başlık dosyası**"using System.Collections;"** 'dır.Bu eklemeden sonra artık**List,ArrayList,Stack(Yığın),Queue(Kuyruk)** vs gibi sınıfları kullanabilme imkanı tanır.

usingSystem.Collections;

**"Kisi"** sınıfından oluşturduğumuz nesneleri saklayacak olan **List** nesnemizi oluşturalım.

Aşağıdaki kod satırında olduğu gibi;

List<Kisi> KisiBilgileriListe;

Burada dikkat ederseniz sadece belirtme yaptık,**"new"**  anahtar sözcüğüyle oluşturmadık nesnemizi.Nesneyi oluşturma işlemini **Form\_Load** metodu içine yazacağız.Böylece form yüklendiğinde otomatik olarak nesnemiz oluşturulmuş olacaktır.

private void Form1\_Load(objectsender, EventArgs e)

{

    KisiBilgileriListe = new List<Kisi>();

}

 Evet,bu aşamaya kadar geldiysek,şimdi metodlarımızı yazalım.İlk önce kayıt eklemek için gerekli olan metodumuzu yazıyoruz.Bu metod ismine **KayitEkle** adını verdim.

**KayıtEkle** metodu texboxlardan girilen verileri Kisi sınıfındaki değişkenlere doldurup,listeye eklemeye yarar. Metodumuzun kodları açıklamalarıyla birlikte aşağıdaki gibidir.

public void KayitEkle()

{

            KisiMyKisi = new Kisi();// geçici bir Kisi sınıfına ait nesne oluşturduk.

            MyKisi.KISI\_ADI =txt\_kaydet\_adi.Text.Trim();// Bu dört satırlık kodformdan gelen verileri,

            MyKisi.KISI\_SOYADI =txt\_kaydet\_soyadi.Text.Trim();//Kisi sınıfının değişkenlerineatama işlemi yapılır.

            MyKisi.KISI\_MESLEGI =txt\_kaydet\_Meslek.Text.Trim();

            MyKisi.KISI\_OKULU =txt\_kaydet\_Okulu.Text.Trim();

            KisiBilgileriListe.Add(MyKisi);//Verileri yüklediğimiz nesneyi List'imize ekleyelim.

            Kapasite =KisiBilgileriListe.Count;//Kaç tane kayıt oluştuğunuhafıza bir değişkende tutuyoruz.

            MessageBox.Show("Kaydınız eklendi!");//kayıt eklendiğine dair kullanıcaya bilgi mesajı

}

Evet,kayıt ekleme işlemini rahatlıkla yapabiliriz.Şimdi eklenen kayıtlar arasında dolaşmayı sağlayacak butonların metodlarına kodlarımızı yazalım.Aşağı kısımda üç çeşit buton bulunmakta.Bunlardan iki tanesi bir ileri,bir geri gitmeyi sağlar.Bir buton ilgili kaydın silinmesini sağlar.Şimdi aşağıdaki kod parçası bir ileri gitmeyi sağlayan kodlardır.

 private void btn\_bir\_sonraki\_Click(objectsender, EventArgs e)

{

            Bas++;//basdeğişkeni -1 den başlıyor her butona basılma olayında bas değişkeni 1 artar.

            if(Bas < Kapasite)//bas değişken değeri eğerlistedeki eleman sayısından fazla ise ekrana mesaj verilir.

            {

                KisiGeciciNesne = KisiBilgileriListe[Bas];// Kisi sınıfındangecici bir nesne oluşturulur.Listeden aldığımız nesneyi bu gecici nesneye atarız.

                txt\_goster\_adi.Text =GeciciNesne.KISI\_ADI;//gecici nesnedeki verileritextboxlara dolduruyoruz.

                txt\_goster\_soyadi.Text =GeciciNesne.KISI\_SOYADI;

                txt\_goster\_meslek.Text =GeciciNesne.KISI\_MESLEGI;

                txt\_goster\_okulu.Text =GeciciNesne.KISI\_OKULU;

            }

            else

            {

                MessageBox.Show("Kayıt sınırını aştınız!");

            }

 }

Aşağıdaki kod parçası ise bir geri gitmeye yarayan kodlardır.

private void btn\_bir\_onceki\_Click(objectsender, EventArgs e)

{

            Bas--;

            //herbutona basılma olayında Bas adlı değişkeni bir azaltıyoruz.

            if(Bas > -1)//-1 değerinden büyük olduğu sürece dizisınırını aşmamışız demektir.

            {

                KisiGeciciNesne = KisiBilgileriListe[Bas];//seçili olannesneyi gecici olarak oluşturuğumuz nesneye atıyoruz.

                txt\_goster\_adi.Text = GeciciNesne.KISI\_ADI;//nesneye ait bilgileri textboxlara dolduruyoruz.

                txt\_goster\_soyadi.Text =GeciciNesne.KISI\_SOYADI;

                txt\_goster\_meslek.Text =GeciciNesne.KISI\_MESLEGI;

                txt\_goster\_okulu.Text =GeciciNesne.KISI\_OKULU;

            }

            else

            {

                MessageBox.Show("Kayıt sınırını aştınız!");

            }

}

Ve son olarakda silme işlemini  gerçekleştiren metodumuz.

private void btn\_sil\_Click(objectsender, EventArgs e)

{

            KisiBilgileriListe.RemoveAt(Bas);

            Kapasite--;

            Bas--;

}