

## Programmation de machines à l'ESI École Supérieure d'Informatique (Haute École de Bruxelles) 2<sup>e</sup> années

**12**

```

C:\Documents and Settings\All Users\Documents\User\ESI\LabC_Cpp\Projet\CO2\Smb\graphEditor\SMBWinLab...
graphique - Classes
+ Graphe
+ Observateur
+ SujetObservation
+ TArc
+ TForm
+ TNoeudGUI
+ WinLabObserver
+ WinLabObserver
+ WinLabObserver
+ WinLabObserver
+ Fonctions
Unit.cpp WinLabObserver.h suetObservation.h
#include <cvl.h>
#include "graphe.h"
#include "observateur.h"
#include "sujetobservation.h"
#include "tarc.h"
#include "tform.h"
#include "tnoeudgui.h"
#include "winlabobserver.h"
class WinLabObserver : public Observateur
{
public:
    TLabels * info;
    virtual fastcall WinObs(TComponent * proprio);
    virtual fastcall ~WinObs();
    void _fastcall AltF4(TObject *Sender, WORD «Key, TShiftState Shift);
};
WinObs * wo;
Graphe * suet;
TComponent * proprio;
public:
    _fastcall WinLabObserver( Graphe * s, TComponent * proprio );
    ~WinLabObserver();
    void rafraichir(SujetObservation * so);
};
    
```

**10**

```

void _fastcall TForm1::SupprimerNoeud(TObject *Sender)
{
    this->Enabled=false;
    this->Visible=false;
    string cp = string(this->Noeud->Nom.C_str());
    //cp += " -> " + string(this->Noeud->GetDestin() + 1);
    int i = this->Graphe->SupprimerNoeud(cp);
    if (i == -1)
    {
        Application->MessageBox("Ce noeud n'existe pas", "Echec suppression", 0);
        Destroy();
        this->Noeud->Clear();
    }
}

void TForm1::SupprimerNoeud()
{
    this->AppNoeud = new WinLabObserver(this);
    this->AppNoeud->SetParent(this);
    this->AppNoeud->SetCaption("Supprimer un Noeud");
    this->AppNoeud->SetIcon(this->Icon);
    this->AppNoeud->SetLocation(this->Noeud->GetDestin() + 1);
    this->AppNoeud->SetCaption("Supprimer le Noeud");
    this->AppNoeud->SetIcon(this->Icon);
}
    
```

**8**

**7**

**5**

**4**

```

class Graf
{
private:
    class TArce;
    class TNoeudi{
    public:
        TNoeudi(string nom, TNoeudi* svt=NULL, TArce* lien=NULL)
        {Nom=nom;SVT=svt;Lien=lien;}
        string Nom;
        TNoeudi* Svt;
        TArce* Lien;
    };

    class TArce{
    public:
        friend class Graf::TNoeudi;
        TArce(TNoeudi* destination=NULL, TArce* suivant=NULL)
        {Destination=destination; Suivant=suivant;}
        TNoeudi* Destination;
        TArce* Suivant;
    };
public:
    Graf(string temp="Graphe 1");
    string NomGraphe() {return NomGraf;}
    void ChangerNomGraphe(string);

    int AjouterNoeud(string);
    bool NoeudExiste(string);
    int SupprimerNoeud(string);
    int NombreDeNoeuds();
}
    
```

**2**

**1**

Afficher le graphe? (A)  
Renome le graphe? (B)  
Ajouter un Noeud? (C)  
Supprimer un Noeud? (D)  
Afficher le nombre de Noeuds? (E)  
Ajouter un Arc? (F)  
Supprimer un Arc? (G)  
Afficher le nombre d'Arcs? (H)  
Matrice d'accesibilite? (I)  
Matrice de Warshall? (J)  
Matrice d'accesibilite? (K)  
Vecteur de TNoeud? (L)  
Vecteur de TArce? (M)  
Le plus court chemin? (N)  
Quitter? (Q)

Modificateurs  
Nouveau nom du graphe :  
Acter nouveau nom du graphe  
Noeud :  
Ajouter Noeud  
Supprimer Noeud  
Arc ou Chemin :  
Noeud départ : Noeud arrivée  
D :> A->C->F  
Ajouter Arc  
Supprimer Arc  
Recherche d'un chemin  
Représentation schématique du graphe MaVille  
Définition en extension du graphe MaVille  
NOEUDS: (A, B, C, D, E, F)  
ARCS: ((A, B), (A, C), (C, F), (D, A), (D, E), (E, D), (F, A))

Toutes les photographies représentées ici proviennent de Mikael Degeer.