

Kontroller

Label	: IblAd , IblSoyad , IblSubeAd , IblSubeKod
LinkLabel	: IlblAd , IlblSoyad
Button	: btnKaydet , btnDuzelt , btnSil , btnKapat ,
TextBox	: txtAd , txtSoyad , txtSubeAd , txtSubeKod
MainMenu	: mmenuDokuman , mmenuArsiv ,
CheckBox	: chkSpor , chkKultur , chkGazete
RadioButton	: rbtnEvli , rbtnBekar , rbtnYeniKayit
GroupBox	: grpMedeniHal
PictureBox	: pctImza , pctNufusCuzdan
Panel	: pnlKimlik , pnlAdres , pnlTelefon
ListBox	: lstKategoriTip , lstSubeKod
CheckedListBox	: csltMailAdres
ComboBox	: cmbSubeAd , cmbAdresKod
ListView	: lviewGorusme
TreeView	: tviewOrganizasyon
TabControl	: tabMusteriTanim
DateTimePicker	: dtpTarih , dtpIseGirisTarih
MonthCalendar	: mcTakvim
HScrollBar	: hsbGorus
VScrollBar	: vsbGorus
Timer	: timerKayit , timerLog
Splitter	: splitterMusteri
TrackBar	: trackbarFileUpload
ProgressBar	: progbarFileUpload , progbarMuhasebe
RichTextBox	: rtxtAciklama , rtxtOneri
ImageList	: ilstMenu , ilstDokuman
HelpProvider	: hprvSicilNo
ToolTip	: ttipTara , ttipFarkliKaydet
ContextMenu	: cmenuDosya
ToolBar	: tbarKaydet
StatusBar	: sbarKaydet , sbarrAc
NotifyIcon	: niconUyari
ErrorProvider	: eprvSubeKod
DataGridView	: grdSube

Developer Express:

BarManager	: barmngMuhasebe
PopupMenu	: popmenuMusteri
NavBarControl	: navbarMuhasebe
VGridControl	: vgrdParametre
GridControl	: grdParametre
GridView	: grdwParametre
SimpleButton	: btnKaydet
DefaultLookAndFeel	: dlfMusteri
XtraTabControl	: xtabMusteri
ButtonEdit	: ebtnKaydet
CalcEdit	: calcKaydet
CheckEdit	: chkFutbol , chkBasketbol
CheckedListBoxControl	: csltMailAdres
ComboBoxEdit	: cmbSubeKod
ControlNavigator	: ctrlnavMusteri

DateEdit	: dateDogumTarih
ImageEdit	: imgMusteri
ImageListboxControl	: imglistSubeAd
ListboxControl	: lstSubeAd
LookUpEdit	: lueSubeKod
MemoEdit	: memoAciklama
PictureEdit	: pctMusteri
ProgressBarControl	: pbcUpload
RadioGroup	: rgSpor
SpinEdit	: spinAdres
TextEdit	: txtAd
TimeEdit	: timeBaslangicSaat
GrupControl	: groupMedeniHal
HScrollBar	: hsbMusteri
VScrollBar	: vsbMusteri
ImageCollection	: imgcollectCek
PanelKontrol	: pnlMusteri
ToolTipController	: ttcKimlik

Namespace :

Firat.Musteri.Forms
Firat.Musteri.Service
Firat.Musteri.Interface
Firat.Musteri.Business
Firat.Musteri.SP
Firat.Musteri.Helper
Firat.Hesap.Forms
Firat.Hesap.Service
Firat.Hesap.Genel.Service
Firat.Hesap.Detay.Service
Firat.Hesap. Interface
Firat.Hesap.Genel.Interface
Firat.Hesap.Detay.Interface
Firat.Hesap.Business
Firat.Hesap.Genel.Business
Firat.Hesap.Detay.Business
Firat.Hesap.Sp
Firat.Hesap.Genel.Sp
Firat.Hesap.Detay.Sp

Class:

FMusteri
SMusteri
IMusteri
BMusteri
SpMusteri (static T_MUSTERI , P_MUSTERI database'deki package name'leri)
HMusteri

Method:

MusteriEkle

MusteriBul

SP'ler için [get_bakiye](#) (database de ne ise aynı olacak)

Public Değişkenler:

public string [kisaAd](#)

public long [sivilNo](#)

Private Değişkenler:

private decimal [_bakiye](#) (Sadece property'leri için tanımlanlar private değişkenler _ ile başlayacak)

private int [_adresKod](#)

private string [_kisaAd](#)

Property :

Bakiye (esas teşkil eden private değişken [_bakiye](#) olarak tanımlanmalı)

AdresKod

Exception :[ex](#), [ex1](#), [ex2](#)

Oracle DotNet Provider:

OracleConnection [conn](#)

OracleCommand [cmd](#)

OracleTransaction [trans](#)

OracleParameter [prm](#)

OracleDataAdapter [da](#)

OracleDataReader [dr](#)

OracleCommandBuilder [cb](#)

System.Data

DataSet [ds](#) dsEkran,dsOperator

DataTable [dt](#) dtEkran,dtOperator

DataView [dv](#) dvEkran,dvOperator

DataRow [drow](#) drowKisi,drowOperator

DataColumn [dcol](#) dcolSubeKod,dcolSubeAd

Control Library

[CtrlLibSubeKod](#)

[CtrlLibHesapNo](#)

[CtrlLibEkNo](#)

[ucSubeKod](#)

[ucHukukiYapi](#)

[ucHesapNo](#)

- User Kontrollere yazdığımız her property,metot vs.. [x](#) ile başlamalı.Nedeni ise kod geliştirirken kendi yazdığımız bu tür özelliklere intellisense'tan kolayca ulaşmaktadır.
- Veritabanı ile ilişkisi olan her kontrolün [xEcranParam](#) Property'si tanımlamalı.
- Eğer bir veya birden fazla property'i set eden bir metot yazacaksak metodun adı "[xSetParams](#)" olmalı.
- Eğer Bir Value property lazımsa ismi [xValue](#) şeklinde olmalı ve bu değer set edildiğinde Text property'si de değiştirilmeli.
- Hesap No,Kisino,Vergi No gibi sayısal değerlerde [long](#) tipini kullanmalıyız.

For Döngülerinde kullanılan counter'lar: i,j,k

Class'lardan nesne oluştururken:

[OpenFileDialog](#) : [ofdUpload](#) , [ofdDownload](#)

[PrintDialog](#) : [pdDekont](#), [pdFis](#)

Programlama ve Kodlama ile ilgili standartlar

Açıklama satırları (Comment): Birden fazla açıklama satırı olacak ise;

```
/*
 * Açıklama satırı (Created by ..... , DD/MM/YYYY)
 * Açıklama satırı (Fonksiyonu, uyarılar)
 * Açıklama satırı (Edited by ..... , DD/MM/YYYY,
Neden edit edildiği)
```

```
/*
Tüm assembly, bu assembly'ler içerisindeki class ve bu
class'lar içerisindeki metod'lar için yukarıda standart kullanılmalı.

Tek satırlık açıklamalar için // ..... veya /* ..... */
*/ kullanılabilir.
```

Girintili yazma (Indentation) : .Net editörü girintilerde default olarak 4 karakter boşluk bırakıyor. Bu standarta uyulmalı.

```
-for döngüsü
for(int i=0;i<5;i++)
{
    .....
}
-if koşulu
if (a<b)
{
    .....
}
else
{
    .....
}
-Exception
try
{
    .....
}
catch(Exception ex)
{
    .....
}
```

Control Library Standartları

```

Assembly içerisindeki sql'ler yazılımı
string sql = @"SELECT t.kategori,
                     v.versiyon,
                     v.tarih,
                     v.op_adi,
                     v.dokuman_no
                FROM das_dokuman d,das_dokuman_versiyon
    v,das_kategori_tip t
               WHERE d.dokuman_no = v.dokuman_no
                 AND d.aktif_versiyon = v.versiyon
                 AND d.ana_kategori_tip = t.ana_tip
                 AND d.kategori_tip=t.tip
                 AND d.ana_kategori_tip=" +
lueAnaKategori.EditValue.ToString() + "@"
                 AND d.takip_no=" + txtTakipNo.Text + "";
               ORDER BY t.kategori,v_op_adi

```

```

string sql = @"UPDATE m_operator
               SET sube_kod=" + ucSubeKod.Text @"
               WHERE op_adi="" +
FSubeDegistir.ekranPrm.kulFrm.OpAd + """;

```

String Concatenation işlemlerinde string class'ı yerine StringBuilder kullanılmalı.

```

string sonuc;
for (int i=0;i<10;i++)
{
    sonuc += i.ToString();
}

yerine;

StringBuilder sonuc = new StringBuilder();
for (int i=0;i<10;i++)
{
    sonuc.Append(i.ToString());
}

```

Form Görüsel Tasarım Standartları:

Font : Tahoma
Font-Size : 8.25

Info (readonly veya disabled) alanlar için : Web.LightYellow
Labellar sağa veya sola yanaşık olabilir.

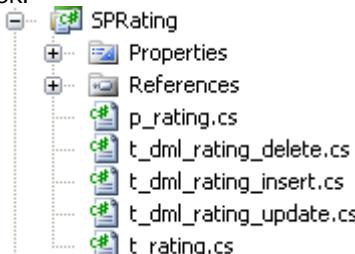
SPOObject Katmanı Standartları:

Bu katmanda orta katmanda kullanılacak SP'leri çağrıran sınıflar yazılacak. .Net-Oracle tür uyuşmazlığı nedeniyle Oracle tarafından rowtype veya özel type return eden SP'ler yazılmırken body'si ile birlikte yazılacak. Yani bu SP'lerin SELECT kısmı .Net tarafında yazılacak. İsimlerin birebir aynı olmasına dikkat edilecek.

Bu katmanda OracleConnection kurulmayacak. Conn bilgisi procedure'lere parametre olarak çağrıldığı yerden (Services veya Business Object) gönderilecektir.

SPOBjeleri yazılrken Her bir modül ayrı proje halinde olacak.Modüle ait Package'lerin her biri yeni bir .cs dosyası yani class olacak.

ÖR:



BusinessObject Katmanı Standartları:

Bu katmanda, birden fazla servis(modül) tarafından kullanılacak birden fazla SP objesini kullanan veya DML işlemi yapan anlamlı işlemler bütünü(havale,kredi kartı ödemesi vs,T_MUHASEBE gb)

Sevice Katmanı Standartları:

Client tarafından ilgili modulel alaklı istekleri karşılayacak olan katmandır. Yazılacak metodların başka servislerden de kullanılma ihtiyali var ise bu metodların içeriği business object katmanında yazılmıştır. Bu katmandan sadece çağrılmaları beklenir. Yok sadece bu servise özel ise doğrudan bu katmanın içinde geliştirileceklidir.

Dataset,Datable Standartları:

Dataset ve datatable nesneleri tek satırlık kayıt içerseler bile size'ları çok yüksek oluyor. Mümkün mertebe bu nesneleri client tarafına parametrelerle geçmekte fayda var. Orta katman ile Oracle arasında aynı networkte oldukları için herhangi bir sorun yok. Client gönderilecek datatable larda mümkünse kayıt sayısı kontrolü koymak gerekmektedir. (rownum<100)

Metod ve parametreleri ile ilgili açıklamalar

```

/// <summary>
/// Data table'ı update eder
/// </summary>
/// <param name="ci">ClientInfo</param>
/// <param name="dt">DataTable</param>

/// <param name="onErrorRollBack">Hata
durumunda rollback</param>
/// <param name="dtLast">DataTable'in
son hali</param>
/// <param name="rowsAffected">Etkilenen
kayıt sayısı</param>
/// <returns>String döner,hata yoksa
null döner</returns>

```