

Exercice 1, découverte UML

OBJECTIF

A l'issue de la réalisation de cet exercice, les étudiants doivent être de réaliser un ensemble de diagramme de classe et d'obtenir le squelette du programme en C++ correspondant.

INSTALLATION D'ARGO UML (PAS NECESSAIRE)

Le fichier d'installation d'ArgoUML se trouve sous :

K:\ES\Maitres-Eleves\SLO\Modules\SL228_POBJ\CoursUML\ArgoUML

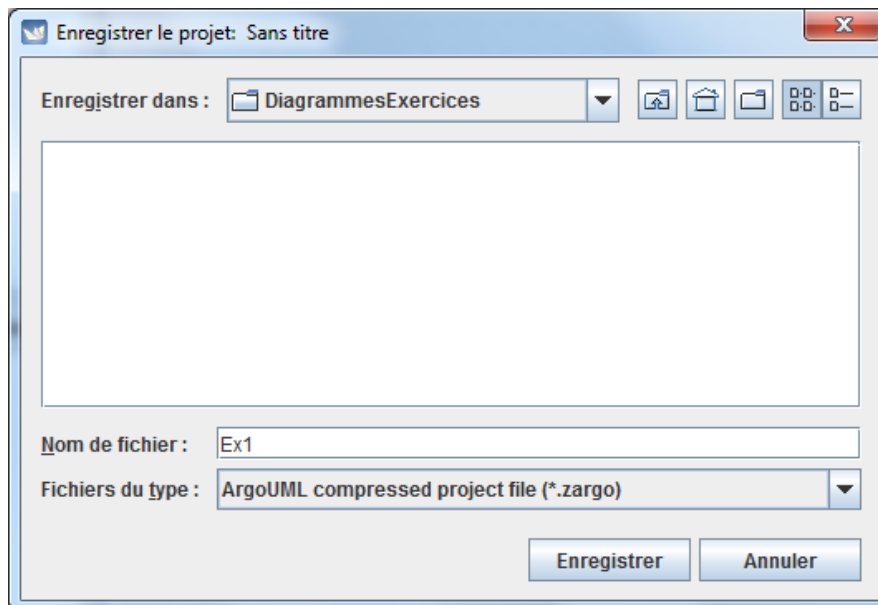
Démarrez l'installation par un double-clic sur *ArgoUML-0.34-setup.exe*.

PREPARATION DU PROJET

Voici le principe de la création d'un projet de modélisation avec ArgoUML.

Créez sous votre H: une structure de répertoires comme POBJ\UML\Exercices.

Lancer ArgoUML et enregistrez le projet "Sans titre" en le nommant Ex1.



CONFIGURATION

Veuillez-vous référer au page 3 et 4 du chapitre 3, pour choisir C++ au niveau de la notation et ajouter C++ au niveau du profile.

REALISATION DES DIAGRAMMES DE CLASSE

Le principe de l'exercice est de réaliser une classe **Ex1** permettant de choisir une forme, de saisir les paramètres et d'afficher le résultat.

Il faut associer la classe Ex1 avec la classe abstraite **FormesGéométriques**, qui est dérivée en **TriangleRectangle**, **Rectangle** et **Cercle**.

DETAIL DE LA CLASSE Ex1

La classe Ex1 va permettre la réalisation du programme principal. Veuillez introduire au niveau du diagramme de classe :

ATTRIBUTS PUBLIQUE

int CodeFormeChoisie

METHODES PUBLIQUES

int main()
int ChoisirForme()
void SaisirParametres(int CodeForme)
void ExecuterLesCalculs(int CodeForme)
void AfficherResultats(int CodeForme)

DETAIL DE LA CLASSE ABSTRAITE FORMESGEOMETRIQUES

La classe abstraite FormesGéométriques est prévues pour offrir un support générique au calcul de la surface et du périmètre d'une forme géométrique plane quelconque. Veuillez introduire au niveau du diagramme de classe :

ATTRIBUTS PRIVES

double Surface
double Perimetre

METHODES PUBLIQUES

virtual double CalculerSurface(double x1, double x2)
virtual double CalculerPerimetre(double x1, double x2)
double GetSurface()
double GetPerimetre()

DETAIL DE LA CLASSE DERIVEE TRIANGLERECTANGLE

METHODES PUBLIQUES

double CalculerSurface(double base, double hauteur)
double CalculerPerimetre(double base, double hauteur)

DETAIL DE LA CLASSE DERIVEE RECTANGLE

METHODES PUBLIQUES

double CalculerSurface(double largeur, double hauteur)
double CalculerPerimetre(double largeur, double hauteur)

DETAIL DE LA CLASSE DERIVEE CERCLE

METHODES PUBLIQUES

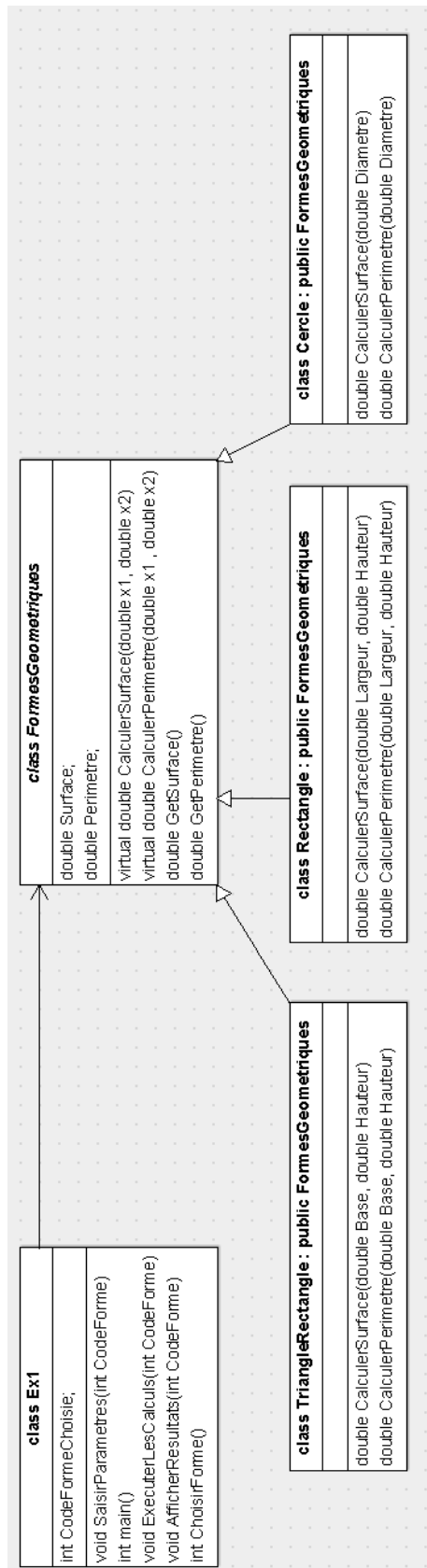
double CalculerSurface(double diametre)
double CalculerPerimetre(double diametre)

RELATION ENTRE LES CLASSES

Il faut établir une relation d'association (uni association) entre la classe Ex1 et la classe FormesGéométriques.

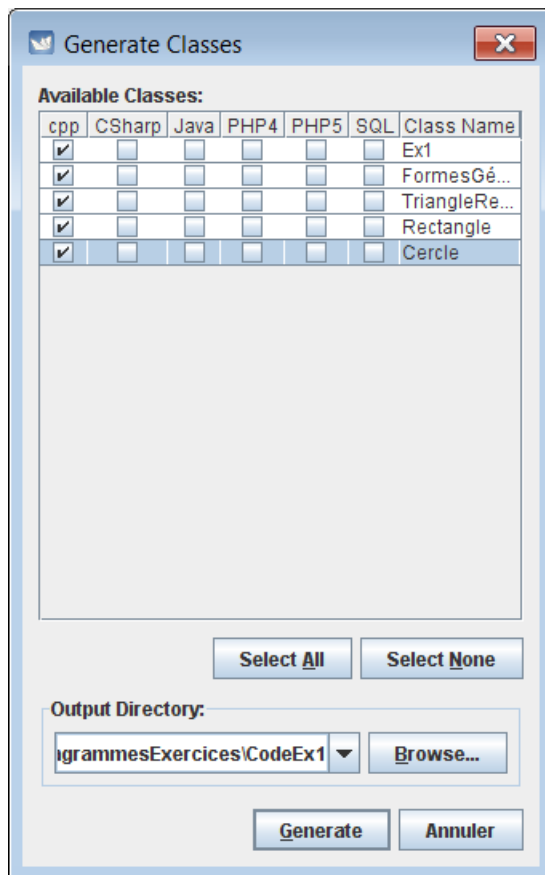
Les classes TriangleRectangle, Rectangle et Cercle sont des spécialisations de la classe FormesGéométrique.

RESULTAT A OBTENIR



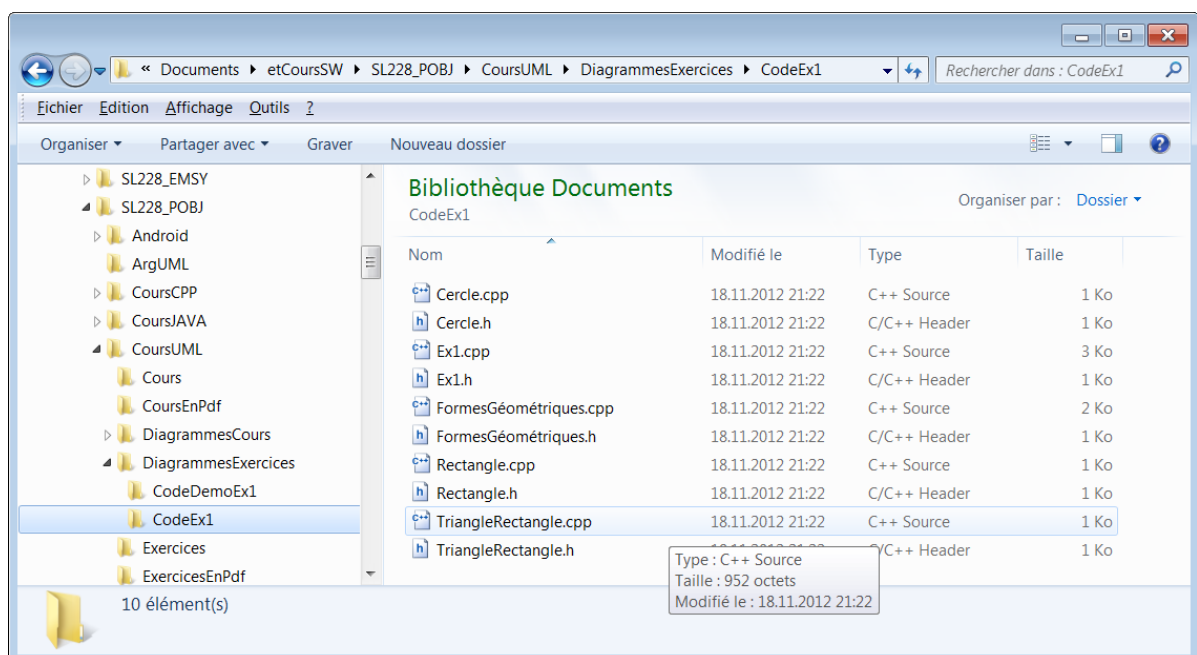
GENERATION DU CODE

Utilisez le menu **Génération**, **Générer toutes les classes...**



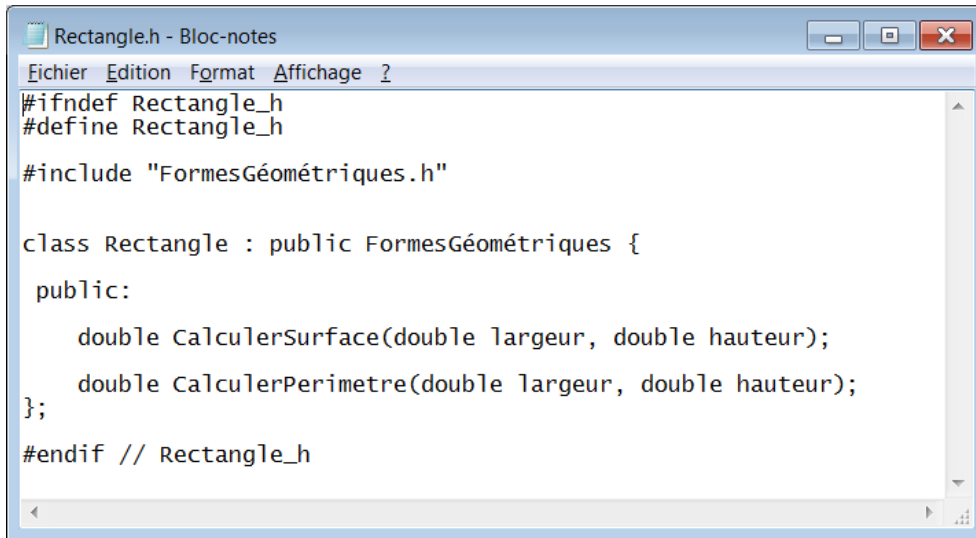
Puis cliquer sur Generate.

On doit obtenir les fichiers suivants :



RECTANGLE.H

Voici le contenu du fichier Rectangle.h



```
Fichier Edition Format Affichage ?
#ifndef Rectangle_h
#define Rectangle_h

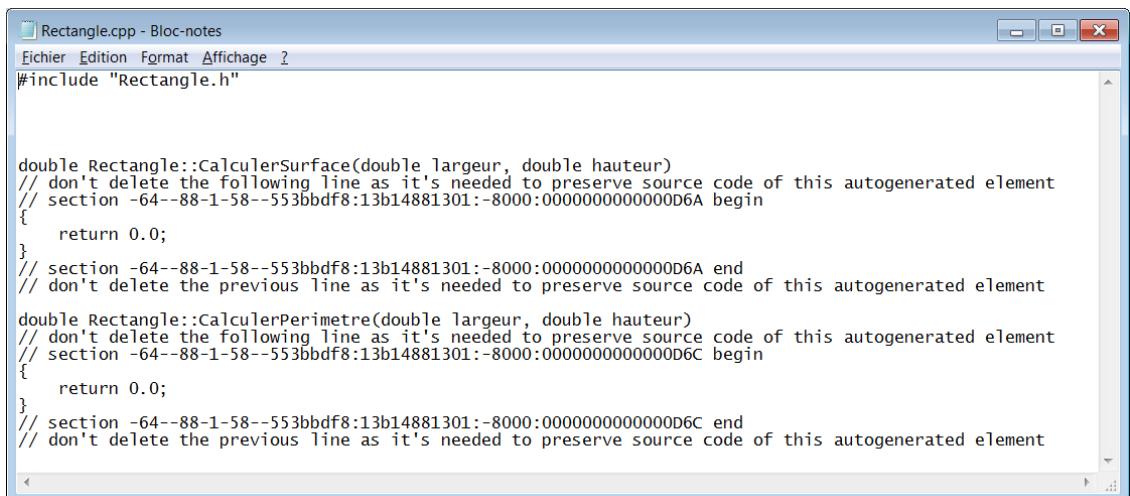
#include "FormesGéométriques.h"

class Rectangle : public FormesGéométriques {
public:
    double CalculerSurface(double largeur, double hauteur);
    double CalculerPerimetre(double largeur, double hauteur);
};

#endif // Rectangle_h
```

RECTANGLE.CPP

Voici le contenu du fichier Rectangle.cpp



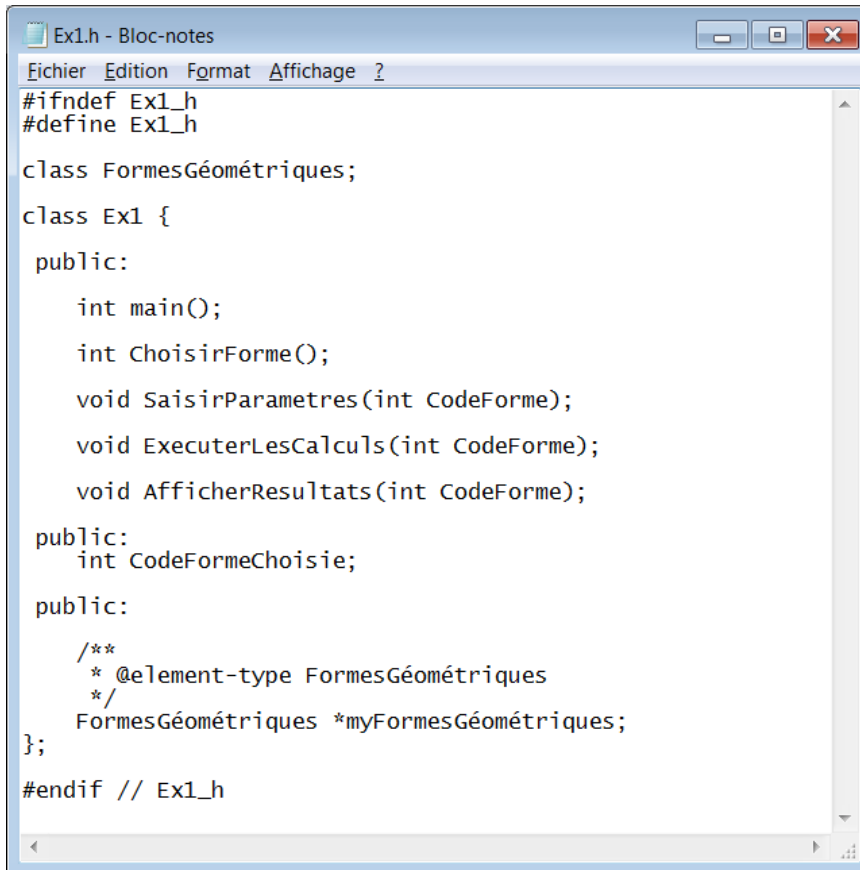
```
Fichier Edition Format Affichage ?
#include "Rectangle.h"

double Rectangle::CalculerSurface(double largeur, double hauteur)
// don't delete the following line as it's needed to preserve source code of this autogenerated element
// section -64--88-1-58--553bbdf8:13b14881301:-8000:0000000000000D6A begin
{
    return 0.0;
}
// section -64--88-1-58--553bbdf8:13b14881301:-8000:0000000000000D6A end
// don't delete the previous line as it's needed to preserve source code of this autogenerated element

double Rectangle::CalculerPerimetre(double largeur, double hauteur)
// don't delete the following line as it's needed to preserve source code of this autogenerated element
// section -64--88-1-58--553bbdf8:13b14881301:-8000:0000000000000D6C begin
{
    return 0.0;
}
// section -64--88-1-58--553bbdf8:13b14881301:-8000:0000000000000D6C end
// don't delete the previous line as it's needed to preserve source code of this autogenerated element
```

Ex1.h

Voici le contenu du fichier Ex1.h



```
Ex1.h - Bloc-notes
Fichier Edition Format Affichage ?
#ifndef Ex1_h
#define Ex1_h

class FormesGéométriques;

class Ex1 {
public:
    int main();
    int ChoisirForme();
    void SaisirParametres(int CodeForme);
    void ExecuterLesCalculs(int CodeForme);
    void AfficherResultats(int CodeForme);

public:
    int CodeFormeChoisie;

public:
    /**
     * @element-type FormesGéométriques
     */
    FormesGéométriques *myFormesGéométriques;
};

#endif // Ex1_h
```

OBSERVATION DES CODES GENERE

Veuillez observer le contenu des autres fichiers générés.

CONCLUSION

Les fichiers générés peuvent être utilisé comme canevas d'un projet console en C++. Il est nécessaire de compléter le corps des méthodes.

Les codes numériques des sections sont prévus pour pouvoir modifier les diagrammes de classe sans perdre le code ajouté par la suite. (Par exemple ajout d'un attribut ou d'une méthode oubliée lors du design).

SUITE PRATIQUE

Il vous sera demandé par la suite, comme exercice C++ de compléter le code généré et de l'adapter pour permettre la saisie des informations et l'affichage.