

ELEMI ALKALMAZÁSOK FEJLESZTÉSE II. Újrafelhasználás

Készítette: Gregorics Tibor
a Gábor Dénes Főiskola Objektum orientált
programozás című tantárgy vizsgafeladatai
alapján.

Tartalom



- Egy ablakozó könyvtár megismerése
- Feladatok megoldása újrafelhasználással

Egy könyvtár

- Tároló osztályok
 - Egy őselemosztály
 - Általános tároló osztály
 - Rendezett tároló osztály
- Ablakozó osztályok
 - Adat beolvasó ablak osztály
 - Görgető ablak osztály
 - Fájl tartalmát görgető ablak osztály
- Billentyűzet kezelés
 - Leütött billentyűt beolvasó függvény
 - Billentyű konstansok

ulist.h

```
#ifndef ULIST_H
#define ULIST_H

class Titem {...};
class TList : public Titem {...};
class TorderedList : public TList {...};

#endif
```

ulist.h

Őselem osztály

```
class Titem {
public:
    Titem();
    virtual ~Titem();
    Titem* Next() { return FNext; };
    ...
private:
    Titem* FNext;
    friend class TList;
    friend class TorderedList;
};
```

ulist.h

Tároló osztály

```
class TList : public Titem {
public:
    TList();

    virtual void Insert(Titem* item);
    bool Exists(Titem* item);
    void Delete(Titem* item);
    void DeleteAll();

    bool Empty();
    Item* First();
    Item* RNext(Titem* item); // ciklikusan
    void Each(TitemProcedure todo);
    ...
    virtual ~TList(); // DeleteAll()
};
```

ulist.h **typedef void (*TitemProcedure)(Titem* item);**

Rendezett tároló osztály

```
class TOrderedList: public TList{
public:
    TOrderedList(){};

    void Insert(TItem* item);    // Ordered()
    virtual bool Ordered(TItem* item1, TItem* item2)
        = 0;
};
```

ulist.h

uwin.h

```
#ifndef UWIN_H
#define UWIN_H

namespace Color{
    const int white = ...;
    const int black = ...;
    const int gray = ...;
    const int cyan = ...;
    ...
}

class TWindow          : public TItem {...};
class TFrameWindow     : public TWindow {...};
class TTitledWindow    : public TFrameWindow {...};

class TReadWindow      : public TTitledWindow {...};
class TScrollWindow    : public TTitledWindow {...};
class TScrollFileWindow : public TScrollWindow {...};
```

uwin.h

Általános ablak osztály

```
class TWindow : public TItem{
public:
    TWindow(int x, int y,
            int w, int h, int bc, int pc);
    virtual ~TWindow();
    ...
    virtual void Show();
    void Hide();
    void SetColor(int bc, int pc);
protected:
    int xcoord, ycoord, width, height;
    int backcolor, pencolor;
    int cursorx, cursory;
};
```

uwin.h

Keretes és fejléces-keretes ablak

```
class TFrameWindow: public TWindow{
    ...
    void SetFrame(...);
};

class TTitledWindow: public TFrameWindow{
public:
    TTitledWindow(string str, int x, int y,
                  int w, int h, int bc, int pc);
    ...
    virtual string GetTitle();
    virtual void SetTitle(string str);
};
```

uwin.h

Adat beolvasó ablak osztály

```
class TReadWindow: public TTitledWindow{
public:
    TReadWindow(string str, int x, int y,
                int w, int h, int bc, int pc);
    ...
    virtual char ReadChar();
    virtual double ReadNum();
    virtual string ReadString();
};
```

uwin.h

Görgető ablak osztály

```
class TScrollWindow: public TTitledWindow{
public:
    TScrollWindow(string str, int x, int y,
                  int w, int h, int bc, int pc);
    ...
    virtual void Show();    // FLines[] sorai
    virtual void ScrollUp();
    virtual void ScrollDown();
    void Insert(string& str);
protected:
    int FLinesNum;
    int FLines[size];    // size=200 konstans
    ...
};
```

uwin.h

Fájl tartalmát görgető ablak osztály

```
class TScrollFileWindow: public TScrollWindow {
public:
    TScrollFileWindow(string str, int x, int y,
                      int w, int h, int bc, int pc);
    ...
private:
    string FFileName;
};
```

uwin.h

ukey.h

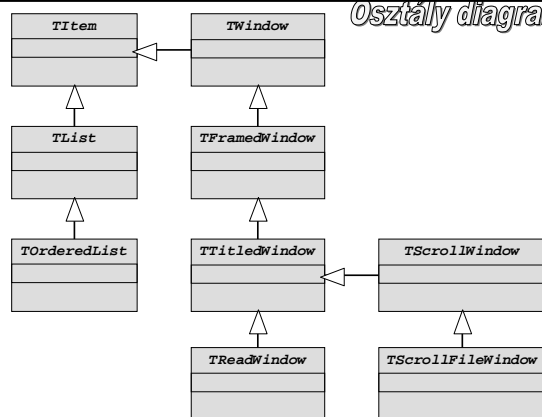
```
#ifndef UKEY_H
#define UKEY_H
namespace Key{
    const int  esc    = ...;
    const int  f1     = ...;
    const int  f2     = ...;
    ...
    const int  ins    = ...;
    const int  tab    = ...;

    const int  up     = ...;
    const int  down   = ...;

    int GetKey(void);
}
#endif
```

ukey.h

Osztály diagram



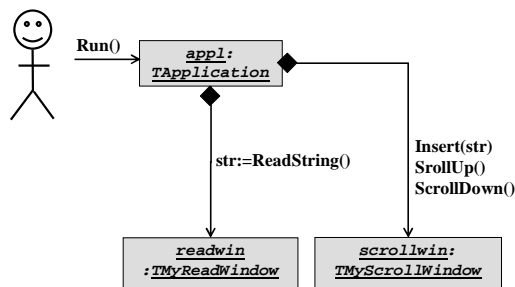
1. Feladat

A képernyő felső felét lefedi egy szürke-kék színű keretes görgetőablak, mely kezdetben üres. Ezt az ablakot paraméter nélkül inicializáljuk.

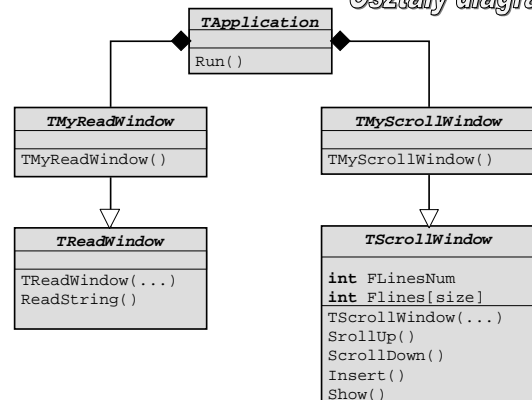
Ezután a program a következő billentyűkre reagál:

- ◆ Ins gomb: Megjelenik egy kék-fekete, 'Név' című ablak a képernyő alsó felén, melyből egy nevet kérünk be. Beolvasás után az ablak eltűnik a képernyőről. A beolvasott név megjelenik a görgetőablakban, utolsó névként.
- ◆ Kurzor fel, illetve le: A görgetőablakban levő neveket fel, illetve fel lehet görgetni.
- ◆ Az Esc-re tűnjön el a görgető ablak, és kilépés.

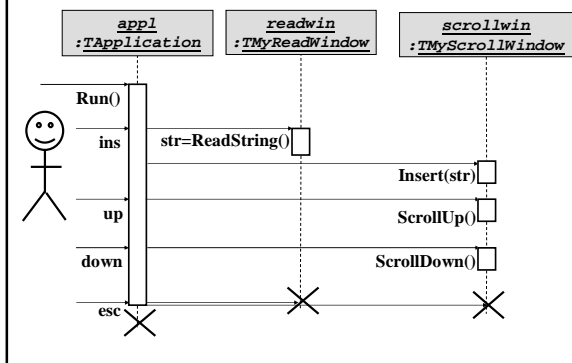
Objektum és együttműködési diagram



Osztály diagram



Szekvencia diagram



Esemény kezelő

```

void TApplication::Run()
{
    int key;
    for(;;){
        key = GetKey();
        switch (key){
            case ins : scrollwin.Insert(
                        readwin.ReadString());
                        break;
            case up : scrollwin.ScrollUp();
                        break;
            case down: scrollwin.ScrollDown();
                        break;
            case esc : return;
        }
    }
}
main.cpp

```

TApplication

```

class TApplication{
protected:
    TMyReadWindow readwin;
    TMyScrollWindow scrollwin;
public:
    void Run();
};

```

main.cpp

TMyWindow osztályok

```

class TMyScrollWindow: public TScrollWindow{
public:
    TMyScrollWindow()
        :TScrollWindow("Görgető",1,1,80,5,gray,black)
    {
        Show();
    }
};

class TMyReadWindow: public TReadWindow{
public:
    TMyReadWindow()
        :TReadWindow("Beolvasó",1,12,80,5,cyan,black){}
};

```

main.cpp

TApplication másképp

```

class TApplication{
protected:
    TMyReadWindow *readwin;
    TMyScrollWindow *scrollwin;
public:
    TApplication(){
        readwin = new TMyReadWindow;
        scrollwin = new TMyScrollWindow;
    }
    void Run();
    ~TApplication(){
        delete readwin;
        delete scrollwin;
    }
};

```

main.cpp

Főprogram

```

#include "ukey.h"
#include "uwin.h"
using namespace Key;
using namespace Color;

class TMyReadWindow;
class TMyScrollWindow;
class TApplication;

int main()
{
    TApplication appl;
    appl.Run();
}

```

main.cpp

2. Feladat



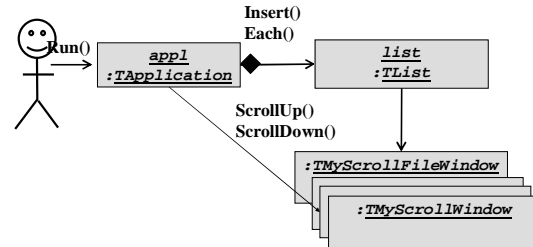
A képernyőn két féle, 20*5 méretű ablak található:

- ♦ az egyikben egy megadott állományt lehet fel-le görgetni.
- ♦ a másikban a következő nevek görgethetők:
Aladár, Béla, Gyula, Kázmér, Jenő.

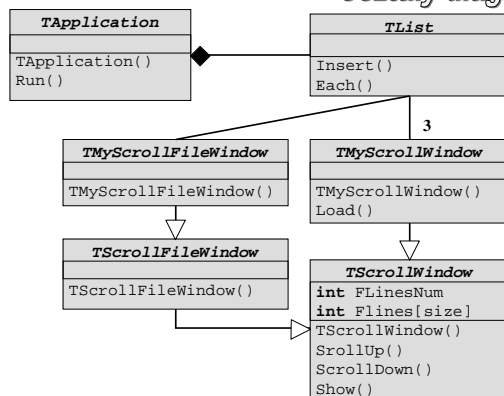
A program indulásakor jelenjen meg egy ablak az első fajtából, három pedig a második fajtából.

- ♦ A fel-le kurzorvezérlő billentyűkkel az ablakokban levő szövegeket egyszerre lehessen görgetni!
- ♦ Az Esc-re tűnjenek el az ablakok, és kilépés.

Objektum és együttműködési diagram



Osztály diagram



Főprogram

Top-Down

```

#include "ukey.h"
#include "uwin.h"
#include "ulist.h"
using namespace Key;
using namespace Color;

class TMyReadWindow ;
class TMyScrollFileWindow ;
class TApplication ;

int main()
{
    TApplication appl;
    appl.Run();
}

```

main.cpp

TApplication

Top-Down

```

class TApplication{
protected:
    TList list;
public:
    TApplication();
    void Run();
};

```

main.cpp

Konstruktor

Top-Down

```

TApplcation::TApplcation()
{
    list.Insert(new TMyScrollFileWindow
        ("input.txt",1,1,red,white));

    for(int i = 1; i<=3; i++){
        list.Insert(new TMyScrollWindow
            (1,i*6+1,blue,white));
    }
}

```

main.cpp

TMy..Window osztályok

Top-Down

```
class TMyScrollFileWindow: public TScrollFileWindow{
public:
    TMyScrollFileWindow
        (string str,int x,int y,int bc,int pc)
        :TScrollFileWindow(str,x,y,80,6,bc,pc){Show();}
};

class TMyScrollWindow: public TScrollWindow{
public:
    TMyScrollWindow
        (int x,int y,int bc,int pc)
        :TScrollWindow("",x,y,80,6,bc,pc){Load();Show();}
protected:
    void Load();
};
```

main.cpp

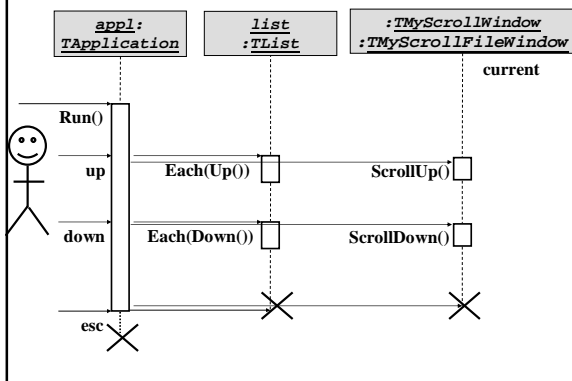
TMyScrollWindow::Load()

Top-Down

```
void TMyScrollWindow::Load()
{
    FLinesNum = 5;
    FLines[0] = "Aladár";
    FLines[1] = "Béla";
    FLines[2] = "Gyula";
    FLines[3] = "Kázmér";
    FLines[4] = "Jenő";
}
```

main.cpp

Szekvencia diagram



Esemény kezelő

Top-Down

```
void TApplication::Run()
{
    int key;
    for(;;){
        key = GetKey();
        switch (key){
            case up : list.Eeach(Up);
                    break;
            case down: list.Eeach(Down);
                    break;
            case esc : return;
        }
    }
}
```

main.cpp

Segédfüggvények

Top-Down

```
void Up(TItem* w)
{
    w->ScrollUp();
}

void Down(TItem* w)
{
    w->ScrollDown();
}
```

TApplication appl:
 TMyScrollFileWindow *a1
 TMyScrollWindow *a2,*a3,*a4

list.Insert():
 a1 a2 a3 a4
 ↓ ↓ ↓ ↓
 list: p1,p2,p3,p4 (TItem*)

list.Eeach(Up):
 Up(p1);Up(p2);Up(p3);Up(p4)
 p1->ScrollUp()
 p2->ScrollUp()

main.cpp

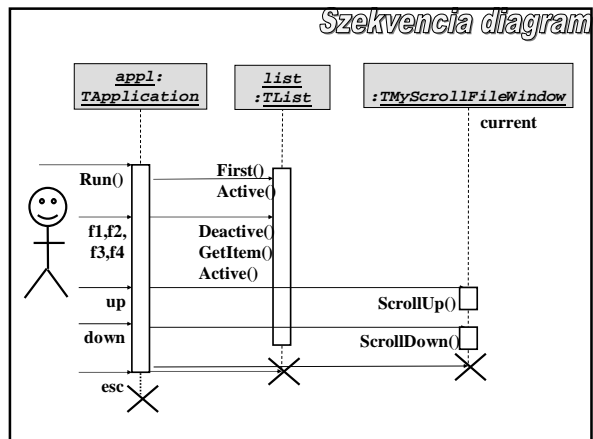
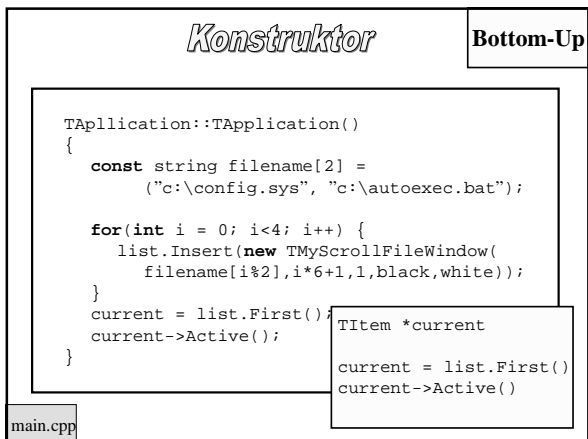
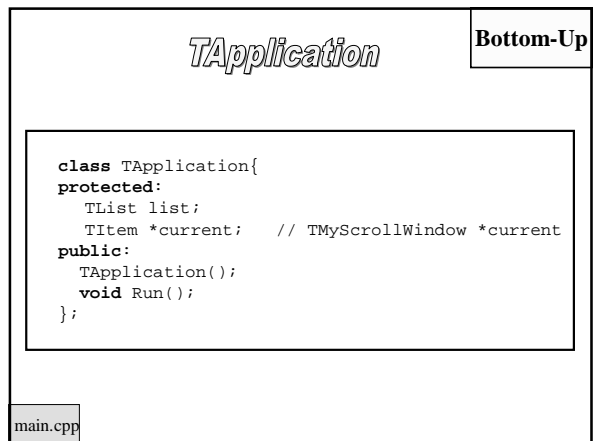
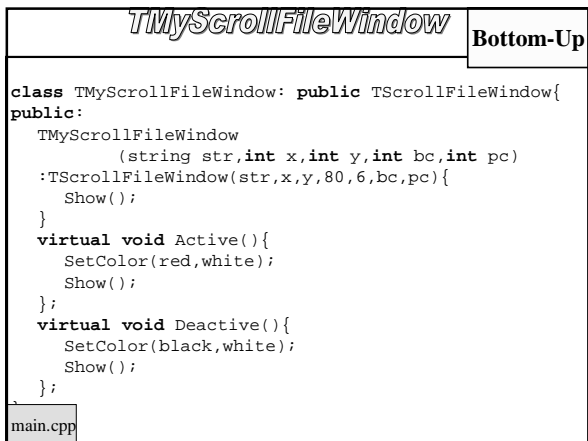
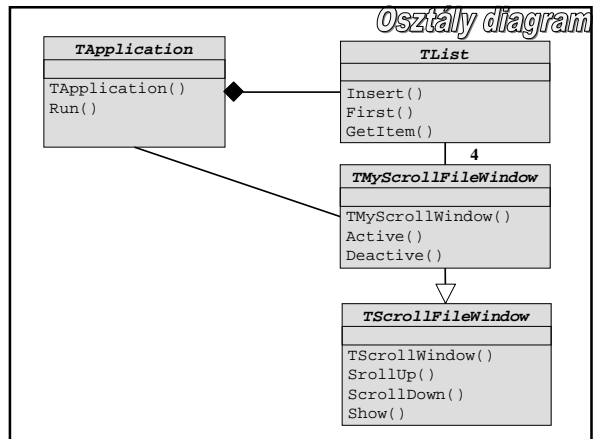
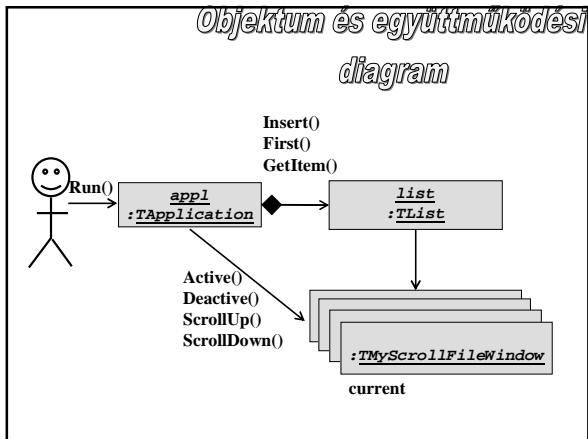
3. Feladat



Jelenítsen meg a képernyőn négy fájl-görgető ablakot egymás alatt. Bal felső sarkaik rendre az 1,1; 1,7; 1,13; 1,19. Szélességük egységesen 80, magasságuk: 6. Kezdetben az első görgető ablak az aktuális. Az aktuális ablak háttérszíne mindig piros, a többi ablak háttérszíne az inicializáláskor megadott háttérszín. Az 1. és a 3. ablakban a c:\autoexec.bat file-t, a 2. és a 4. ablakban a c:\config.sys fájl jelenjen meg.

A program vezérlése:

- ♦ A fel-le kurzorvezérlő billentyűkkel az aktuális ablak tartalmát lehessen görgetni
- ♦ Az F1, F2, F3, F4 billentyűkkel aktivizálhassuk az 1., 2., 3. vagy a 4. Ablakot.
- ♦ Az Esc-re tűnjenek el az ablakok, és kilépés.



Esemény kezelő

Bottom-Up

```

void TApplication::Run()
{
    int key;
    for(;;){
        key = GetKey();
        switch (key){
            case f1 : case f2 : case f3 : case f4 :
                current->Deactive();
                current = list.GetItem(key-f1+1);
                current->Active();
                break;

            case up : current-> ScrollUp();
                break;

            case down: current-> ScrollDown();
                break;

            case esc : return;
        }
    }
}
main.cpp

```

Főprogram

Bottom-Up

```

#include "ukey.h"
#include "uwin.h"
#include "ulist.h"
using namespace Key;
using namespace Color;

class TMyReadWindow ;
class TMyScrollFileWindow ;
class TApplication ;

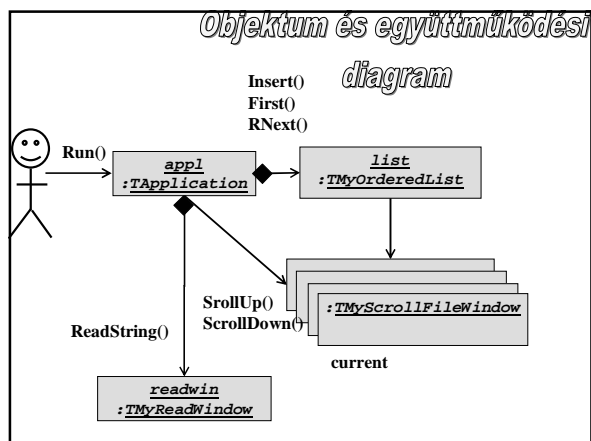
int main()
{
    TApplication appl;
    appl.Run();
}
main.cpp

```

4. Feladat

A képernyő alján jelenjen meg egy beolvasóablak “Állomány neve:” címmel, amelyből egymás után kérjünk be négy darab állománynevet. (Tegyük fel, hogy az állományok léteznek.) Minden beolvasás után tegyük a képernyőre egy-egy ablakot (mindig az előző ablak alá), mely a beolvasott nevű állományt tudja görgetni. Az állományablakok közepén, egymás alatt helyezkednek el, két alapon fehérek, és 40*5-ös méretűek. Az állományok beolvasása végén a beolvasó ablakot tüntessük el! Az ablakok bejárás sorrendje az állománynevek abc sorrendje legyen. Kezdetben az aktuális állományablak a névsorban első.

- ♦ A Tab billentyűvel lehessen lépegetni az ablakok között.
- ♦ A fel-le billentyűre görgethessük az aktuális állományt.
- ♦ Az Esc-re tüntessük el az állományablakokat és kilépés.



Főprogram

```

#include "ukey.h"
#include "uwin.h"
#include "ulist.h"
using namespace Key;
using namespace Color;

class TMyReadWindow ;
class TMyScrollFileWindow ;
class TMyOrderedList ;
class TApplication ;

int main()
{
    TApplication appl;
    appl.Run();
}
main.cpp

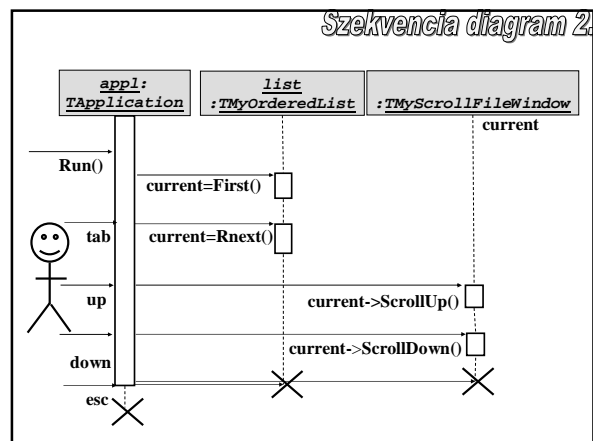
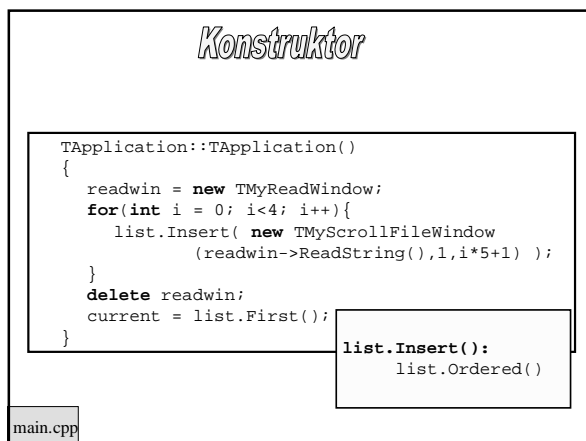
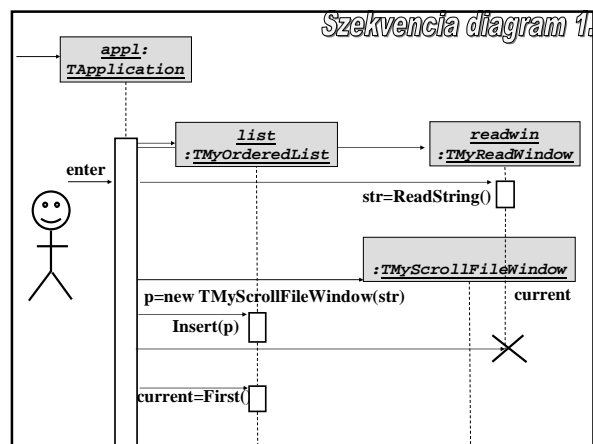
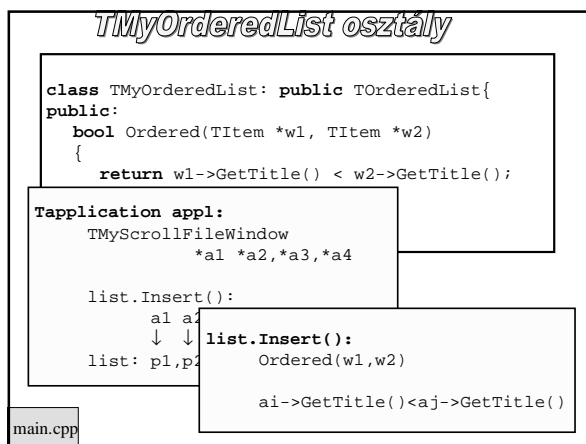
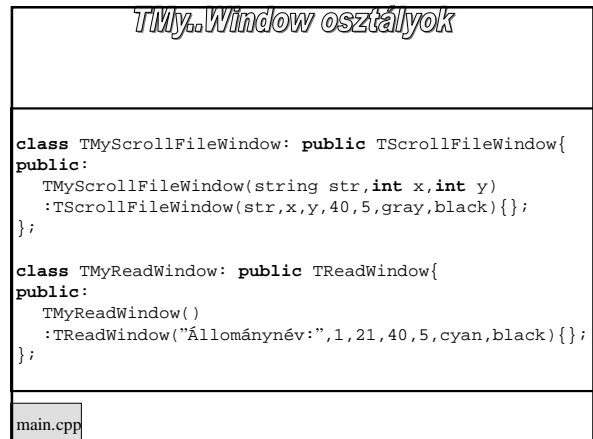
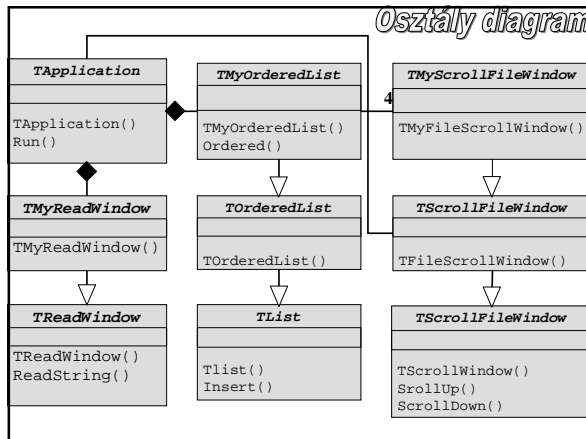
```

TApplication

```

class TApplication{
protected:
    TMyOrderesList list;
    TMyReadWindow *readwin;
    TItem *current; // TMyScrollWindow *current
public:
    TApplication();
    void Run();
};
main.cpp

```

Esemény kezelő

```
void TApplication::Run()
{
    int key;
    for(;;){
        key = GetKey();
        switch (key){
            case tab : current = list.RNext(current)
                        break;
            case up  : current->ScrollUp();
                        break;
            case down: current->ScrollDown();
                        break;
            case esc : return;
        }
    }
}
```

main.cpp

VÉGE