

# Bibliotecas de Classes

## MFC

MFC e a API ODBC

# MFC *Microsoft Foundation Class Library*

MFC *Library* é um conjunto de classes C++ destinada ao desenvolvimento de aplicações para o Windows que vai de encontro aos seguintes objectivos de desenvolvimento:

- Redução significativa do esforço na escrita de aplicações;
- Velocidade de execução comparável ao do uso directo da API Win32 em C;
- Mínimo *overhead* no tamanho do código;
- Permitir chamar qualquer função em C do Windows directamente.
- Conversão fácil de uma aplicação C para uma escrita em C++;
- Tirar partido da experiência adquirida da programação em C para o Windows;
- Uso mais simples da Windows API com C++ do que com C.
- Fácil de usar, contudo com poderosas abstracções de características complexas tais como ActiveX, suporte para base de dados, e outras;
- Verdadeira C++ Windows API que efectivamente usa as características da C++.

# MFC *Microsoft Foundation Class Library*

## Hierarquia das classes por categoria

### CObject

- Application Architecture
- Exceptions
- File Services
- CComdTarget
  - Window Support
  - CWnd
    - Frame Windows
    - Dialog Boxes
    - Views
    - Controls
    - Control Bars
    - Property Sheets

- Graphical Drawing
- Control Support
- Graphical Drawing Objects
- Menus
- Command Line
- ODBC Database Support
- DAO Database Support
- Synchronization
- Windows Sockets

- Arrays
- Lists
- Maps
- Internet Services

### Classes Not Derived from CObject

- Internet Server API
- Run-time Object Model Support
- Simple Value Types
- Structures
- Support Classes
- Typed Template Collections
- OLE Type Wrappers
- OLE Automation Types
- Synchronization

# MFC/ODBC

<b>CDatabase</b>	Encapsula a ligação à fonte de dados;
<b>CRecordset</b>	Encapsula um conjunto de dados seleccionados da fonte de dados. Permite navegar entre registos, actualizar registos , permitindo filtrar e ordenar a selecção;
<b>CDBException</b>	Tem o objectivo de tratar excepções no acesso e processamento de dados;
<b>CFieldExchange</b>	Fornece a informação de contexto para a troca de dados entre os campos membros de dados do objecto <i>recordset</i> e os campos dos registos da respectiva tabela da fonte de dados.

# MFC/ODBC Classes Relacionadas

<b>CLongBinary</b>	Encapsula o mecanismo que trata do armazenamento de objectos binários de grandes dimensões;
<b>CDBVariant</b>	Permite o armazenamento de valores sem existir a preocupação de qual é o tipo de dados correspondente a esse valor;
<b>CRecordView</b>	Fornece uma vista sobre a forma de formulário directamente ligada a um objecto do <i>recordset</i> . Existe um mecanismo da troca de dados (DDX - <i>dialog data exchange</i> ) entre o recordset e os controlos do formulário. Os controlos do formulário são desenhados tal como numa caixa de diálogo com o editor de recursos.

# Classe *CDatabase*

## *CDatabase::Open*

```
virtual BOOL Open( LPCTSTR lpszDSN, BOOL bExclusive = FALSE,
                  BOOL bReadOnly = FALSE,
                  LPCTSTR lpszConnect = "ODBC;",
                  BOOL bUseCursorLib = TRUE );

virtual BOOL OpenEx( LPCTSTR lpszConnectionString, DWORD
                    dwOptions = 0 );
```

```
CDatabase m_dbCust;
```

```
// Connect the object to a read-only data source where the ODBC
connection dialog box
```

```
// will always remain hidden
```

```
m_dbCust.OpenEx( _T( "DSN=MYDATASOURCE; UID=JOES" ),
                CDatabase::openReadOnly | CDatabase::noOdbcDialog );
```

# Classe *CRecordset*

Membros de dados	Construção	Atributos	Atualização	Navegação
<u>m_hstmt</u> <u>m_nFields</u> <u>m_nParams</u> <u>m_pDatabase</u> <u>m_strFilter</u> <u>m_strSort</u>	<u>CRecordset</u> <u>Open</u> <u>Close</u>	<u>CanAppend</u> <u>CanRestart</u> <u>CanTransact</u> <u>CanUpdate</u> <u>GetRecordCount</u> <u>GetTableName</u> <u>GetSQL</u> <u>IsOpen</u> <u>IsBOF</u> <u>IsEOF</u> <u>IsDeleted</u>	<u>AddNew</u> <u>CancelUpdate</u> <u>Delete</u> <u>Edit</u> <u>Update</u>	<u>Move</u> <u>MoveFirst</u> <u>MoveLast</u> <u>MoveNext</u> <u>MovePrev</u>

# Acesso à Base de Dados

```
CDatabase db;
db.OpenEx( NULL, CDatabase::forceOdbcDialog );

CRecordset rs( &db );
rs.Open( CRecordset::forwardOnly,
        _T( "SELECT * FROM SomeTable" ) );

CDBVariant varValue;
short nFields = rs.GetODBCFieldCount( );
while( !rs.IsEOF( ) ) {
    for( short index = 0; index < nFields; index++ ) {
        rs.GetFieldValue( index, varValue );
        // do something with varValue
    }
    rs.MoveNext( );
}
rs.Close( );
db.Close( );
```



# ***CRecordset*** Especialização

**Exemplo de Referência**

**Vendas de uma Loja de Informática**

**A aplicação no *Visual Studio***

# Transacções

```
BOOL CEnrollDoc::RemoveStudent( CString strStudentID )
{
    if ( !m_dbStudentReg. BeginTrans( ) ) return FALSE;
    CEnrollmentSet rsEnrollmentSet(&m_dbStudentReg);
    rsEnrollmentSet.m_strFilter = "StudentID = " + strStudentID;

    if ( !rsEnrollmentSet. Open(CRecordset::dynaset) ) return FALSE;
    CStudentSet rsStudentSet(&m_dbStudentReg);
    rsStudentSet.m_strFilter = "StudentID = " + strStudentID;
    if ( !rsStudentSet. Open(CRecordset::dynaset) ) return FALSE;

    TRY {
        while ( !rsEnrollmentSet.IsEOF( ) ) {
            rsEnrollmentSet.Delete( ); rsEnrollmentSet.MoveNext( );
        }
        rsStudentSet.Delete( ); m_dbStudentReg. CommitTrans( );
    }
    CATCH_ALL(e) {
        m_dbStudentReg. Rollback( ); return FALSE;
    } END_CATCH_ALL
    rsEnrollmentSet.Close( ); rsStudentSet.Close( ); return TRUE;
}
```

# Bibliotecas de Classes

## MFC

MFC e a API ODBC