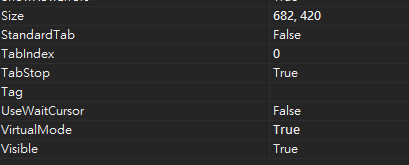
C# tips :

1. 盡量不 Class包class(DLL) ，若改寫可Orverride或另設一函數 巢狀類別不在這範圍(If也能用其他方式替代)
2. Datagridview 可以用虛擬化(virtualmode)方式來實現List對應資料視窗Cell的對應管理，記得開啟CellValueNeeded值更新對應功能與CellValuePushed資料視窗更動值能夠回推回List的資料





List<> lReadData(修改矩陣)

=>

Dgv\_ReadDataGrid\_CellValueNeeded(自己不能控制值)

1. 程式中結構以變數為準，若後方修改可避免程式錯位問題
2. List的格式嚴謹所以能輸入格式限制方式也嚴謹，避免跳出範圍外或有不相干字元(要限制輸入格式問題)
3. 陣列格式
   * 1. [] ：特定型別，固定長度的陣列，長度需事先宣告。

Int arr[5]

* + 1. List<>：特定型別，不固定長度的陣列。

List<string>

* + 1. Array：不特定型別，固定長度的陣列，長度需事先宣告。

Array arr[5]

* + 1. ArrayList：不特定型別，不固定長度的陣列。

ArrayList

補充：

Array與ArrayList是靠儲存 Object(弱型別)來達成非特定型態儲存，所以在執行階段調用時，會有轉型的成本跟風險，盡量避免使用。

Array是抽象類別，不能用new array建立。必須呼叫Array.CreateInstance() 來實作。

ArrayList的命名空間是System.Collections 預設不會載入，必須自行引用。

//===================================================================

Git和Subversion的命令比較表

|  |  |  |
| --- | --- | --- |
| **操作** | **Git** | **Subversion** |
| 複製數據庫 | git clone | svn checkout |
| 提交 | git commit | svn commit |
| 查看提交的詳細記錄 | git show | svn cat |
| 確認狀態 | git status | svn status |
| 確認差異 | git diff | svn diff |
| 確認記錄 | git log | svn log |
| 增加 | git add | svn add |
| 移動 | git mv | svn mv |
| 刪除 | git rm | svn rm |
| 取消修改 | git checkout / git reset | svn revert (※1) |
| 建立分支 | git branch | svn copy (※2) |
| 切換分支 | git checkout | svn switch |
| 合併 | git merge | svn merge |
| 建立標籤 | git tag | svn copy (※2) |
| 更新 | git pull / git fetch | svn update |
| 上傳到遠端 | git push | svn commit (※3) |
| 忽視檔案清單 | .gitignore | .svnignore |

※1. SVN的 revert 是用來取消修改，但Git的 revert 是用來取消提交。即使是同樣的命令，在SVN和Git裡的含義是不同的。

※2. SVN的分支與標籤在構造上是相同的，但在Git上明顯是不一樣的。

※3. SVN沒有本地端數據庫/遠端數據庫的概念，所以一提交馬上就會上傳到在遠端裡。但Git的本地端數據庫和遠端數據庫的表現方法是不一樣的。

//===================================================================

資料庫建置

Server用法

1. 安裝SSMS的windows的SQL Server
2. 以SQL Server mansgement studio 編輯資料表

指令相關:

select colA,colB,colC into tNewTable from tOldTable where colA=’x’ and colB is not null；

select後面是一個個的欄位，要哪個選哪個

into意味著放到一張新表裡面，沒有就是查詢出來

where之後的就是我們的條件，等於某個值，或者是不是空值

is表示”是否為”

not表示”不是”

null表示空值

//-----

select colA from oldtable group by colA

//----

增加欄位：alter tTable sTableName add sColName sVarChar;

減少欄位：alter tTable sTableName drop column colname;

//==========================================================

路徑:

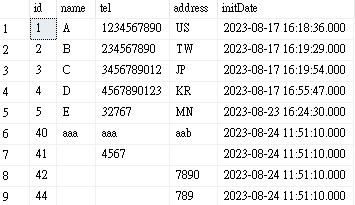
1. 當前路徑:
   * 1. string str = System.Environment.CurrentDirectory;
     2. Result: C:\xxx\xxx
2. 應用程序的可執行文件的路徑:
   * 1. string str = System.Windows.Forms.Application.StartupPath;
     2. Result: C:\xxx\xxx
3. 主模塊的完整路徑，包含文件名:
   * 1. string str = System.Diagnostics.Process.GetCurrentProcess().MainModule.FileName;
     2. Result: C:\xxx\xxx\aaa.exe
4. 獲取應用程序的當前工作目錄:
   * 1. string str = System.IO.Directory.GetCurrentDirectory();
     2. Result: C:\xxx\xxx

SQL資料庫檔案的實際路徑

指令:

SELECT name, physical\_name

FROM sys.master\_files;



SQL指令:

1.查詢項目

SELECT TOP 1000 [id]

,[name]

,[tel]

,[address]

,[initDate]

FROM [MVC\_TestDB].[dbo].[Table\_Customers]

//-------------------------------------------------

2.更新項目

DECLARE @name varchar(200)--先定義

DECLARE @tel varchar(200)

DECLARE @address varchar(200)

DECLARE @id INT

SET @name = 'aaa'--給值

SET @tel = 'aaa'

SET @address = 'aaa'

SET @id = 40

UPDATE [MVC\_TestDB].[dbo].[Table\_Customers] SET name = @name, tel = @tel,

address = @address WHERE id = @id

//---------------------------------------------------------------

3.刪除

DECLARE @id INT

SET @id = 43

DELETE [MVC\_TestDB].[dbo].[Table\_Customers] WHERE id = @id

4.新增(其中id與initDate會自己生成不用加)

DECLARE @name varchar(200)--先定義

DECLARE @tel varchar(200)

DECLARE @address varchar(200)

SET @name = 'aaa'--給值

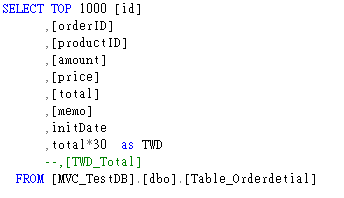
SET @tel = 'aaa'

SET @address = 'aaa'

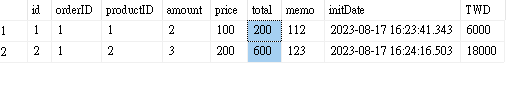
INSERT INTO [MVC\_TestDB].[dbo].[Table\_Customers] (name, tel, address)

VALUES (@name, @tel, @address)

語法很活 要多看怎麼應用



Result



//-----------------------------------------------------------------------------------------------

在ShareMemory中測試了兩個方案的傳輸值

1. ShareMemory: 在記憶體中MemoryMappedFile一個區塊，使單一執行續能對區塊寫入(要定義區塊)，要利用WaitOne的方式去鎖定or解除執行續使用，反之讀取一樣的方式。
2. SendMessage: 以winuser.h的函數庫做視窗的訊息傳遞，因為對應數值皆為包裝完成的方式，基本上是借用FindWindow找出要傳送之視窗代碼、以SendMessage附帶用COPYDATASTRUCT包裝輸出訊息 與WndProc 的listener方式讀取COPYDATASTRUCT的資料解為輸出資料

//---------------------------------------------------------------------------------------

Volatile

三大特性

1. volatile修飾的變數具有可見性

2. volatile避免指令優化重排

3. volatile不保證Atomic