

BTS SN 2^{ème} ANNÉE – Option IR

IR2NUM3

LANGAGE C++ LES CLASSES

Durée : 6h

2020-2021

Introduction

La programmation orientée objet est basée sur l'utilisation des classes. Dans cette séance de travaux pratiques nous allons nous intéresser à :

- La création de classes.
- La création et la manipulation des objets.
- La définition et le fonctionnement des constructeurs et des destructeurs.
- L'utilisation de la surcharge des opérateurs classiques.

Dans chaque partie, il vous sera demandé de réaliser une certaine tâche professionnelle. A chaque fois, en conclusion de votre travail, vous devrez :



- Montrer au professeur que le **cahier des charges** a bien été rempli et répondre à ses questions ;
- Répondre aux questions du **document-réponse**.

1. CREATION D'UNE CLASSE

2. SURCHARGE DES OPERATEURS

3. REPRESENTATION GRAPHIQUE

4. Annexe

Seront pris en compte dans l'évaluation de votre travail :

- La bonne réalisation des **installations** ;
- Le **soin** accordé au matériel ;
- Le bon **rangement** du matériel en fin de TP ;
- La **clarté** des explications données au professeur ;
- La **qualité rédactionnelle** du document-réponse.

En fin du TP, vous veillerez à :

- **Ranger** soigneusement le matériel ;
- Modifier le nom du document-réponse, en remplaçant les « **YYY** » par vos noms ;
- **L'envoyer** par mail ou le glisser dans la BAL de votre professeur.

Matériel disponible :



- Ordinateur équipé de *Windows 10*, et des logiciels *CodeBlock* ;
- Baies informatiques câblées et équipées de switches ;
- Câbles RJ45.

Ressources disponibles :



- Cours sur langage C++;
- « *Guide du Technicien* ».

Introduction

1. CREATION D'UNE CLASSE

2. SURCHARGE DES OPERATEURS

3. REPRESENTATION GRAPHIQUE

4. Annexe

1. CREATION D'UNE CLASSE

Créez un projet de type console (voir annexe) et ajoutez à votre projet une classe nommée "Rectangle". En vous aidant du cours et du TD complétez les fichiers "Rectangles.cpp" et "Rectangle.h" de façon à ce que votre classe comporte :

- Deux attributs de type **double** nommés **hauteur** et **largeur**.
- Un attribut de type **chaîne de 30 caractères** nommé **nom**.
- **Un constructeur** qui initialise à zéro tous les attributs de la classe et affiche "**Rectangle créé avec H = 0 et L = 0**".
- **Un constructeur** attendant deux arguments nommés **h** et **l** de type **double** destinés à initialiser les attributs hauteur et largeur de notre classe. Lors de son exécution le constructeur affiche "**Rectangle créé avec H = ... et L = ...**" où les ... représentent les valeurs des arguments **h** et **l**.
- **Un constructeur** identique au précédent avec en plus un argument nommé **name** de type chaîne de 30 caractères qui initialisera, en plus de **h** et **l**, l'attribut **nom** de la classe.
- **Un destructeur** qui affiche "**Rectangle détruit**".
- Une méthode nommée **grossirRectangle** possédant un argument de type **float** nommé **coef**. Cette méthode a pour objectif de multiplier les attributs hauteur et largeur de la classe par la valeur de l'argument **coef**.
- Une méthode nommée **elargirRectangle** possédant un argument de type **float** nommé **coefL**. Cette méthode a pour objectif de multiplier l'attribut largeur de la classe par la valeur de l'argument **coefL**.
- Une méthode nommée **grandirRectangle** possédant un argument de type **float** nommé **coefH**. Cette méthode a pour objectif de multiplier l'attribut hauteur de la classe par la valeur de l'argument **coefH**.
- Une méthode nommée **afficher** qui va afficher les caractéristiques de l'objet de classe rectangle, c'est-à-dire, **hauteur, largeur, nom, surface et périmètre**.



Dans le programme principal créez **trois pointeurs** sur des objets de type rectangle (nommés **rectangle1**, **rectangle2** et **rectangle3**) en utilisant le principe de l'allocation dynamique de mémoire puis **affichez leurs caractéristiques** et enfin **détruisez-les**. Rectangle1 ne sera pas initialisé, Rectangle2 aura une hauteur de 2 et une largeur de 4 et le rectangle3 aura une hauteur de 3, une largeur de 5 et sera nommé rect3sur5.



- Complétez le compte-rendu de mesures.
- Commentez précisément l'affichage sur la console lors de l'exécution de votre programme

2. SURCHARGE DES OPERATEURS

Dans cette deuxième partie nous allons imaginer les opérations suivantes :



- Lorsqu'on additionne deux rectangles leurs hauteur et leurs largeurs s'additionnent.
- Lorsqu'on soustrait deux rectangles leurs hauteur et leurs largeurs se soustraient.
- L'opération `rectangle2 = -rectangle1` donne un rectangle2 dont la hauteur devient la largeur de rectangle1 et la largeur la hauteur de rectangle1.

Proposez une modification de votre classe Rectangle qui intègre les opérations précédentes et utilise la surcharge d'opérateur. Testez votre nouvelle classe en réalisant un programme qui additionne, soustrait et inverse des objets rectangles. A chaque opération utilisez la méthode affiche pour valider le bon fonctionnement des opérations.



- Complétez le compte-rendu de mesures.
- Si vous avez le temps :
 - ✓ Proposez la surcharge de l'opérateur "+=".
 - ✓ Proposez la surcharge de l'opérateur "-=".

Introduction

1. CREATION D'UNE
CLASSE

2. SURCHARGE DES
OPERATEURS

3. REPRESENTATION
GRAPHIQUE

4. Annexe

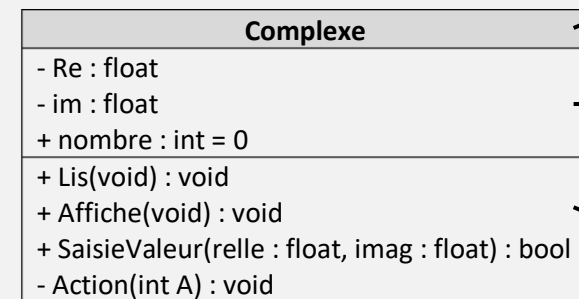
3. REPRESENTATION GRAPHIQUE

Dans cette dernière partie nous allons nous intéresser à la représentation graphique de la classe que nous venons de créer. Cette représentation se nomme diagramme des classes.

- En utilisant le site <https://www.draw.io/> et l'exemple ci-dessous, tracez le diagramme des classes de votre classe.



```
class Complexe {
public:
    void Lis();
    void Affiche();
    bool SaisieValeur(float relle, float imag);
    int nombre = 0;
private:
    float re, im;
    void Action(int A);
};
```



Nom de la class

Liste des attributs

Liste des méthodes



- Complétez le compte-rendu de mesures.

Introduction

1. CREATION D'UNE CLASSE

2. SURCHARGE DES OPERATEURS

3. REPRESENTATION GRAPHIQUE

4. Annexe

1. ANNEXE

Création d'un projet de type console :

Introduction

1. CREATION D'UNE CLASSE

2. SURCHARGE DES OPERATEURS

3. REPRESENTATION GRAPHIQUE

4. Annexe

The sequence of screenshots illustrates the process of creating a Qt Console Application project in Qt Creator:

- Qt Creator Main Window:** The 'Projet' menu is highlighted with a red dashed circle (1). The 'Qt Console Application' template is selected in the 'Choisir un modèle' dialog (2).
- Qt Console Application Wizard - Project Location:** The 'Nom' field is set to 'prog1' (labeled 'Nom du projet') and the 'Créer dans' field is set to 'C:\temp\QT' (labeled 'Dossier du projet'). The 'Suivant >' button is highlighted with a red dashed circle.
- Qt Console Application Wizard - Kit Selection:** The 'Desktop Qt 5.14.2 MinGW 64-bit' kit is selected. The 'Suivant >' button is highlighted with a red dashed circle.
- Qt Console Application Wizard - Translation File:** The 'Language' is set to '<none>'. The 'Suivant >' button is highlighted with a red dashed circle.
- Qt Console Application Wizard - Define Build System:** The 'Build system' is set to 'qmake'. The 'Suivant >' button is highlighted with a red dashed circle.

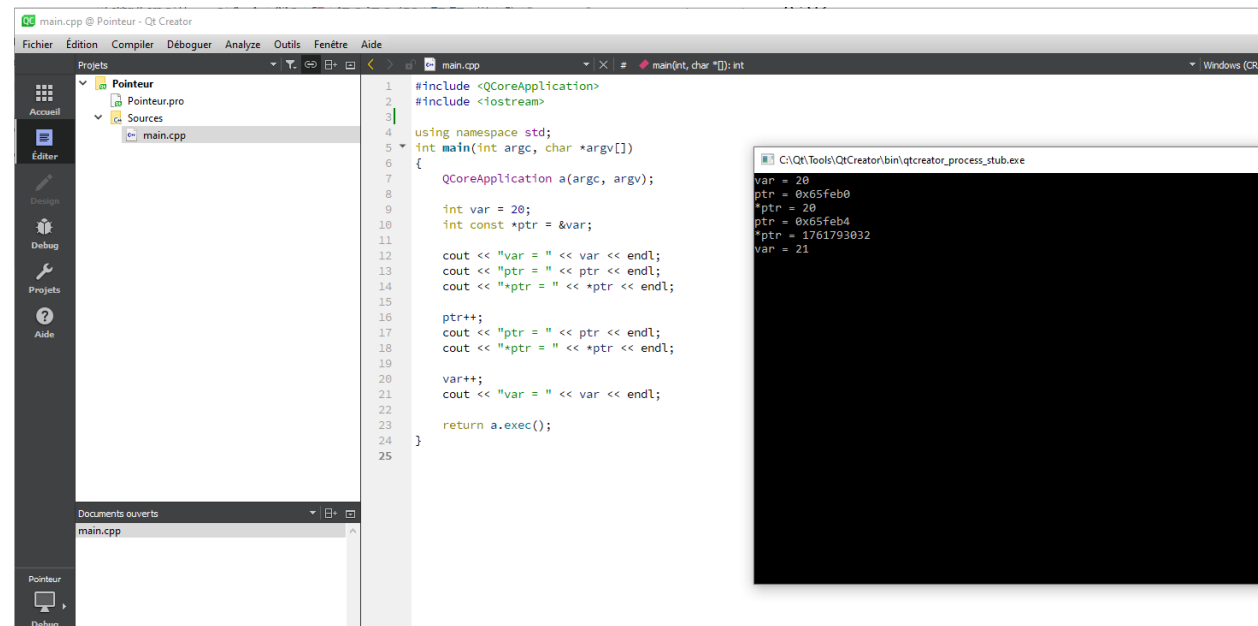
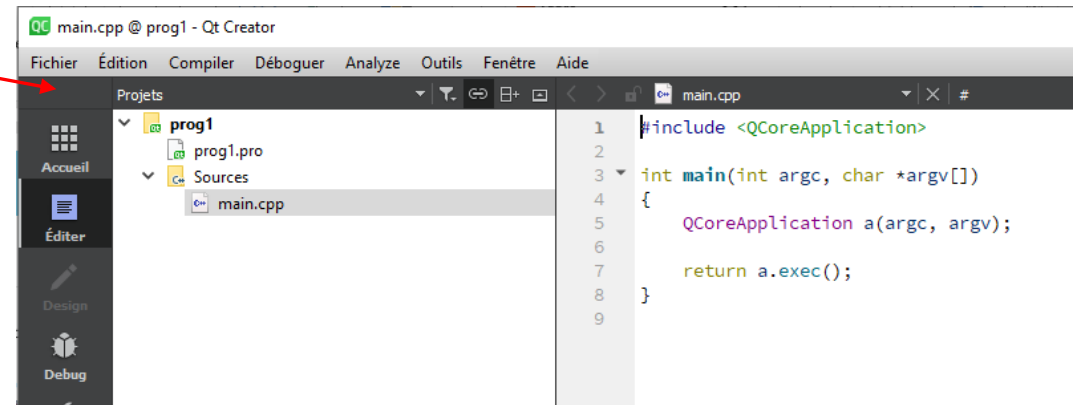
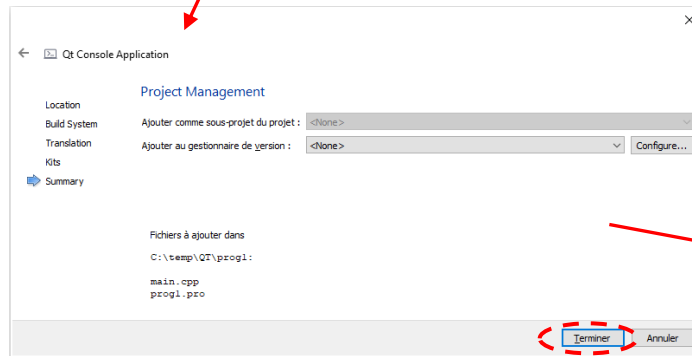
Introduction

1. CREATION D'UNE CLASSE

2. SURCHARGE DES OPERATEURS

3. REPRESENTATION GRAPHIQUE

4. Annexe



Ajout d'une classe au projet :

main.cpp @ allocation_dynamique - Qt Creator

Fichier Édition Compiler Débuguer Analyze Outils Fenêtre Aide

Nouveau fichier ou projet... Ctrl+N

Ouvrir un fichier ou projet... Ctrl+O

Ouvrir le fichier avec...

Fichiers récents

Projets récents

Sessions

Close Project "allocation_dynamique"

Close All Files in Project "allocation_dynamique"

New File or Project - Qt Creator

Choisir un modèle :

Projets

Application

Library

Autre projet

Non-Qt Project

Import Project

Fichiers et classes

C++

Modeling

Qt

GLSL

General

Java

Python

C++ Class

C++ Source File

C++ Header File

Créer un C++ header et un fichier source pour une nouvelle classe que vous pouvez ajouter à un projet C++.

Plateformes supportées: Desktop Generic Linux Device Périphérique Android WebAssembly Runtime Windows Runtime (Local)

Choose... Annuler

Define Class

Class name: **Nom de la classe**

Base class: <Custom>

Project Management

Ajouter au projet : **Nom du projet.pro**

Ajouter au gestionnaire de version : <None> Configure...

point.h @ classe_1 - Qt Creator

Fichier Édition Compiler Débuguer Analyze Outils Fenêtre Aide

Projets

classe_1

classe_1.pro

Headers

point.h

Sources

main.cpp

point.cpp

Exemple de la classe nommée Point qui a été ajoutée au projet classe_1

Introduction

1. CREATION D'UNE CLASSE

2. SURCHARGE DES OPERATEURS

3. REPRESENTATION GRAPHIQUE

4. Annexe