

Elemi alkalmazások IV. - Bank

Refszám	Név	Cím	Státusz
5555	Kovács Anna	3434 Alma u. 8.	magán
1111	Kis Péter	1234 Cseresznye u. 7.	üzleti
3333	Nagy Sára	6666 Meggy u. 1.	magán
2222	Nagy Pál	4321 Körte u. 13.	magán
6666	Kiss Katalin	1212 Barack u. 7.	üzleti
4444	Nagy Katalin	7878 Szilva u. 123.	üzleti

Ready NUM

## Elemi alkalmazások fejlesztése IV.

### 3. előadás

#### 2. Feladat: Rendezett lista

Módosítsuk az „Adatbáziskezelés közvetlen eléréssel – I.” fejezetben elkészült alkalmazásunkat úgy, hogy egy adott oszlopra kattintva az adott oszlop szerinti rendezettségben jelenjenek meg adataink.

Elemi alkalmazások IV. - Bank

Refszám	Név	Cím	Státusz
1111	Kis Péter	1234 Cseresznye u. 7.	üzleti
2222	Nagy Pál	4321 Körte u. 13.	magán
3333	Nagy Sára	6666 Meggy u. 1.	magán
4444	Nagy Katalin	7878 Szilva u. 123.	üzleti
5555	Kovács Anna	3434 Alma u. 8.	magán
6666	Kiss Katalin	1212 Barack u. 7.	üzleti

Ügyfél lista 6 elem Refszám szerint rendi

Refszám szerint rendezve

Elemi alkalmazások IV. - Bank

Refszám	Név	Cím	Státusz
5555	Kovács Anna	3434 Alma u. 8.	magán
3333	Nagy Sára	6666 Meggy u. 1.	magán
2222	Nagy Pál	4321 Körte u. 13.	magán
1111	Kis Péter	1234 Cseresznye u. 7.	üzleti
6666	Kiss Katalin	1212 Barack u. 7.	üzleti
4444	Nagy Katalin	7878 Szilva u. 123.	üzleti

Ügyfél lista 6 elem Státusz szerint rendi

Státusz szerint rendezve

## Metódusok a rendezett lista megjelenítéséhez

### FillHeader

Felépíti a listánkat. ✓

### FillSortedData (← FillData)

Rendezetten kitölti a listánkat az adatbázis adataival.

### OnColumnclick

A kattintás callback függvénye

### OnInitialUpdate

Első megjelenítés

### OnColumnclick

FillHeader ✓

FillSortedData

Refszám	Név	Cím	Státusz
1111	Kis Péter	1234 Cseresznye u. 7.	üzleti
2222	Nagy Pál	4321 Körte u. 13.	magán
3333	Nagy Sára	6666 Meggy u. 1.	magán
4444	Nagy Katalin	7878 Szilva u. 123.	üzleti
5555	Kovács Anna	3434 Alma u. 8.	magán
6666	Kiss Katalin	1212 Barack u. 7.	üzleti

Ügyfél lista 6 elem Refszám szerint rend...

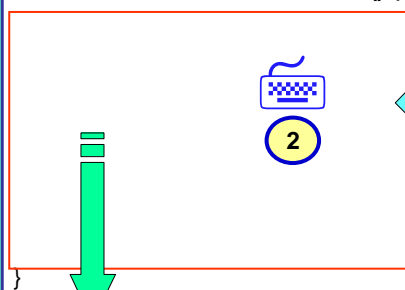
## Nézet osztály: FillSortedData()

### BankView.h

```
public:
    ...
    void FillData();
    void FillSortedData(int sColumn);
    ...
```

### BankView.cpp

```
void CBankView::FillSortedData() {
```



Workspace/ClassView/Jobb egérfél/  
Add Member Functions

### Add Member Function

Function Type:  
void

Function Declaration:  
FillSortedData(int sColumn)

Access:  
☒ Public ☐ Protected ☐ Private

☐ Static ☐ Virtual

1

Ezt a metódust a már meglévő  
FillData metódus alapján fogjuk  
elkészíteni.

### Nézet osztály: FillData()

#### BankView.cpp

```
void CBankView::FillData() {
```

```
    Ugyfel uf;  
    CListCtrl* pList= &GetListCtrl();  
    CBankDoc* pDoc = GetDocument();  
    pDoc->ExecuteQuery("select * from ugyfel;");  
    pDoc->MoveFirst();  
    for (int i=0; i<pDoc->Count(); i++){  
        pDoc->GetCurrentItem(uf);  
        CString str;  
        str.Format("%d",uf.Refszam());  
        pList->InsertItem(i,str);  
        pList->SetItemText(i,1,uf.Nev());  
        pList->SetItemText(i,2,uf.Cim());  
        pList->SetItemText(i,3,uf.Status());  
        pDoc->MoveNext();  
    }
```

„select \* from ugyfel order by refsza  
„select \* from ugyfel order by vnev  
„select \* from ugyfel order by knev  
„select \* from ugyfel order by status

Csak ezt a részt  
kell módosítani.

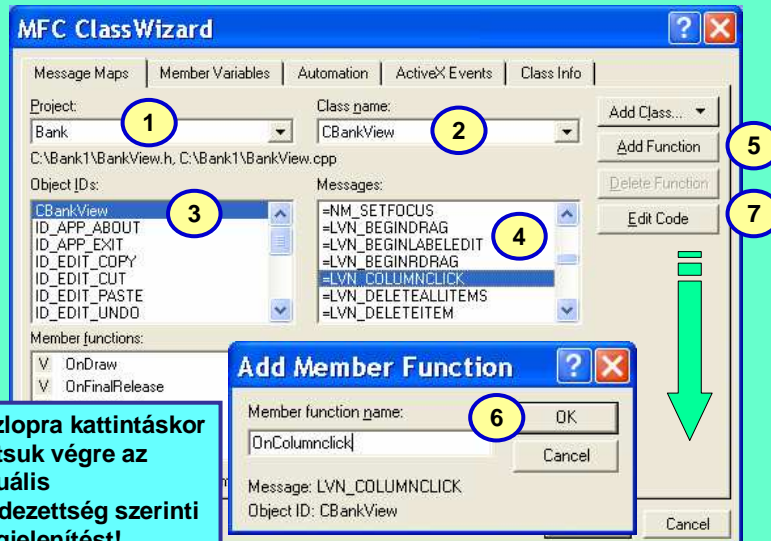
### Nézet osztály: FillSortedData(int sColumn)

```
void CBankView::FillSortedData(int sColumn) {
```

```
    Ugyfel uf;  
    CListCtrl* pList= &GetListCtrl();  
    CBankDoc* pDoc = GetDocument();  
    switch(sColumn) {  
        case 0:  
            pDoc->ExecuteQuery("select * from ugyfel order by refsza  
            break;  
        case 1:  
            pDoc->ExecuteQuery("select * from ugyfel order by nev;  
            break;  
        ...  
    }  
    pDoc->MoveFirst();  
    for (int i=0; i<pDoc->Count(); i++){  
        pDoc->GetCurrentItem(uf);  
        CString str;  
        str.Format("%d",uf.Refszam());  
        pList->InsertItem(i,str);  
        pList->SetItemText(i,1,uf.Nev());  
        ...  
        pDoc->MoveNext();  
    }
```



## Nézet osztály: OnColumnclick() - létrehozás



## Nézet osztály: OnColumnclick() - szerkesztés

BankView.cpp

```
void CBankView::OnColumnclick(NMHDR* pNMHDR, LRESULT* pResult)
{
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    // TODO: Add your control notification handler code here

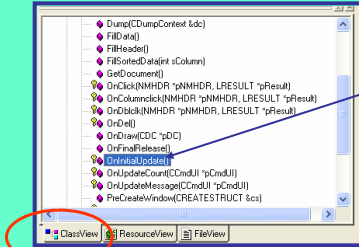
    GetListCtrl().DeleteAllItems();
    FillSortedData(pNMListView->iSubItem);

    *pResult = 0;
}
```

pNMListView

```
typedef struct tagNMLISTVIEW{
    NMHDR hdr;
    int item;
    int iSubItem;
    UINT uNewState;
    UINT uOldState;
    UINT uChanged;
    POINT ptAction;
    LPARAM lParam; } NMLISTVIEW, FAR *LPNMLISTVIEW;
```

## Nézet osztály: OnInitialUpdate() - módosítás



2x

A nézet osztály első megjelenítésekor a lista **refszam szerint rendezve** jelenjen meg.

BankView.cpp

```
void CBankView::OnInitialUpdate()
{
    CListView::OnInitialUpdate();
    // TODO: You may populate your ListView with items by directly accessing
    // its list control through a call to GetListCtrl().
    CBankDoc* pDoc = GetDocument();
    pDoc->ConnectDataBase("Bank");
    FillHeader();

    FillSortedData(0);
}
```

## Rendezettség nyilvántartása

### •Új adattag:

**sColumn**

Annak az oszlopnak a sorszáma, amelyre legutoljára rákattintottunk.

### •Módosítandó metódusok:

**OnColumnclick()**

Oszlopra kattintásnál aktualizálni kell az **sColumn** értékét

**OnInitialUpdate()**

Kezdetben legyen **sColumn** értéke 0.  
(Első oszlop szerint lesz rendezve a lista)

### Nézet osztály: sColumn

BankView.h

```
class CBankView : public CListView
{
    ...
protected:
    int sColumn;
    ...
}
```

### Nézet osztály: OnColumnclick() - módosítás

#### BankView.cpp

```
void CBankView::OnColumnclick(NMHDR* pNMHDR, LRESULT* pResult)
{
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    // TODO: Add your control notification handler code here

    GetListCtrl().DeleteAllItems();
    FillSortedData(pNMListView->iSubItem);

    sColumn=pNMListView->iSubItem;

    *pResult = 0;
}
```

## Nézet osztály: OnInitialUpdate() - módosítás

BankView.cpp

```
void CBankView::OnInitialUpdate()
{
    CListView::OnInitialUpdate();
    // TODO: You may populate your ListView with items by directly accessing
    // its list control through a call to GetListCtrl().
    CBankDoc* pDoc = GetDocument();
    pDoc->ConnectDataBase("Bank");
    FillHeader();
    FillSortedData(0);

    sColumn=0;
}
```

### 3. Feladat: Módosítás / Törlés / Beszúrás

A lista egy sorára duplát kattintva jelenjen meg egy párbeszédablak, amelyben módosíthatjuk az adatsort. Üres sorra duplát kattintva új adatot vihetünk fel. Módosításnál ne engedjük meg a kulcsmező (refszam) módosítását.

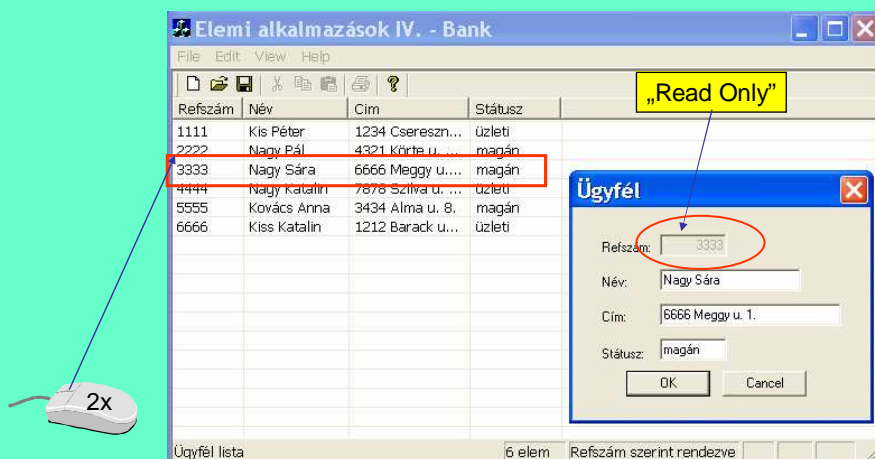


Refszám	Név	Cím	Státusz
1111	Kis Péter	1234 Cseresz...	üzleti
2222	Nagy Pál	4321 Körte u. ...	magán
3333	Nagy Sára	6666 Meggy u....	magán
4444	Nagy Katalin	7878 Szilva u. ...	üzleti
5555	Kovács Anna	3434 Alma u. 8.	magán
6666	Kiss Katalin	1212 Barack u...	üzleti

Ügyfél lista 6 elem Refszám szerint re

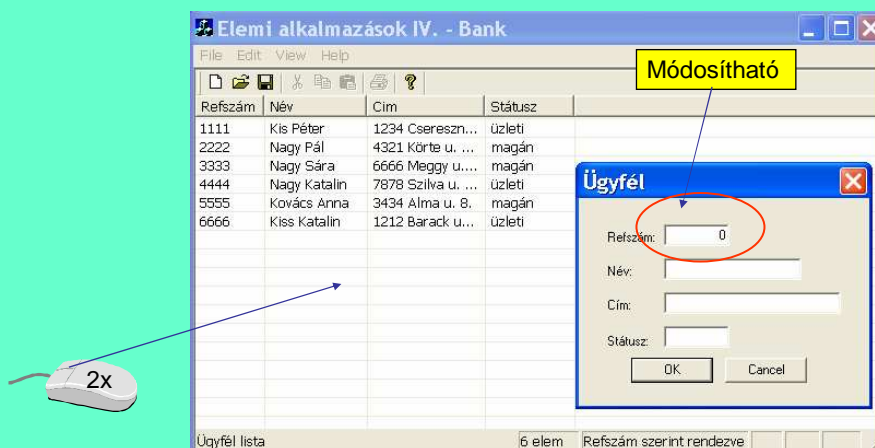
### Módosítás megvalósítása

A lista egy sorára duplát kattintva megjelenik egy párbeszédablak, amelyben módosíthatjuk az adatsort. Ilyenkor nem engedjük meg a kulcsmező (refszám) módosítását.



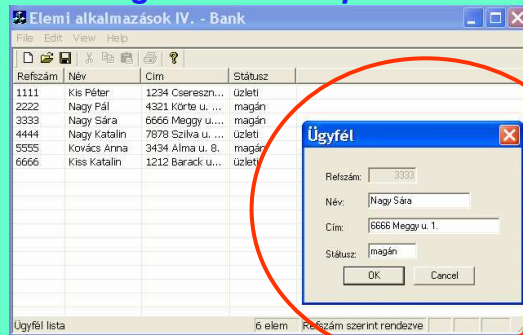
### Beszúrás megvalósítása

Üres sorra duplát kattintva új adatot vihetünk fel.





## A feladat megoldásának lépései



### ÜgyfelDlg osztály:

Párbeszédablak megtervezése

Vezérlőváltozók meghatározása

Saját adattagok bevezetése: enable\_refszam

Saját metódusok bevezetése:

EnableRefszam, DisableRefszam, SetUgyfel, GetUgyfel

### Nézet osztály (CBankView):

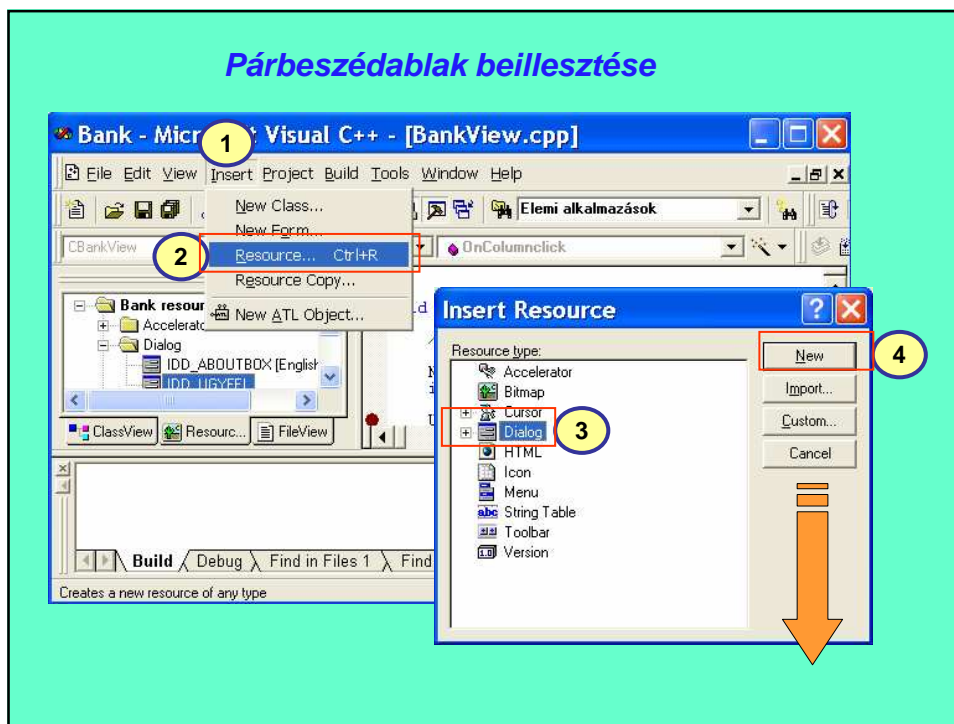
Dupla kattintás callback függvénye

Del+Ctrl callback függvénye

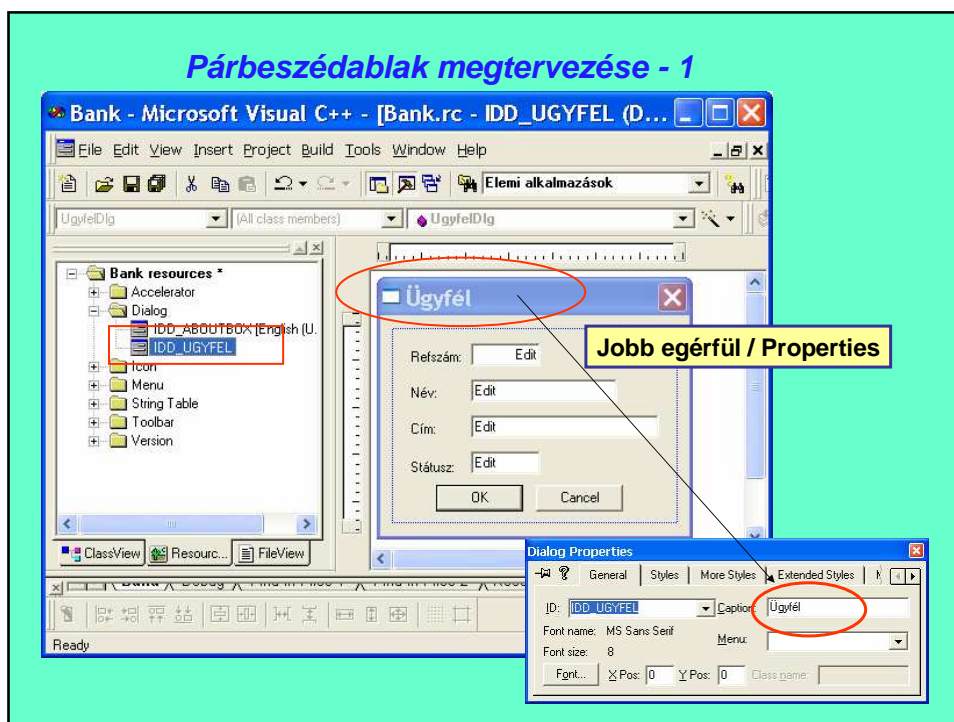
## ÜgyfelDlg osztály



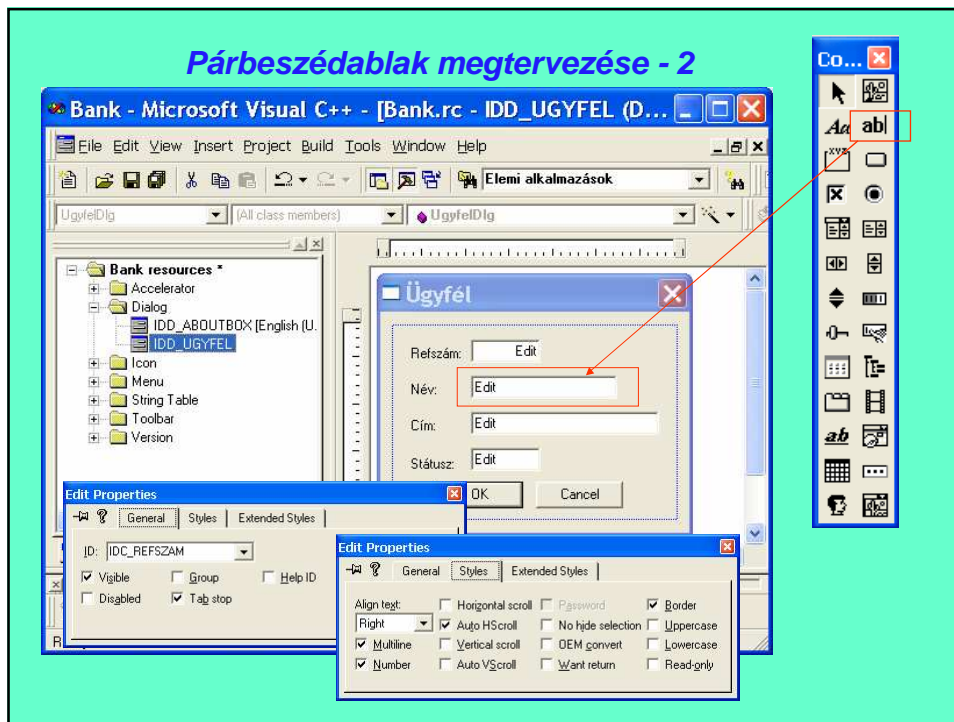
## Párbeszédablak beillesztése



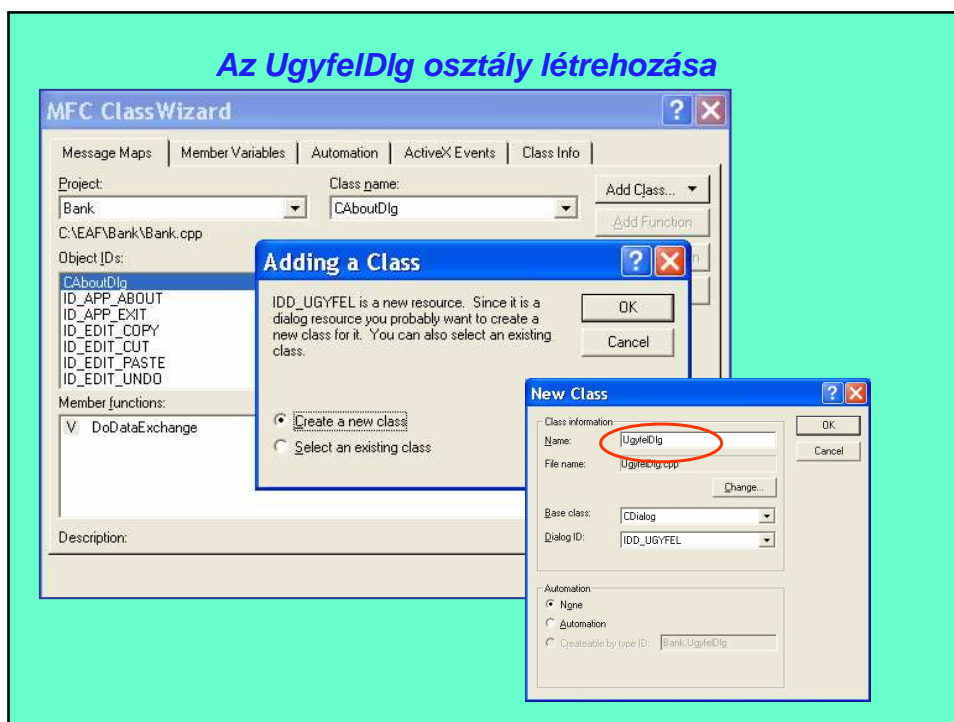
## Párbeszédablak megtervezése - 1



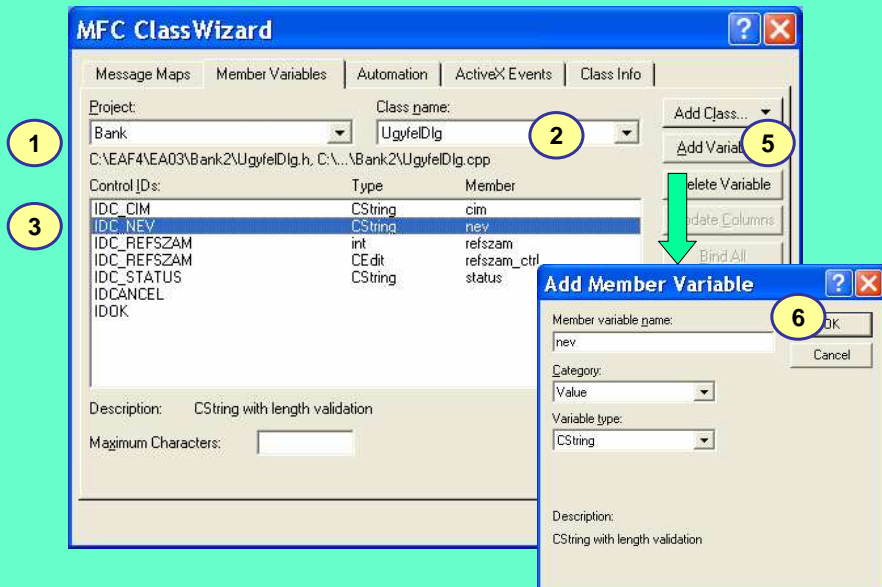
## Párbeszédablak megtervezése - 2



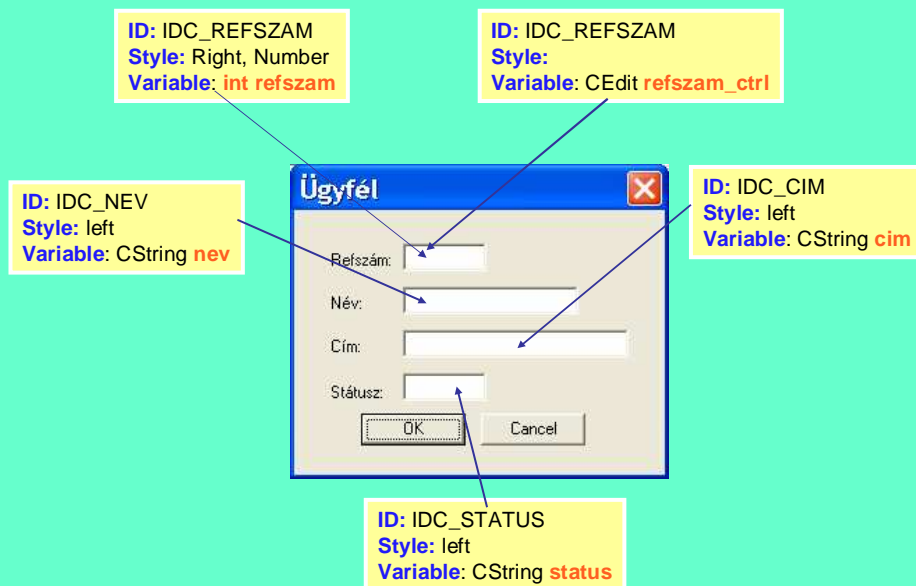
## Az UgyfelDlg osztály létrehozása



## UgyfelDlg: Változók hozzárendelése a vezérlőkhöz



## UgyfelDlg: Változók hozzárendelése a vezérlőkhöz



## Az UgyfelDlg osztály saját adatai és metódusai

### enable\_refszam - adattag

Logikai változó, amely jelzi, hogy módosítható-e a refszám.

### EnableRefszam()

Refszam módosítás engedélyezése

### DisableRefszam()

Refszam módosítás tiltása

### SetUgyfel(Ugyfel& uf);

A párbeszédablak vezérlőváltozóit kitölti a paraméterben megadott ügyfél adataival

### GetUgyfel(Ugyfel& uf);

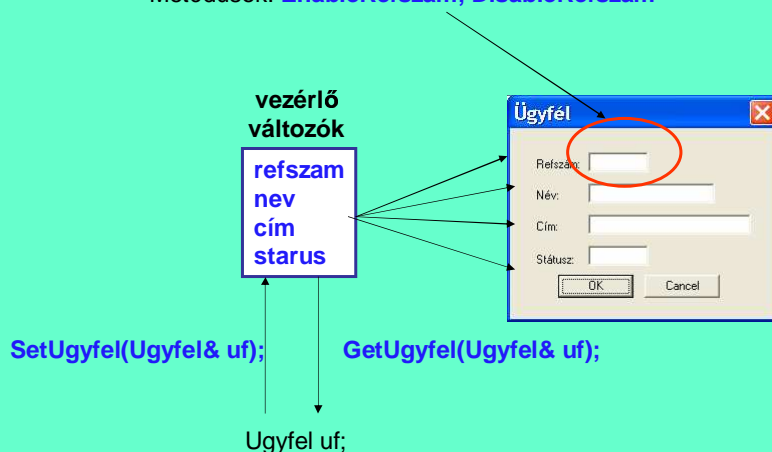
A párbeszédablak vezérlőváltozóit alapján kitölti a paraméterben megadott ügyfél adatait.

## UgyfelDlg: Saját adattag, saját metódus

### enable\_refszam

Logikai változó, amely jelzi, hogy módosítható-e a refszám

Metódusok: **EnableRefszam**, **DisableRefszam**



## UgyfelDlg: saját adattag – enable\_refszam

Ezzel a változóval azt jelezzük, hogy szabad-e módosítani a refszaot a párbeszédpanelen.

UgyfelDlg.h

protected:  
bool enable\_refszam;

## UgyfelDlg: EnableRefszam(), DisableRefszam()

UgyfelDlg.cpp

```
void UgyfelDlg::EnableRefszam()
```

```
{ enable_refszam=true;
```

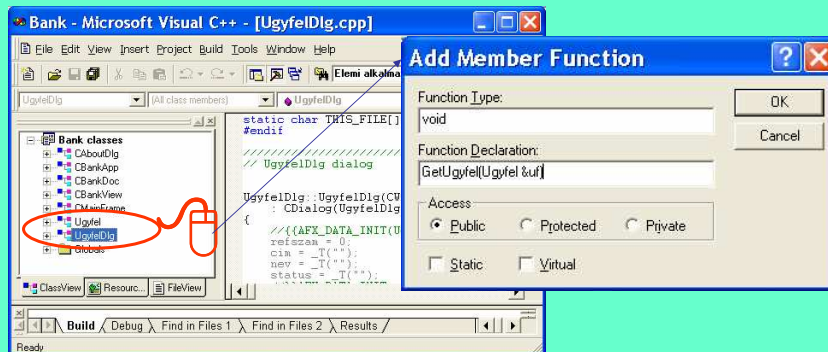


```
void UgyfelDlg::DisableRefszam()
```

```
{ enable_refszam=false;
```



## UgyfelDlg: saját metódus - GetUgyfel()



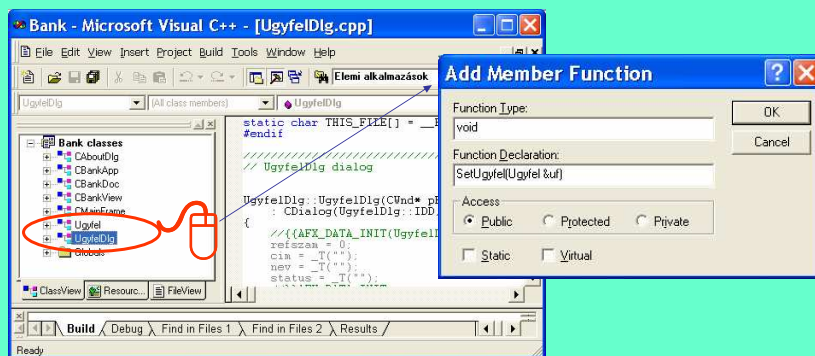
UgyfelDlg.cpp

```
void UgyfelDlg::GetUgyfel(Ugyfel &uf)
```

```
{
    uf.SetRefszam(refszam);
    uf.SetNev(nev);
    uf.SetCim(cim);
    uf.SetStatus(status);
}
```



## UgyfelDlg: saját metódus - SetUgyfel()



UgyfelDlg.cpp

```
void UgyfelDlg::SetUgyfel(Ugyfel &uf)
```

```
{
    refsza = uf.Refszam();
    nev = uf.Nev();
    cim = uf.Cim();
    status = uf.Status();
}
```





## UgyfelDlg: include

UgyfelDlg.h

```
#include "Ugyfel.h"
```

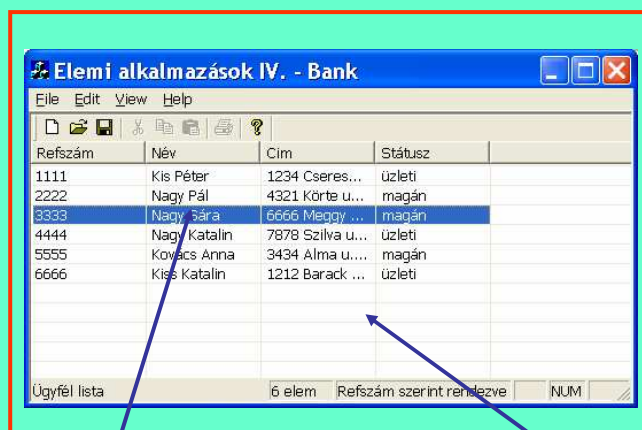
Az **Ugyfel.h** header fájlt illesszük be a **UgyfelDlg.h** header fájlba.

UgyfelDlg.cpp

```
void UgyfelDlg::GetUgyfel(Ugyfel &uf)
{
    uf.SetRefszam(refszam);
    uf.SetNev(nev);
    uf.SetCim(cim);
    uf.SetStatus(status);
}

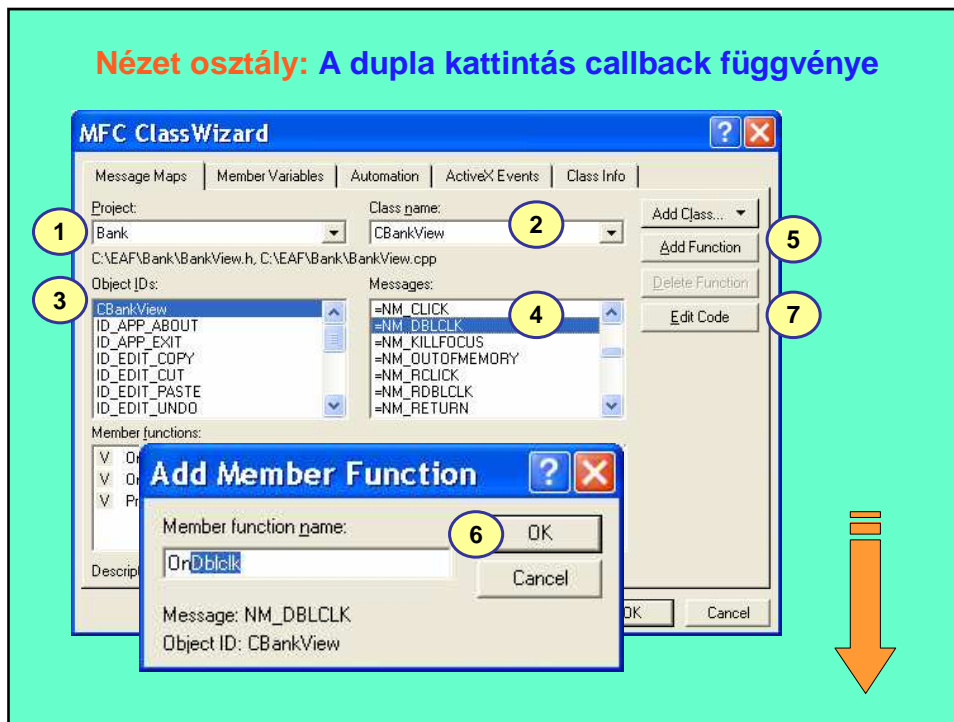
void UgyfelDlg::SetUgyfel(Ugyfel &uf)
{
    refszam = uf.Refszam();
    nev = uf.Nev();
    cim = uf.Cim();
    status = uf.Status();
}
```

## CBankView osztály





## Nézet osztály: A dupla kattintás callback függvénye



```

void CBankView::OnDbclick(NMHDR* pNMHDR, LRESULT* pResult) {
    // TODO: Add your control notification handler code here
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR
    int item = pNMListView->item;
    Ugyfel uf;
    UgyfelDlg dlg;
    CString str;
    if (item == -1) {
        uf.SetRefszam(0);
        uf.SetNev("");
        uf.SetCim("");
        uf.SetStatus("");
        dlg.EnableRefszam();
        bool single_key;
        single_key=false;
        while(single_key){
            single_key=true;
            dlg.SetUgyfel(uf);
            if (dlg.DoModal() == IDOK) {
                dlg.GetUgyfel(uf);
                str.Format("select * from ugyfel where refszam=%d", uf.Refszam());
                GetDocument()->ExecuteQuery(str);
                if (GetDocument()->Count()>0){
                    single_key=false;
                    MessageBox("Van már ilyen refszaám!");
                }
                str.Format("insert into ugyfel values ( %d,%s,%s,%s)", uf.Refszam(), uf.Nev(), uf.Cim(), uf.Status());
                GetDocument()->ExecuteQuery(str);
            }
        }
    }
    else {
        uf.SetRefszam(atoi(GetListCtrl().GetItemText(item,0)));
        uf.SetNev(GetListCtrl().GetItemText(item,1));
        uf.SetCim(GetListCtrl().GetItemText(item,2));
        uf.SetStatus(GetListCtrl().GetItemText(item,3));
        dlg.DisableRefszam();
        dlg.SetUgyfel(uf);
        if (dlg.DoModal() == IDOK) {
            dlg.GetUgyfel(uf);
            str.Format("update ugyfel set nev = '%s', cim = '%s', status = '%s' where refszam = %d", uf.Nev(), uf.Cim(), uf.Status(), uf.Refszam());
            GetDocument()->ExecuteQuery(str);
        }
    }
    GetListCtrl().DeleteAllItems();
    FillSortedData(sColumn);
    *pResult = 0;
}

```

**Előkészületek**

**Beszúrás**

**Módosítás**


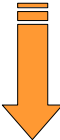
**Lista megjelenítése**

**Részletes bemutató**

## Nézet osztály: OnDblclk(. . .)

```
void CBankView::OnDblclk(NMHDR* pNMHDR, LRESULT* pResult)
{
    // TODO: Add your control notification handler code here
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    int item = pNMListView->item;



    Ugyfel uf;
    UgyfelDlg dlg;
    CString str;
    if (item == -1) {
        //Tennivalók, amikor nem listaelemre kattintottunk - beszúrás.
    }else{
        //Tennivalók, amikor a listára kattintottunk - módosítás
    }
    GetListCtrl().DeleteAllItems();
    FillSortedData(sColumn);
    *pResult = 0;
}
```

### OnDblclk :


#### Tennivalók, amikor nem listaelemre kattintottunk: beszúrás - 1

```
if (item == -1) {
    //Tennivalók, amikor nem listaelemre kattintottunk - beszúrás.
    uf.SetRefszam(0);
    uf.SetNev("");
    uf.SetCim("");
    uf.SetStatus("");
    dlg.EnableRefszam();
    bool single_key;
    single_key=false;
    while(!single_key){
        //Addig „kérdezzük”, amíg egyedi kulcsot nem kapunk
    }
}
else{
    //Tennivalók, amikor a listára kattintottunk - beszúrás
}
}
```

Inicializáljuk az uf  
lokális változót.

1



### OnDbclck :

#### Tennivalók, amikor nem listaelemre kattintottunk: beszúrás - 2

```
if (iltem == -1) {  
    //Tennivalók, amikor nem listaelemre kattintottunk - beszúrás.  
    uf.SetRefszam(0);  
    uf.SetNev("");  
    uf.SetCim("");  
    uf.SetStatus("");  
    dlg.EnableRefszam();  
    bool single_key;  
    single_key=false;  
    while(!single_key){  
        //Addig „kérdezzük”, amíg egyedi kulcsot nem kapunk  
    }  
}else{  
    //Tennivalók, amikor a listára kattintottunk - beszúrás  
}
```

Engedélyezzük a refsza módosítását.

2

### OnDbclck :

#### Tennivalók, amikor nem listaelemre kattintottunk: beszúrás - 3

```
if (iltem == -1) {  
    //Tennivalók, amikor nem listaelemre kattintottunk - beszúrás.  
    uf.SetRefszam(0);  
    uf.SetNev("");  
    uf.SetCim("");  
    uf.SetStatus("");  
    dlg.EnableRefszam();  
    bool single_key;  
    single_key=false;  
    while(!single_key){  
        //Addig „kérdezzük”, amíg egyedi kulcsot nem kapunk  
    }  
}else{  
    //Tennivalók, amikor a listára kattintottunk - beszúrás  
}
```

A „0” nem egyedi kulcs.

3

**OnDbclck :**  
**Addig „kérdezgetünk”, amíg egyedi kulcsot nem kapunk: beszúrás - 4**

```

single_key=false;
while(!single_key){
    //Addig „kérdezgetünk”, amíg egyedi kulcsot nem kapunk
    single_key=true;
    dlg.SetUgyfel(uf);
    if (dlg.DoModal() == IDOK) {
        dlg.GetUgyfel(uf);
        str.Format("select * from ugyfel where refsza=%d;", uf.Refszam());
        GetDocument()->ExecuteQuery(str);
        if (GetDocument()->Count()>0){
            single_key=false;
            MessageBox("Van már ilyen refsza!");
        }else{
            str.Format("insert into ugyfel
                values ( %d,'%s','%s','%s');", uf.Refszam(), uf.Nev(), uf.Cim(),
                uf.Status());
            GetDocument()->ExecuteQuery(str);
        }
    }
}

```

A párbeszédpanel kezdeti értékeinek beállítása.

Tennivalók, miután a párbeszédpanelen az OK gombra kattintottak.

**OnDbclck :**  
**Addig „kérdezgetünk”, amíg egyedi kulcsot nem kapunk: beszúrás - 5**

```

single_key=false;
while(!single_key){
    single_key=true;
    dlg.SetUgyfel(uf);
    if (dlg.DoModal() == IDOK) {
        ① dlg.GetUgyfel(uf);
        ② str.Format("select * from ugyfel where refsza=%d;", uf.Refszam());
        ③ GetDocument()->ExecuteQuery(str);
        if (GetDocument()->Count()>0){
            single_key=false;
            MessageBox("Van már ilyen refsza!");
        }else{
            str.Format("insert into ugyfel
                values ( %d,'%s','%s','%s');", uf.Refszam(), uf.Nev(), uf.Cim(),
                uf.Status());
            GetDocument()->ExecuteQuery(str);
        }
    }
}

```

Válogassuk ki a dialógusban megadott refsza értékű adatokat az adatbázisból!

**OnDbclck :**  
Addig „kérdezzük”, amíg egyedi kulcsot nem kapunk: **beszúrás - 6**

```

single_key=false;
while(!single_key){
    single_key=true;
    dlg.SetUgyfel(uf);

    if (dlg.DoModal() == IDOK) {
        dlg.GetUgyfel(uf);
        str.Format("select * from ugyfel where refsza=%d", uf.Refszam());
        GetDocument()->ExecuteQuery(str);
        if (GetDocument()->Count()>0){
            ① single_key=false;
            ② MessageBox("Van már ilyen refsza!");
        }else{
            str.Format("insert into ugyfel
                values ( %d,'%s','%s','%s');", uf.Refszam(), uf.Nev(), uf.Cim(),
                    uf.Status());
            GetDocument()->ExecuteQuery(str);
        }
    }
}

```

Ha van már ilyen refsza, akkor a single\_key változót állítsuk hamisra és küldjünk figyelmeztető üzenetet.

**OnDbclck :**  
Addig „kérdezzük”, amíg egyedi kulcsot nem kapunk: **beszúrás - 7**

```

single_key=false;
while(!single_key){
    single_key=true;
    dlg.SetUgyfel(uf);

    if (dlg.DoModal() == IDOK) {
        dlg.GetUgyfel(uf);
        str.Format("select * from ugyfel where refsza=%d", uf.Refszam());
        GetDocument()->ExecuteQuery(str);
        if (GetDocument()->Count()>0){
            single_key=false;
            MessageBox("Van már ilyen refsza!");
        }else{
            ① str.Format("insert into ugyfel
                values ( %d,'%s','%s','%s');", uf.Refszam(), uf.Nev(), uf.Cim(),
                    uf.Status());
            ② GetDocument()->ExecuteQuery(str);
        }
    }
}

```

Ha a párbeszédpanelen megadott refszámmal nincs még adatunk, akkor ezt az adatot szúrjuk be az adatbázisba.

## Nézet osztály: OnDblclk(. . .)

```
void CBankView::OnDblclk(NMHDR* pNMHDR, LRESULT* pResult)
{
    // TODO: Add your control notification handler code here
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    int item = pNMListView->item;

    Ugyfel uf;
    UgyfelDlg dlg;
    CString str;
    if (item == -1) {
        //Tennivalók, amikor nem listaelemre kattintottunk - beszúrás.
    }
    else {
        //Tennivalók, amikor a listára kattintottunk - módosítás
        GetListCtrl().DeleteAllItems();
        FillSortedData(sColumn);
        *pResult = 0;
    }
}
```

✓

↓

## OnDblclk :

### tennivalók, amikor listaelemre kattintottunk: módosítás - 1

```
void CBankView::OnDblclk(NMHDR* pNMHDR, LRESULT* pResult)
{
    // TODO: Add your control notification handler code here
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    int item = pNMListView->item;
}

else {
    uf.SetRefszam(atoi(GetListCtrl().GetItemText(item,0)));
    uf.SetNev(GetListCtrl().GetItemText(item,1));
    uf.SetCim(GetListCtrl().GetItemText(item,2));
    uf.SetStatus(GetListCtrl().GetItemText(item,3));
    dlg.DisableRefszam();
    dlg.SetUgyfel(uf);
    if (dlg.DoModal() == IDOK) {
        dlg.GetUgyfel(uf);
        str.Format("update ugyfeset nev = '%s', cim = '%s', status = '%s' where refsza = %d;", uf.Nev(), uf.Cim(), uf.Status(), uf.Refszam());
        GetDocument()->ExecuteQuery(str);
    }
}
```

↓

Töltsük ki az uf lokális változót a kiválasztott sor elmeivel.

GetListCtrl()

Refszám	Név	Cím	Státusz
1111	Kis Péter	1234 Cseres...	üzleti
2222	Nagy Pál	4321 Körte u...	magán
3333	Nagy Sára	6666 Meggy ...	magán
4444	Nagy Katalin	7878 Szilva u...	üzleti
5555	Kovács Anna	3434 Alma u....	magán
6666	Kiss Katalin	1212 Barack ...	üzleti

Ügyfél lista 6 elem Refszám szerint rend

**uf.SetNev(GetListCtrl().GetItemText(iltem,1));**

#### OnDblick :

tennivalók, amikor listaelemre kattintottunk: **módosítás - 2**

```

}else{
    uf.SetRefszam(atoi(GetListCtrl().GetItemText(iltem,0)));
    uf.SetNev(GetListCtrl().GetItemText(iltem,1));
    uf.SetCim(GetListCtrl().GetItemText(iltem,2));
    uf.SetStatus(GetListCtrl().GetItemText(iltem,3));
    dlg.DisableRefszam();
    dlg.SetUgyfel(uf);
    if (dlg.DoModal() == IDOK) {
        dlg.GetUgyfel(uf);
        str.Format("update ugyfeset nev = '%s', cim = '%s', status = '%s'
            where refsza = %d;", uf.Nev(), uf.Cim(), uf.Status(), uf.Refszam());
        GetDocument()->ExecuteQuery(str);
    }
}

```

Tiltsuk meg a refsza módosítását.

**OnDbclck :**  
**tennivalók, amikor listaelemre kattintottunk: módosítás - 3**

```

}else{
    uf.SetRefszam(atoi(GetListCtrl().GetItemText(iltem,0)));
    uf.SetNev(GetListCtrl().GetItemText(iltem,1));
    uf.SetCim(GetListCtrl().GetItemText(iltem,2));
    uf.SetStatus(GetListCtrl().GetItemText(iltem,3));
    dlg.DisableRefszam();
    dlg.SetUgyfel(uf);
    if (dlg.DoModal() == IDOK) {
        dlg.GetUgyfel(uf);
        str.Format("update ugyfeset nev = '%s', cim = '%s', status = '%s'
            where refszam = %d;", uf.Nev(), uf.Cim(), uf.Status(), uf.Refszam());
        GetDocument()->ExecuteQuery(str);
    }
}
    
```

Az OK gomb  
leütésekor módosítsuk  
az adatokat az  
adatbázisban.

**Módosításnál „csak olvasható” a refsza**: OnInitDialog()

The image shows two screenshots of the MFC ClassWizard dialog box, illustrating the steps to add a message handler for `WM_INITDIALOG`.

**Left Screenshot (Step 1-5):**

- 1: Project: Bank
- 2: Class name: UgyfelDlg
- 3: Object IDs: UgyfelDlg
- 4: Messages: WM\_INITDIALOG
- 5: Add Function button

**Right Screenshot (Step 6-7):**

- 6: Member functions: OnInitDialog
- 7: OK button

A blue box labeled **WM\_INITDIALOG** points to the message in the right screenshot.



### Párbeszéd ablak: OnInitDialog()

**GetDlgItem(IDC\_REFSZAM)**



```
BOOL UgyfelDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    // TODO: Add extra initialization here
```



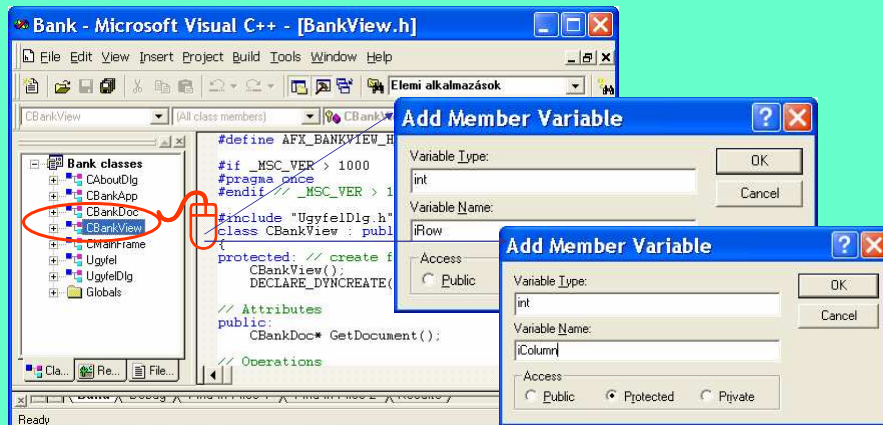
```
    if (enable_refszam)
        GetDlgItem(IDC_REFSZAM)->EnableWindow(TRUE);
    else
        GetDlgItem(IDC_REFSZAM)->EnableWindow(FALSE);
```

```
    return TRUE; // return TRUE unless you set the focus to a control
    // EXCEPTION: OCX Property Pages should return FALSE
}
```

### Ctrl+Del billentyűre töröljük a sort

- Felveszünk két adattagot : iRow, iColumn
- Az OnClick(...) esemény bekövetkezésekor aktualizáljuk a kattintás helyét
- Kezeljük a Ctrl+Del eseményt
  - Azonosítót rendelünk a tevékenységhez (ID\_DEL)
  - Az azonosítóhoz hozzárendeljük a Ctrl+DEL billentyűt
  - Az azonosítóhoz hozzárendeljük a callback függvényt

## Nézet osztály: saját adattag - iRow, iColumn



BankView.h

```
protected:
    int iRow;
    int iColumn;
```

## Nézet osztály: OnClick(...)



## Nézet osztály: *OnClick(...)*

```
void CBankView::OnClick(NMHDR* pNMHDR, LRESULT* pResult)
{
    // TODO: Add your control notification handler code here

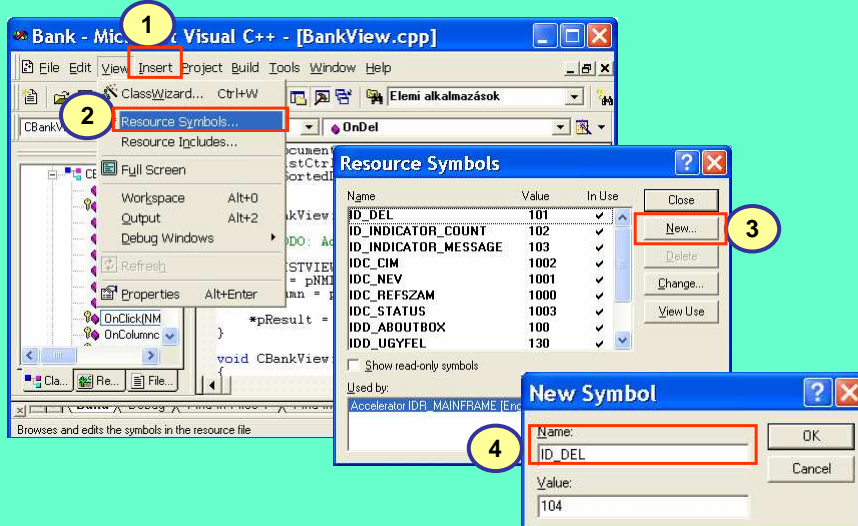
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;

    iRow = pNMListView->iItem;
    iColumn = pNMListView->iSubItem;

    *pResult = 0;
}
```



## Az ID\_DEL azonosító megadása



## A Ctrl+DEL és ID\_DEL összekapcsolása

The screenshot shows the Microsoft Visual C++ IDE with the 'Bank - Microsoft Visual C++ - [Bank.rc - IDR\_MAINFRAME...]' window. The 'Bank resources' tree on the left has 'Accelerator' expanded, and 'IDR\_MAINFRAME (English (U.S.))' is selected. The 'ID' list on the right shows various IDs, with 'ID\_DEL' selected. A context menu is open over 'ID\_DEL', and the 'Accel Properties' dialog is shown in the foreground. The dialog has 'ID: ID\_DEL' and 'Key: VK\_DELETE'. The 'Modifiers' section has 'Ctrl' checked. The 'Type' section has 'ASCII' selected. The 'Next Key Typed' button is visible. Numbered callouts 1 through 4 indicate the steps: 1. Select 'Bank resources', 2. Select 'Accelerator', 3. Select 'ID\_DEL', 4. Click 'New Accelerator'.

## Az ID\_DEL callback függvénye - beillesztés

The screenshot shows the MFC ClassWizard dialog with the 'Messages' tab selected. The 'Project' is 'Bank' and the 'Class name' is 'CBankView'. The 'Object IDs' list on the left has 'ID\_DEL' selected. The 'Messages' list on the right has 'COMMAND' selected. The 'Add Member Function' dialog is shown in the foreground. The dialog has 'Member function name: OnDel' and 'Message: COMMAND'. The 'Object ID: ID\_DEL' is also visible. Numbered callouts 1 through 7 indicate the steps: 1. Select 'Bank', 2. Select 'CBankView', 3. Select 'ID\_DEL', 4. Select 'COMMAND', 5. Click 'Add Function...', 6. Enter 'OnDel', 7. Click 'OK'.

## Az ID\_DEL callback függvénye - szerkesztés

```
void CBankView::OnDel()
{
    // TODO: Add your command handler code here

    int refszaam = atoi(GetListCtrl().GetItemText(iRow,0));
    CString str;
    str.Format("delete from ugyfel where refszaam = %d ;",refszaam);
    GetDocument()->ExecuteQuery(str);
    GetListCtrl().DeleteAllItems();
    FillSortedData(sColumn);
}
```

## 4.feladat: Új indikátorok bevezetése a státuszsorban

Refszám	Név	Cím	Státusz
1111	Kis Péter	1234 Cseresznye u. 7.	üzleti
2222	Nagy Pál	4321 Körte u. 13.	magán
3333	Nagy Sára	6666 Meggy u. 1.	magán
4444	Nagy Katalin	7878 Szilva u. 123.	üzleti
5555	Kovács Anna	3434 Alma u. 8.	magán
6666	Kiss Katalin	1212 Barack u. 7.	üzleti

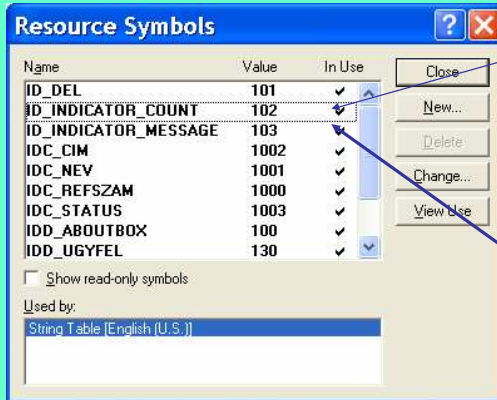
Ügyfél lista      6 elem      Refszám szerint reni

Hány elemű a lista?

Hogyan van rendezve?

## Új indikátor azonosítók bevezetése

View/Resource symbols/New ...



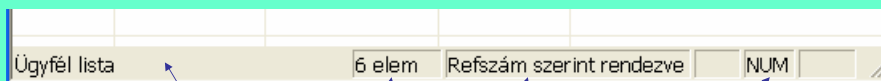
Hány elemű a lista?

**ID\_INDICATOR\_COUNT**

Hogyan van rendezve?

**ID\_INDICATOR\_MESSAGE**

## Az új indikátorok helyének meghatározása



### Mainfrm.cpp

```
static UINT indicators[] =
{
    ID_SEPARATOR,           // status line indicator
    ID_INDICATOR_COUNT,
    ID_INDICATOR_MESSAGE,
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,};
```

## Az új indikátorok tartalmának megadása

Bank - Microsoft Visual C++ - [Bank.rc - String Table ...]

File Edit View Insert Project Build Tools Window Help

Elemi alkalmazások

CBankView [All class members] OnDel

Bank resources

- Accelerator
- Dialog
- Icon
- Menu
- String Table
- String Table [English (U.S.)]
- Toolbar
- Version

1

2x

2

3

1x jobb

4

5

ID	Value	Caption
ID_INDICATOR_COUNT	102	xxx elem
ID_INDICATOR_MESSAGE		
IDR_MAINFRAME		
AFX_IDS_APP_TITLE		
AFX_IDS_IDLEMESSAGE		
ID_FILE_NEW		new docume
ID_FILE_OPEN		existing doc
ID_FILE_CLOSE		active d
ID_FILE_SAVE		active d
ID_FILE_SAVE_AS		
ID_FILE_MRU_FILE1	57616	Open this document
ID_FILE_MRU_FILE2	57617	Open this document
ID_FILE_MRU_FILE3	57618	Open this document
ID_FILE_MRU_FILE4	57619	Open this document
ID_FILE_MRU_FILE5	57620	Open this document
ID_FILE_MRU_FILE6	57621	Open this document

String Properties

General

ID: INDICATOR\_MESSAGE

Caption: Retszám szerint rendezve

## Új indikátorok callback függvénye - prototípus

### BankView.h

// Generated message map functions

protected:

```
int iColumn;
int iRow;
int sColumn;
//{{AFX_MSG(CBankView)
afx_msg void OnColumnClick(NMHDR* pNMHDR, LRESULT* pResult);
afx_msg void OnDblclk(NMHDR* pNMHDR, LRESULT* pResult);
afx_msg void OnDel();
afx_msg void OnClick(NMHDR* pNMHDR, LRESULT* pResult);
//}}AFX_MSG
```


```
afx_msg void OnUpdateMessage(CCmdUI *pCmdUI);
afx_msg void OnUpdateCount(CCmdUI *pCmdUI);
```



```
DECLARE_MESSAGE_MAP()
```


## Új indikátorok callback függvénye – üzenettábla

```
BEGIN_MESSAGE_MAP(CBankView, CListView)
//{{AFX_MSG_MAP(CBankView)
ON_NOTIFY_REFLECT(LVN_COLUMNCLICK, OnColumnclick)
ON_NOTIFY_REFLECT(NM_DBLCLK, OnDbclick)
ON_COMMAND(ID_DEL, OnDel)
ON_NOTIFY_REFLECT(NM_CLICK, OnClick)
//}}AFX_MSG_MAP
ON_UPDATE_COMMAND_UI(ID_INDICATOR_MESSAGE, OnUpdateMessage)
ON_UPDATE_COMMAND_UI(ID_INDICATOR_COUNT, OnUpdateCount)
END_MESSAGE_MAP()
```



## Az ID\_INDICATOR\_MESSAGE callback függvénye – OnUpdateMessage(...)

```
void CBankView::OnUpdateMessage(CCmdUI *pCmdUI)
{
    pCmdUI->Enable();
    switch(sColumn){
        case 0:
            pCmdUI->SetText("Refszám szerint rendezve");
            break;
        case 1:
            pCmdUI->SetText("Név szerint rendezve");
            break;
        case 2:
            pCmdUI->SetText("Cím szerint rendezve");
            break;
        case 3:
            pCmdUI->SetText("Státusz szerint rendezve");
            break;
    }
}
```





**Az ID\_INDICATOR\_COUNT callback függvénye –  
OnUpdateCount(...)**

```
void CBankView::OnUpdateCount(CCmdUI *pCmdUI)
{
    pCmdUI->Enable();
    int i = GetListCtrl().GetItemCount();
    CString str;
    str.Format("%d elem", i);
    pCmdUI->SetText(str);
}
```



**Vége**