\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*Class Açıklamaları\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-Kullanıcı arayüzünün tanımlancağı kısımdır.

--------------------------------------------

- İlk olarak Win kısmının tasarlanmasından başlanarak adım adım gidilecektir.

-----------------------------------------

- İki adet base form tanımımız olacak. Diğer formlar bu iki base formdan türeyecek.

1- Veri girişi yapmış olduğumuz formlar için bir tane base form

2- Kartlar olarak isimlendirilen, listelemeleri yapmış olduğumuz ve kullanıcıya göstermiş olduğumuz

base form

BaseKartForm.cs -> Veri girişi olarak kullanacağımız formlar için

BaseKartlarForm.cs -> Listeleme yapacağımız formlar için

-------------------------------------------

-> "BaseKartForm" da Çıkış isimli buton "ShourtcutKeyDisplayString" kısmına "Esc" yazılır. "Esc"

normalde "ItemShourtKey" kısmına kısayol tuşu olarak atanamaz. Daha sonra bu olayı

keydown olaylarında tanımlayacağız.

-------------------------------------------

-> Açmış olduğumuz UserControls/Controls klasörünün içersine kullanacağımız kontrollerin kendimize

özel ayarlar içerecek şekilde class' lar hazırlamak için kullanılır. ButtonEdit, TextEdit vb kontroller

için denetim sağlamamızı sağlar.

--------------------------------------------

-> [ToolboxItem(true)] attribute başlığı ile oluşturduğumuz custım tool'u Toolbox kısmına taşırız.

public event EventHandler<IdChangedEventArgs> IdChanged = delegate { }; satırı ile IdChanged event'ının boş değer

döndürmesini engellemiş oluruz. Böylelikle sistem hatalarından kurtulmuş oluruz. Bu işlemi delegate{} işlemi yaparız.

---------------------------------------------

Navigator, veriler arasında dolaşmamızı sağlayan araç çubuğudur.

Üç adet navigator yapacağız.

SmallNavigator : Çok fazla veri olma ihtimali olmayan tabloalarda kullanılacak.

LongNavigator : Veriler arasında dolaşılabilir. Önceki sayfa ve sonraki sayfaya geçişlerde bu navigator kullanılır.

InsUpNavigator : Kayıtlar arasında dolaşmamızı sağlayacak navigator.

Bu kontroller user control ile oluşturulacaktır. Ama diğer kontrollerimiz gibi kod ile de oluşturulabilir.

Görsel değişimi hemen görebilmek için user control kullanıldı.

"ImageCollection" ile oluşturacağımız navigator kontrolünün simgelerini değiştirebiliriz.

Navigator da "TextLocation" isminde bir ayar vardır. Bu ayar ile navigator çubuğumuza text ekleyebiliriz.

-------------------------------------------------

GridControl kısmını ayarlayacağız. Default olarak kolonsuz gelen grid'e kolon ekleyeceğiz.

Başlık ve yazı kontrollerini yapacağız.

GridControl dataları listelediğimiz ve ekranda gösterdiğimiz kontroldür.

Boş instance oluşturma örnek; "Public void MyGridView(){ }"

"MyGridInfoRegistrator" class'ının içerisinde override edebileceğimiz diğer methodlar

1- CreateHandler ile Handler'lara müdahale edebiliyoruz.

2- CreatePainter ile GridView'in şekline müdahale edebiliyoruz. Çizgilerinin rengi, kalınlığı, arka plan kalınlığı vb.

3- CreateViewInfo ile diğer farklı özelliklere müdahale edebiliyoruz.

4-

--------------------------------------------------

BandedGridView

Normal GridView' in çizgili hali olarak düşünülebilir.

1- OnColumnChangedCore methodunu override ederek öncelikle girilen değer tipinin türü "RepositoryDateEdit"

ise ortalaması için "column.AppearanceCell.TextOptions.HAlignment = HorzAlignment.Center;" komutunu kullandık

Video 32:

Bütün kontrollerimizi “MyDataLayoutControl” isimli kontrol üzerinde ayarlayacağız. Bu bize 10 satır ve 3 sütundan oluşan bir layout oluşturacak. Yani Template gibi çalışan bir tool. Kontrollerimizi üzerine bırakıp düzneli bir şekilde sıralanmış olacak.

“MyDataLayoutEdit” 10 satır yapmamızın nedeni yeni bir oluşturup ayarlarını değiştirmektense olan bir satırı silerek kolayca şekil verebiliriz.

Video 33.

Devexpress te hemen her kontrolde olan “MenuManager” özelliği bulunur. Bu özellik formda bulunan kontrol üzerine sağ tuşa tıklandığında nasıl bir menü açılacağını ayarlamak için kullanılır.

Okul Kartı formunda “Okul Adı” ve “Açıklama” kısmı form sonuna kadar gitmektedir. Bunu yapabilmek için, yapmak istediğimiz satırı seçip Option->OptionsTableLayoutItem kısmında “ColumnSpan” ve “RowSpan” isminde iki adet özellik bulunur.

ColumnSpan : Kaç adet sütunun birleştirileceği

RowSpan : Kaç adet satırın birleştirileceği

Caption olarak adlandırılan bölüm. Başlıkların bulunduğu kısımdır.



DataGridView’ de Caption ile kontroller arası boşluğu ayarlama

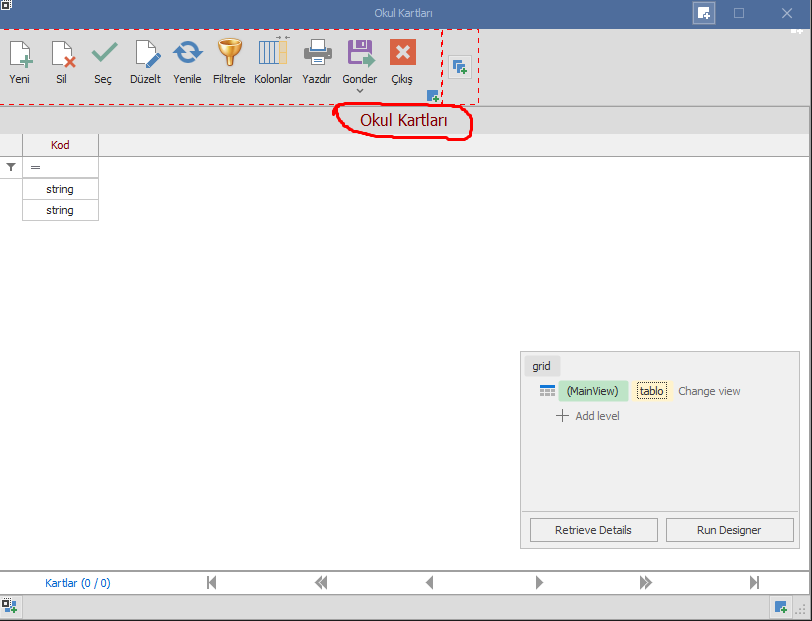
Bu Caption ile text’ler arası boşluk aralığını ayarlayabiliriz. Bunun yapabilmek için satır seçilip Options-> TextAllignMode ayarı CustomSize olarak ayarlanır.

İkinci ayarımız ise “TextToControlDistance” ayarını int bir değer girilerek uzaklık ayarlanır.

Formumuzu belirli bir orandan daha küçük veya daha büyük olmamasını istersek Option kısmında MinimumSize ve MaximumSize özelliklerini ayarlamamız gerekir.

Video 34.

Aşağıdaki resimde “Okul Kartları” yazan kısımda “MainView” tanımı yazmaktaydı. Bu satırı renkli “tablo” yazısına tıklayarak Options->ViewCaption bölümünden değiştirebiliriz.



* Sonraki kolonlara yeni kolon eklerken
* Caption: Başlıkta görünecek isim. Yani kolon başlığı
* FieldName: Source’ de yer alan, alan adı

## 62. Formlardaki kodlama işlemlerinin hangi mantıkla yapılacağı

OkulKartları UI ekranındaki ribbon control içerisindeki bütün butonları dolaşıp arka planlarını kodlayacağız.

Burada form işlemlerinde yapılacak çoğu işlem için base class a tanımlayıp kullanacağız ve her form için aynı işlemleri yazmaktan kurtulacağız.

Aşağıda önemli bir konu var.

List methodunda verileri listemelek için kullanılır. Verileri database katmanında listelemek için “BaseList” kısmında değil “List” kısmında “ToList()” methodunu kullandık.

