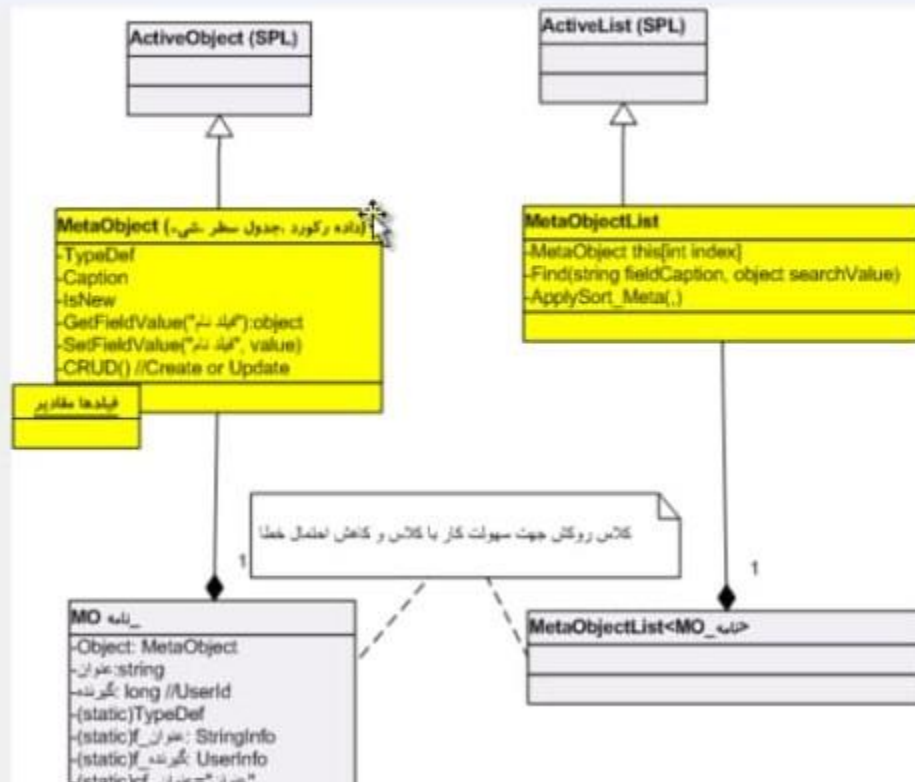


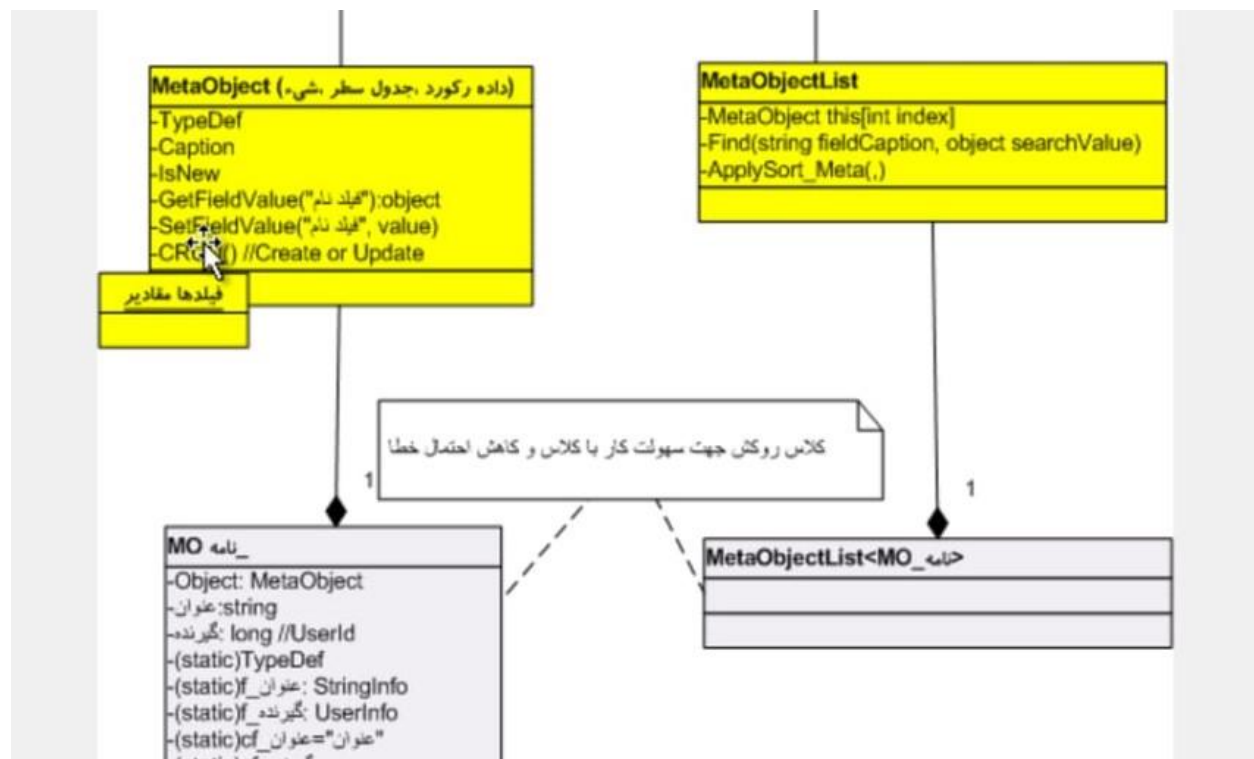
کلاسهای ساختاری و داده ای در سیستم ساز
 کلاسهای ساختاری: سیستم - موجودیت - فیلد - گزارش - فرآیند -

کلاسهای داده: MetaObjectList - MetaObject

MetaObject: پدر همه اشیاء ذخیره شونده در سیستم ساز - ارتباط با TypeDef - عنوان - متد
 گرفتن مقدار و تنظیم مقدار - LoadById - معرفی ساختار شناسه در سیستم ساز و شناسایی نوع از Id

۶.۲ کلاسهای داده





نوع کدی که مینویسیم :

C# + اسکریپت دیتابیس + جاوا اسکریپت وب

لایه بندی اجرای کد:

کدهای سمت سرور: ترجیح بر استفاده از این نوع کد برای تطابق کامل بین ویندوز و وب
 کلاینت ویندوز: رخدادهای سمت کلاینت خصوصا کدهای مربوط به دکمه ها، بعضا اعتبارسنجی
 کلاینت وب: رخدادهای خصوصا دکمه ها
 نکات:

استراتژی حداقل سازی کدهای سمت کلاینت

نیاز به کوئری نویسی برای مقاصد خاص علاوه بر امکانات گزارش ساز سیستم ساز

ترجیح بر استفاده از روشهای منطقی و C# به جای کوئری

الزام به استفاده از کوئری در مباحث کارایی و سرعت اجرا

توسعه گرامری C# در سیستم ساز: عبارت {} و @// - اثر در فیلدهای محاسباتی و ارتباط پذیری
رابطه بین فیلدها - <%%> در کوری ها

تعریف یک موجودیت از نوع متاآبجکت:

```
MO_نام موجودیت نام شی = new MO_نام موجودیت(mo);
```

ذخیره شی :

```
نام شی.CRUD();
```

نمایش یک فرم با آبجکتی جدید :

```
[deprecated] bool MetaObjectEditForm.ShowObject(string formTitle, long metaObjectId, bool readOnly)
```

مثال:

```
MO_اقدام m = new MO_اقدام();
```

```
MetaObjectEditForm.ShowObject("اقدام", m.Id, false);
```

گرفتن لیست انتخاب شده ها(از گرید) (ویندوز) :

```
MetaObjectList UtilUi.GetSelectedObjectList(object senderOrContainer)
```

مثال :

```
MetaObjectList list=UtilUi.GetSelectedObjectList(sender);
```

مثلا می خواهیم بعد از گرفتن لیست مقدار یک فیلد آن را تغییر دهیم و مجدد ذخیره کنیم :

```
void MetaObject.SetFieldValue(string caption, object value)
```

```
MetaObjectList list=UtilUi.GetSelectedObjectList(sender);
```

```
foreach(MetaObject mo in list){
    mo.SetFieldValue(مقدار , نام فیلد);
    mo.CRUD;
}
```

وب :

```
var report = Bw.App.GetActiveReport();
var mo = report.GetSelectedMo();
```

لود ساختار سازمانی:

```
WorkCenterList wcList = WorkCenter.LoadAll();
```

مثال :

حذف کاربری از ساختار سازمانی :

```
Barsa.Spl.BasicInfo.WorkCenterList lst =
Barsa.Spl.BasicInfo.WorkCenter.LoadAllRelatedWcsToUser(7135205500000000102);
```

```
foreach(Barsa.Spl.BasicInfo.WorkCenter wc in lst)
{
    Barsa.Spl.BasicInfo.WorkCenter wc1=wc;
    wc.AssignedUsers.RemoveByProperty("DisplayName","راهنر سیستم");
    wc.Id=wc.Id;
    wc.CRUD();
}
```

به دست آوردن یک نود ساختار سازمانی با استفاده از آی دی آن :

```
var wc = WorkCenter.LoadById(1051505510000000429);
```

برای پیدا کردن یک سمت با نام خاص در ساختار بعد از لود ساختار :

```
WorkCenterList wcList = WorkCenter.LoadAll();
string wcPath = string.Empty;
if(wcList.Count>0){
    foreach(WorkCenter wc in wcList){
        if(wc.DisplayName == "معاونت فني مهندسي"){
            Wc1 = wc;
        }
    }
}
```

حال مشخصه هایی که در ۱ WC داریم :

```
Wc\ .GetNewId();
Wc\ .DisplayName;"ساختار تست" =
Wc\ .Name;"ساختار تست" =
Wc\ .ParentId = root.Id;
Wc\ .WcType = WorkCenterType.Ou;
Wc\ .Path = root.Path;
Wc\ .CRUD();
```

وب :

```
//%[MetaCommandEvent("ثبت مرخصي")]
this.OnSystem_مرخصي_ثبت = function(sender, e)
{
    Bw.App.StartWorkflow(
        {SystemName:'منابع انساني شرکت برسا', WorkflowName:'درخواست مرخصي'}
    );
}
```

ساختار متد شروع فرایند در ویندوز و وب :

ویندوز :

کد شروع یک فرآیند در رخداد دکمه سراسری

[MetaCommandEvent ("دوره ارزیابی جدید")]

```
public static void OnSystem_دوره ارزیابی جدید(object sender, MetaEventArgs e)
```

```
{
```

```
    پرونده ارزیابی عملکرد MO_پرونده ارزیابی عملکرد parvande = new MO_پرونده ارزیابی عملکرد();
```

```
    WorkflowDef.Start("شروع", "فرآیند آغاز ارزیابی", "شروع", SessionMgr.UserId, parvande.Id, null);
```

```
////
```

شروع فرآیند در وب

```
Bw.App.StartWorkflow(options)
```

Parameters:

options:

an Object with this properties:

WorkflowId: (optional)

Encoded Id of workflow. if this is provided, there is not need for SystemName and WorkflowName . Default is 0

SystemName: (optional)

Name of System. if WorkflowId is not provided, we should set this param and WorkflowName. Default is null

WorkflowName: (optional)

Name of Workflow. Default is null

StartActivityName: (optional)

Name of start activity. Default is null

StartObjectId: (optional)

MetaObject id

روش رپری رسیدن به مقدار در فیلد چند مقداری: خیلی مهم :

```
libSys_.MO_درخواست ثبت نام registration = mo;
```

```
registration.تایید شده.Values درخواست ثبت نام.وضعیت MO_.libSys_ (long) = وضعیت.وضعیت registration;
```

```
registration.CRUD();
```

```
return 0;
```

مزایای این روش به جای اعداد int :

- کد خوانا تر
- داینامیک بودن نامها حفظ می شود

پیدا کردن عضو مرتبط از روی کاربر سیستمی : این کد کاربرد جستجو در رکورد های یک موجودیت با مقادیر خاص از فیلدهای گفته شده را با TypeDef.SelectFirstObjectBy نشان میدهد .

```

MetaObject person= new libSys_.MO_شخص_کتابخانه_نهایی_();
MetaObject member = new libSys_.MO_عضو_();
person =libSys_
.MO_نام_کار.f_شخص_کتابخانه_نهایی_().TypeDef.SelectFirstObjectBy(libSys_.MO_شخص_کتابخانه_نهایی_().Util.CurrentUserId);
member =
libSys_.MO_عضو_().TypeDef.SelectFirstObjectBy(libSys_.MO_شخص_کتابخانه_نهایی_().f_شخص_کتابخانه_نهایی_().Id);

return member;

```

کد یافتن نقشها در فرآیند کتابخانه

```

libSys_.MO_درخواست_عضویت_کتابخانه_ = mo;

Report FindLibraryManager = Report.LoadByName;("یافتن مسئول کتابخانه")

//اجرای یک گزارش ماژول تعریف شده توسط طراح و اجرای آن
((SingleCondition)FindLibraryManager.ReportDataObject.Condition.Children[0]).Parameter.Value = request;کتابخانه.

MetaObject result = FindLibraryManager.ReportDataObject.ExecuteFirst;()

DebugHelper.BreakIfControlIsDown;()

libSys_.MO_پرسنل_ =result;

libSys_.MO_شخص_کتابخانه_نهایی_().personel_ = personel_شخص_کتابخانه_نهایی_().MetaObject;

long userId = person;نام_کاربري.

return userId;

```

ساخت کارت عضویت :

```

Barsa.Workflow.ActivityRunContext context = contextObject as Barsa.Workflow.ActivityRunContext;

Library.MO_عضویت_درخواست_ = mo;

DebugHelper.Break();

Library.MO_کارت_عضویت_ newreg = new Library.MO_کارت_عضویت_();

{
پیوست. = joinReq. تصویر_فیشواریز
}

```

```
(نام + "#-" , joinReq.کتابخانه_Typed.Id.ToString(), joinReq.کتابخانه_Typed.۱۰۰۰) = Sequencer.GetNextValue(joinReq.کتابخانه_Typed.Id.ToString(), joinReq.کتابخانه_Typed.۱۰۰۰) , شماره کارت عضویت ,  
تاریخ عضویت = joinReq.تاریخ عضویت ,  
نوع عضویت = joinReq.نوع عضویت ,  
کتابخانه = joinReq.کتابخانه ,  
عضو = joinReq.عضو ,  
وضعیت ثبت نام = true  
};
```

```
if (joinReq.مدت زمان عضویت == (long) Library.MO_مدت زمان عضویت_Values.یکماهه)  
{  
    newreg.تاریخ پایان عضویت = newreg.تاریخ عضویت.AddMonths(1);  
}  
else if (joinReq.مدت زمان عضویت == (long) Library.MO_مدت زمان عضویت_Values.سهماهه)  
{  
    newreg.تاریخ پایان عضویت = newreg.تاریخ عضویت.AddMonths(3);  
}  
else if (joinReq.مدت زمان عضویت == (long) Library.MO_مدت زمان عضویت_Values.ششماهه)  
{  
    newreg.تاریخ پایان عضویت = newreg.تاریخ عضویت.AddMonths(6);  
}  
else if (joinReq.مدت زمان عضویت == (long) Library.MO_مدت زمان عضویت_Values.یکساله)  
{  
    newreg.تاریخ پایان عضویت = newreg.تاریخ عضویت.AddYears(1);  
}
```

```
TimeSpan diff = (newreg.تاریخ پایان عضویت - newreg.تاریخ عضویت);  
newreg.مهلت اعتبار = diff.Days;  
newreg.CRUD();  
context.SetVariable("کارت عضویت", newreg.Id);  
return newreg;
```

تغییر فرم جاری فرآیند: خیلی مهم

```
context.SetVariable("کارت عضویت", newreg.Id);
```


کد دیباگ کردن در محیط ویژوال استادیو :

```
DebugHelper.Break();
```

نحوه استفاده از ریپر ها اگر به صورت اتوماتیک در بخش ادیتور کد دسترسی نداشته باشیم :

```
//@System:مدیریت کتابخانه_نوروزیان
```

```
Using LibSysN;
```