

PB 应用软件开发

The Design of Component In PB Application Software Development

姜合 孙喆红 杨春花

(济南山东轻工业学院 计算机科学与技术系 250100)



摘要: 在利用PB进行应用软件开发的过程中,常需对一些窗口、控件进行重复设计使用,极大的浪费了程序员的时间,为了提高应用程序的构造速度,我们利用软件复用的思想,将一些操作做成可复用构件,并存入构件库中,以便于将来的重复使用。本文简单介绍了构件的设计方法。

关键词: 软件复用 数据窗口对象 构件 PB

1 引言

各种面向对象开发语言中的基础类库(PB的PFC、VC++的MSFC等),许多开发者在软件开发过程中都已经使用过,这些都是很好的通用构件库,给开发者带来了极大的方便,这要求开发者对语言及类库有较深刻的理解,对多数开发者来说较为困难。我们在这里提到的复用构件是利用已有的类库或其他对象设计的,功能更加复杂且相对通用的构件如各类控件、各类窗口、甚至应用程序框架等,即所谓大粒度构件。

PB是一种可视化、面向对象的快速应用开发工具,它包含数据窗口等多种对象类,已被广大数据库开发人员所采用,可以方便地访问数据库。但在实际开发中,特别在大型项目的开发中,常常遇到某些操作重复出现,如查询窗口、打印窗口等,每个应用系统都有,这些窗口实现的功能相同或相似,如果每次都为它们重复编程明显增加工作量,维护也困难,也没有必要。事实上,只要我们对这些操作精心设计成通用的构件,存放到指定的库(.pbl)中,以后遇到类似的操作只需对其稍加改动即可复用。根据我们在使用PB进行软件开发中的一些体会,给出了PB应用程序开发中一些可复用构件的设计思想。

2 可复用构件的特点及设计思想

2.1 主要特点

通用性: 可以为多个软件经常的反复的使用。

内部功能高效: 构件是构造软件系统的基本单位,其内部代码高效与否直接影响到组成的软件系统的性能,因此代码必须高效、精炼,检测试易理解。提供让复用者易于理解的机制,以便复用时能正确选择构件。

易调整: 较高的通用性是在开发特定软件时,很难直接使用,必须把它具体化,做一定修改。

2.2 设计思想

设计可复用构件。我们要遵循“design by reuse”和“design for reuse”的原则,重于复用,即设计的构件尽量相对独立,具有一定的通用性。为此我们采取设计入口参数的方法实现。

3 几种可复用构件的设计

以下我们介绍在应用软件中经常出现的综合窗口、打印窗口、控件、框架等构件的设计思想。

3.1 综合窗口

这一窗口在每个应用软件中都会多次出现,涉及的内容相同或相似,

主要包括新增、保存、删除、详细、查询、排序等，我们在此仅介绍排序、过滤。如图1所示。

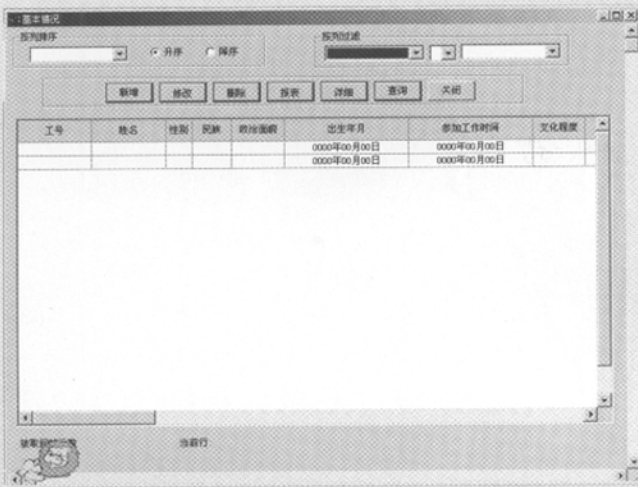


图1 综合窗口

设计综合窗口的关键是设计几个 Windows 窗口函数。

本窗口涉及到的 Windows 窗口函数主要有以下三个，即对下拉式列表框 ddb_data, ddb_filter, ddb_data 进行初始化的 Wf_reset_ddlb(), 进行排序的 Wf_sort_dw(), 查询过滤的 Wf_filter_dw()。

(1) Wf_reset_ddlb() 的主要功能

定义变量

Int li_column_count, li_counter

String ls_column_name ,

初始化

ddlb_data.reset(); ddb_filtercol.reset(); ddb_sortcol.reset()

以下算法是用数据窗口对象的列名给下拉式列表框 ddb_sortcol , ddb_filtercol 赋值。

li_column_count= Integer(dw_1.Object.DataWindow.Column.Count)

For li_counter=li_column_count to 1 step -1

ls_column_name=dw_1.Describe("#"+string(li_counter)+".Name")

ddlb_sortcol.InsertItem(ls_column_name,1)

ddlb_filtercol.InsertItem(ls_column_name,1)

Next

设置排序参数

ddlb_sortcol.text = ls_column_name

is_sort_sequence = "a"

rb_asc.Checked=True

调用窗口函数 wf_sort_dw()进行排序

wf_sort_dw(ddlb_sortcol.Text,is_sort_sequence)

dw_1.setfilter("")

dw_1.filter()

(2) wf_sort_dw(string is_column, string is_sort_sequence)

传递参数，进行排序，算法如下：

dw_1.SetSort(is_column+" "+ is_sort_sequence) // 参数格式设置

dw_1.Sort()

(3) Wf_filter_dw()

定义变量

string ls_quote_start, ls_quote_end, ls_quote, ls_column_name, ls_filter
检查 ddb_data, ddb_filtercol, ddb_sortcol 是否为空，若为空，

messagebox()给出提示，不能为空。

形成过滤表达式，不同的数据类型，过滤表达式的构造方法不同。

If left(ls_col_type,4) = `deci` then // 选择的过滤列为数字型时

ls_quote = ""

ls_column_name = Trim(ddlb_filtercol.text)

ls_filter = ls_column_name + ddb_op.text + ls_quote + ls_match_data + ls_quote

Elseif left(ls_col_type,4) = `date` then // 选择的过滤列为日期型时

ls_quote = ""

ls_quote_start = datetime(" " + ls_quote_end + " ")

ls_column_name=Trim(ddlb_filtercol.text)

ls_filter=ls_column_name+ddb_optext+ls_quote_start+ls_quote +ls_match_data+ls_quote+ls_quote_end

else // 选择的过滤列为字符型时

ls_quote = ""

ls_column_name= "upper("+Trim(ddlb_filtercol.text)+")"

ls_filter = ls_column_name + ddb_op.text + ls_quote + ls_match_data + ls_quote

End if

w_1.Setfilter(ls_filter) // 过滤设置

dw_1.filter()

sle_rows.text = String(dw_1.RowCount())

其中列的数据类型有下式获得。

ls_col_type= dw_1.Describe(ls_column_name+".ColType")

有了这几个 Windows 窗口函数，我们就很容易写出 ddb_data, ddb_filter, ddb_data 及过滤按钮的脚本。也可以增加一些其他功能，在此不再介绍。

3.2 通用打印窗口

建立一通用打印对话框，如图2所示，调用此窗口对数据窗口进行打印操作，下面简要介绍其设计思想。主要介绍窗口的事件及确认按钮脚本。

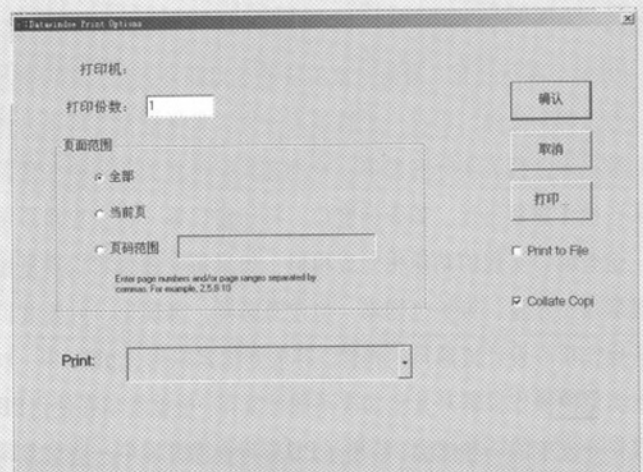


图2 通用打印设置窗口

定义变量

```
datawindow idw_dw
string i_szFileName
窗口 Open 事件
从 Message 对象中提取变元给 idw_dw
idw_dw = message.powerobjectparm
sle_printer.text=idw_dw.describe('datawindow.printer')
若指定了 Print to file 选项, i_szFileName 将保存文件名
i_szFileName=Trim(String(idw_dw.Object.DataWindow. Print.FileName))
    确认操作脚本
string szPage, szModify
long nIndex
szModify="DataWindow.Print.Copies= "+em_copies.text
if cbx_collate.checked then //对复选框 Collate Copies 进行设置
    szModify = szModify + " datawindow.print.collate = yes"
else
    szModify = szModify + "datawindow.print.collate = no"
end if
If cbx_print_to_file.checked then //对复选框 Print to File 进行设置
    szModify= szModify+"DataWindow.Print.FileName="+i_szFileName
Else
    szModify= szModify+"DataWindow.Print.FileName=''"
End if
If rb_all_pages.Checked Then
    szModify= szModify+"DataWindow.Print.Page.Range=''"
Elseif rb_current_page.Checked Then
    szPage=idw_dw.Describe("Evaluate('Page()'," +string(idw_dw.getrow()
+)" ")")
    szModify= szModify+"DataWindow.Print.Page.Range=' "+szPage+"'"
Else
    szModify= szModify+"DataWindow.Print.Page.Range='
"+sle_page_range.text+"'"
End if
nIndex=ddlb_range_include.FindItem(ddlb_range_include.text,0)
szModify= szModify+"DataWindow.Print.Page.RangeInclude="+string
(nIndex-1)
idw_dw.modify(szModify)
parent.visible=False //当系统打印对话框出现时使打印窗口不可见
idw_dw.print(true)
this.SetFocus()
close(parent)
```

3.3 通用控件设计

插入、删除、保存等也是数据窗口经常要使用的操作。用自定义控件实现这些操作是一种可行的方法。

设计一通用的全局函数 f_select_current_row(), 返回当前行号。

```
long ll_currentrow
```

```
adw_datawindow.setredraw(false) // adw_datawindow 声明为
datawindow 类型
```

```
ll_currentrow = adw_datawindow.GetRow()
adw_datawindow.SelectRow(0,False)
adw_datawindow.SelectRow(ll_currentrow,True) // 选择行
adw_datawindow.setfocus()
adw_datawindow.setredraw(true)
return ll_currentrow // 返回当前行号
```

其中, 新增操作的脚本如下:

```
dw_1.InsertRow(dw_1.GetRow()+1)// 插入行
dw_1.SetRow(dw_1.GetRow()+1)
dw_1.SetColumn(1)
```

```
f_Select_Current_Row(dw_1) // 调用上述全局函数
对其他操作可采用类似的方法实现。
```

3.4 应用框架的复用

应用系统主窗口, 菜单设计, 工作区的确定等, 都可以用统一的框架实现。下面以工作区的确定来说明其实现思路。

通过主窗口 w_main 的 resize 事件很容易确定工作区的大小。

```
setredraw(false)
```

```
int WSwidth, WSheight // 定义整型变量
```

```
WSwidth=workspacewidth (this)
```

```
WSheight=workspaceheight (this)
```

```
WSheight=WSheight-(pb_ok.height)-45 //pb_ok是主窗口中的一按钮控
```

件, 为通用性可用一固定值代替

```
WSwidth = WSwidth - pb_ok.width - 65
```

```
mdi_1.Move (this.workspaceX()+225,this.workspaceY()+80) // 设置工
作表的合适位置
```

```
mdi_1.Resize (WSwidth,WSheight) // 设置工作区的大小
```

```
setredraw(true)
```

对于系统中的其他窗口, 可以通过定制将其放在这一工作区中。

4 结束语

以上讨论了 PB 开发中通用构件的设计方法, 对有经验的程序员, 可使用 PFC 库来创建自己的通用构件。随着这些复用构件的不断积累, 开发者逐渐形成自己的构件库, 必将加快应用软件的开发速度。 ■

参考文献

- 1 杨美清等, 软件复用, 软件学报, Vol.6, No.9, 1995.
- 2 [美] Gallagar S, Herbert S. PowerBuilder6.0 程序开发大全 [M], 机械工业出版社, 1998.