Algorithmique/Langage
1ère année

Introduction à Visual C++

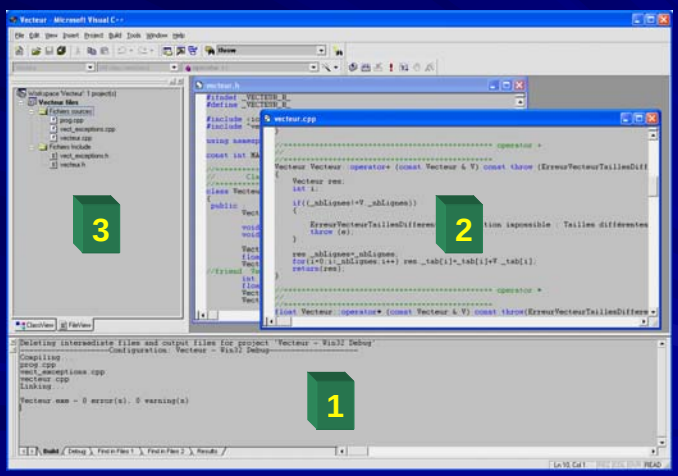
Yacine BELLIK

IUT d'Orsay
Université Paris XI

Plan

- Aperçu global de l'environnement
  - Zone des messages
  - Zone d'édition
  - Zone d'exploration
  - Barre d'outils
- Premiers pas avec la classe Ensemble
  - Création d'un nouveau projet
  - Édition des fichiers
  - Compilation
  - Exécution
  - Travail avec la vue par classes
- Travail autonome

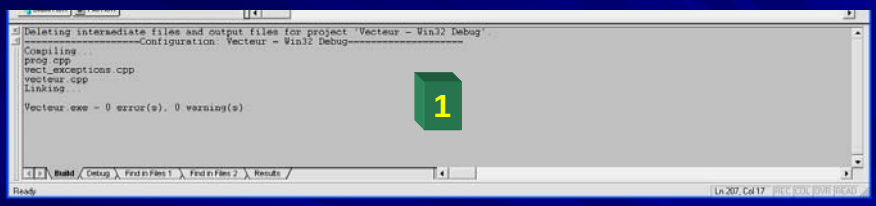
Aperçu global de l'environnement



3 zones principales

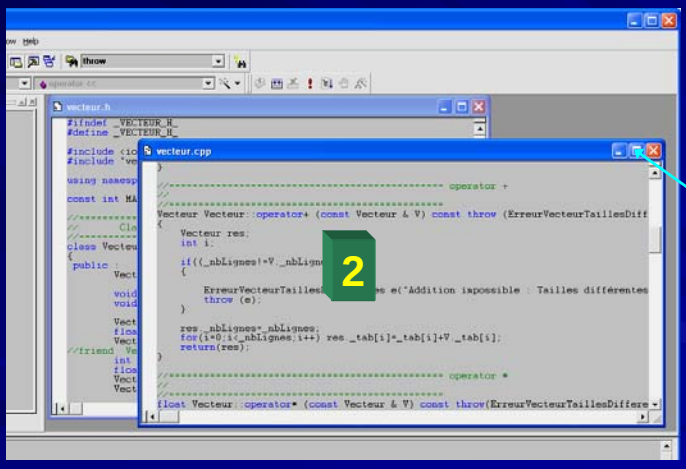
1. Zone des messages
2. Zone d'édition
3. Zone d'exploration

Zone des messages



- Cette zone affiche les messages d'erreurs de compilation
- Un double clic sur un message d'erreur nous amène directement au fichier concerné et à la ligne concernée

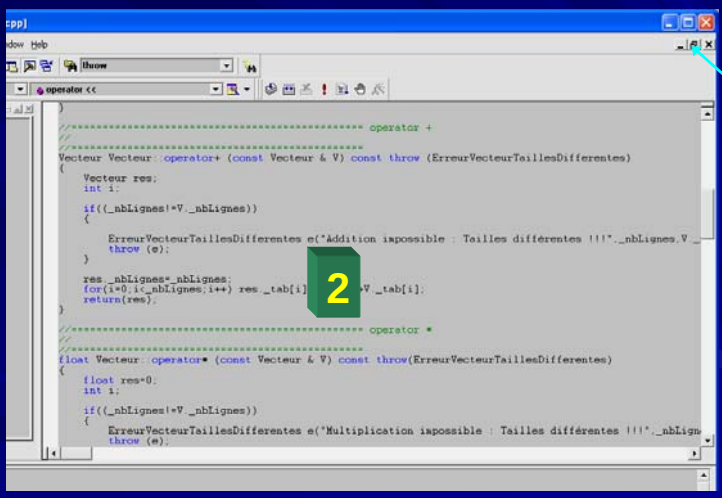
Zone d'édition



Contient les fenêtres des fichiers en cours d'édition

Bouton d'agrandissement

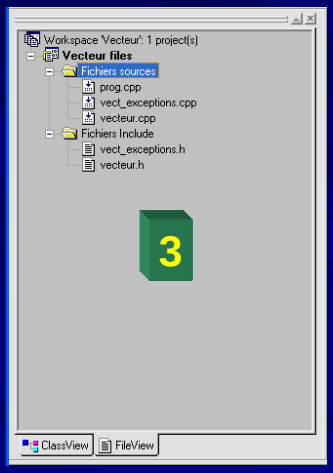
Zone d'édition



Bouton de restauration de taille initiale

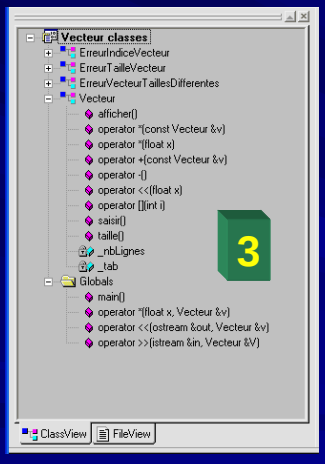
La zone d'édition prend cet aspect si on agrandit une des fenêtres

Zone d'exploration : vue par fichiers



- La zone d'exploration permet de visualiser et de naviguer dans les fichiers sources du projet ou dans ses classes
- Elle présente 2 onglets
  - la vue des fichiers sources
  - La vue des classes
- Les fichiers sources portent l'extension .cpp
- Un double-clic sur un fichier l'ouvre dans la zone d'édition
- Un troisième onglet apparaît dans le cas d'une application graphique (vue des ressources graphiques)

Zone d'exploration : vue par classes



- La vue des classes montre toutes les classes du projet
- Les méthodes apparaissent en grenat et les données membres en bleu cyan
- un verrou apparaît à côté des méthodes ou données membres privées
- Le répertoire Globals montre toutes les variables globales ou fonctions externes
- Un double-clic sur le nom du classe ouvre le fichier ou est déclarée cette classe
- Un double-clic sur le nom du méthode (ou fonction) ouvre le fichier où cette méthode est définie (corps de la méthode)
- Un double-clic sur le nom d'une donnée ouvre le fichier où cette donnée est déclarée
- Dans tous les cas le curseur est automatiquement placé au bon endroit dans le fichier
- On peut également faire un clic avec le bouton droit sur un élément pour faire apparaître un menu contextuel

Barre d'outils

Compiler le fichier
en cours d'édition

Construire un exécutable
(compilation + édition de liens)

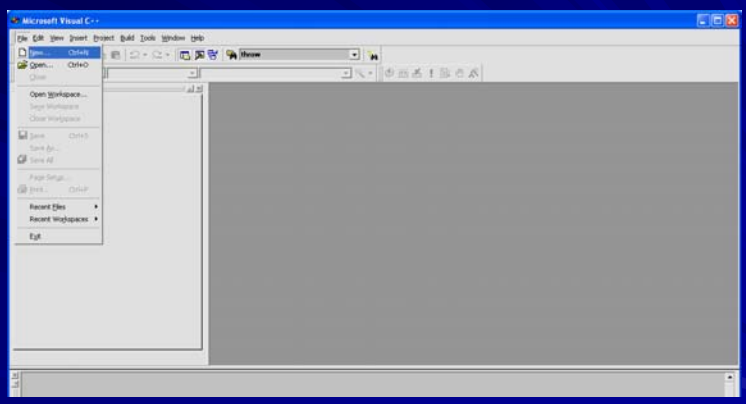
Construire un exécutable
puis lancer l'exécution

Premiers pas avec la classe Ensemble

Démarrage

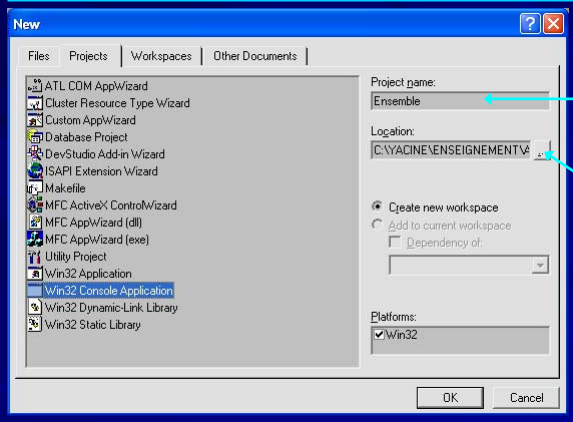
- Pour lancer Visual C++ cliquez sur :
  - Démarrer
  - Tous les programmes
  - Microsoft Visual Studio 6.0
  - Microsoft Visual C++ 6.0
- Si vous en voyez pas apparaître les 3 zones principales :
  - Aller dans le menu View
  - Cliquer sur Workspace et Output
  - Ou bien appuyer sur Alt+0 et Alt+2

Création d'un nouveau projet



- Cliquez sur le menu File puis sur l'item new

Choix du type, du nom et de l'emplacement du nouveau projet



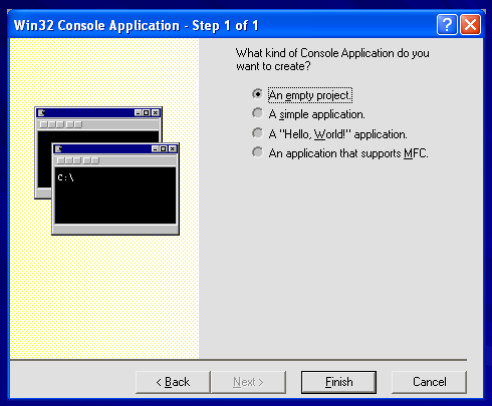
Tapez ici le nom du projet

Cliquez sur ce bouton pour choisir le répertoire du nouveau projet

Attention : un sous-répertoire portant le nom du projet sera automatiquement crée à l'intérieur du répertoire sélectionné

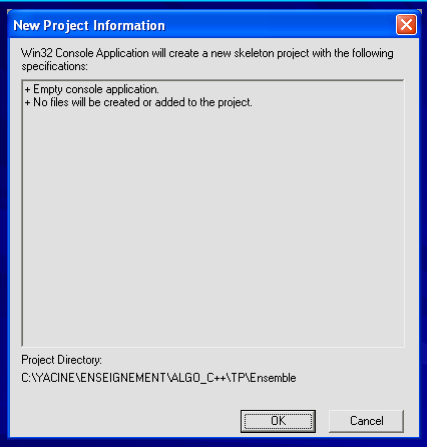
- Cliquez sur le type de projet Win32 Console Application
- Cliquez sur OK

Choix du modèle initial



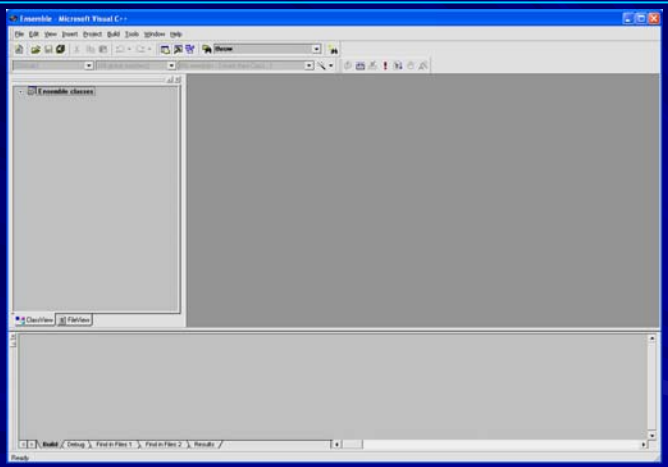
- Choisir An empty project et cliquez sur Finish

Résumé et validation finale



- Une boîte résumant les caractéristiques du nouveau projet apparaît
- Cliquer sur OK pour valider

Nouveau projet

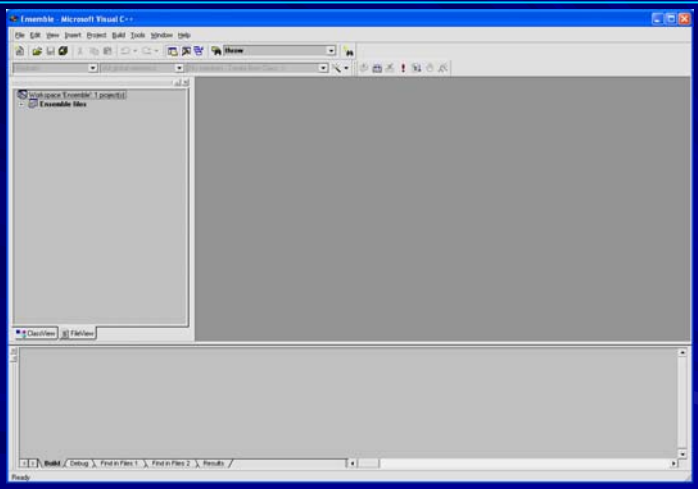


- Voici l'affichage obtenu après validation

Nouveau projet vide

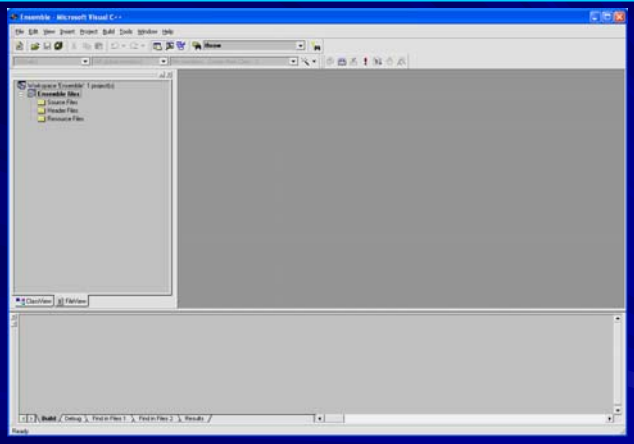
- Pour l'instant notre nouveau projet est vide
- Il ne comporte aucun fichier
- Nous allons donc lui rajouter des fichiers
- Cliquez sur l'onglet File View pour basculer sur la vue par fichiers

Vue par fichiers



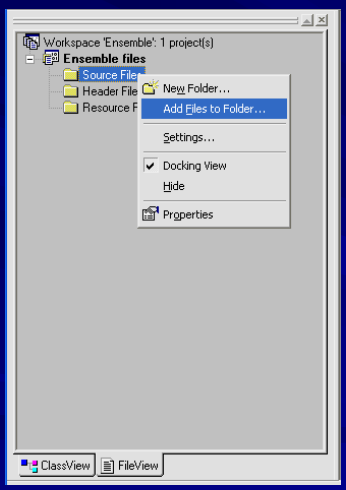
- Cliquez maintenant sur le symbole + à gauche de Ensemble files

Vue par fichiers



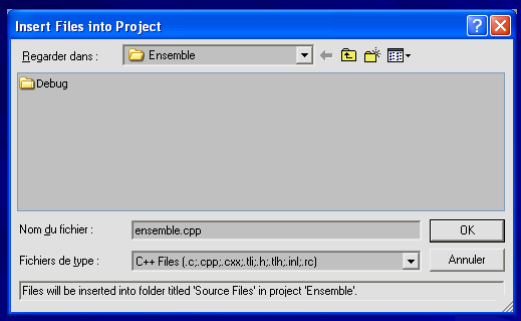
- Visual a déjà crée pour nous des répertoires pour classer nos futurs fichiers
- Attention ces répertoires ne sont pas physiques (disque dur)
- Ils servent juste à avoir une vue ordonnée de nos fichiers

Ajout d'un fichier au projet



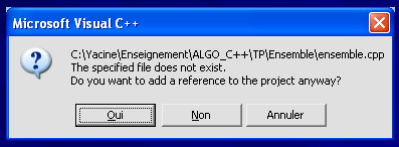
- Nous allons maintenant ajouter un nouveau fichier (ensemble.cpp) au projet
- Cliquez avec le bouton droit de la souris sur le répertoire Source Files
- Un menu apparaît
- Choisir Add Files to folder

Ajout d'un fichier au projet



- Dans la boîte de dialogue qui apparaît, taper ensemble.cpp

Ajout d'un fichier au projet

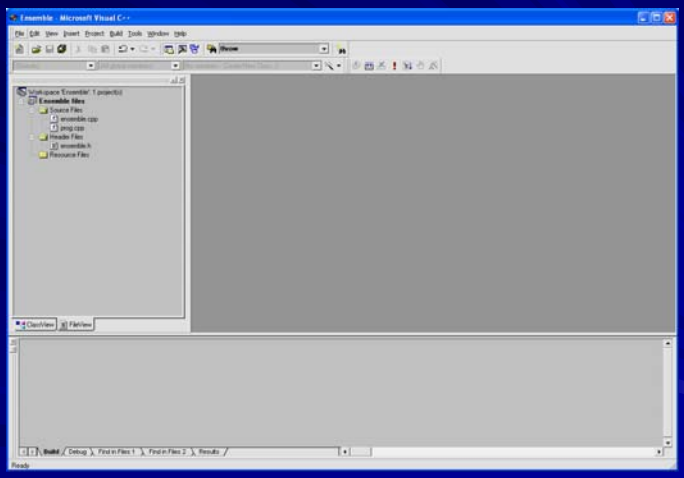


- Visual nous dit que le fichier que nous voulons rajouter au projet n'existe pas
- Il nous demande de confirmer qu'on veut quand même le rajouter
- Cliquer sur Oui, nous créerons le fichier plus tard

Ajout de fichiers au projet

- refaire la même chose pour rajouter le fichier prog.cpp
- Refaire la même chose pour rajouter le fichier ensemble.h mais cette fois-ci dans le répertoire headers

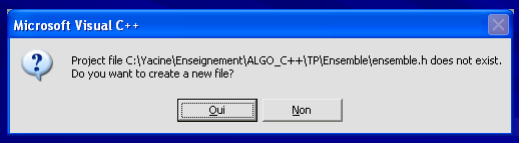
Vue par fichiers



- Notre projet ressemble à ceci maintenant

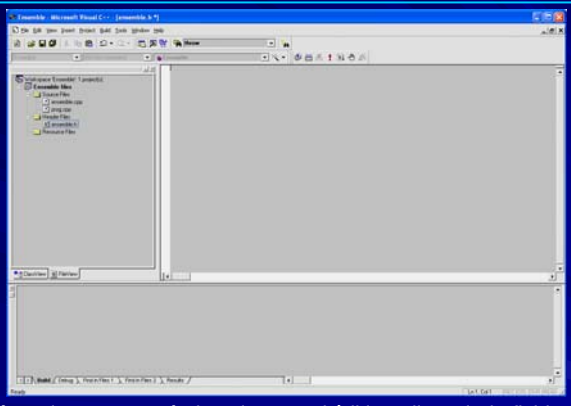
Création du fichier ensemble.h

- Nous allons maintenant créer le fichier ensemble.h
- Pour cela nous allons double-cliquer dessus
- La boîte de dialogue suivante apparaît



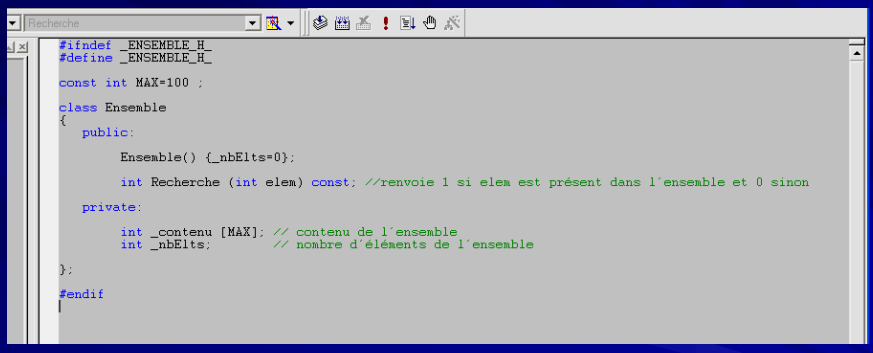
- Visual nous dit que le fichier n'existe pas et nous demande une confirmation pour le créer
- Cliquer sur Oui

Edition du fichier ensemble.h



- Une fenêtre vierge apparaît dans la zone d'édition : il s'agit du fichier ensemble.h
- Remarquer le nom du fichier en cours d'édition sur la barre de titre de la fenêtre
- Le symbole * sur cette barre de titre signifie que le fichier n'a pas encore été sauvegardé

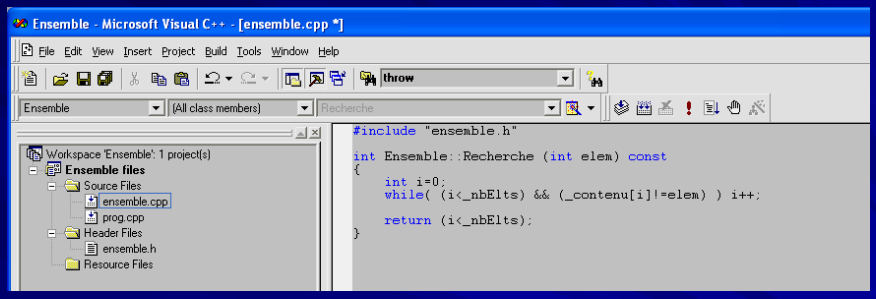
Edition du fichier ensemble.h



```
#ifndef _ENSEMBLE_H_
#define _ENSEMBLE_H_

const int MAX=100 ;

class Ensemble
{
   public:

        Ensemble() {_nbElts=0};

        int Recherche (int elem) const; //renvoie 1 si elem est présent dans l'ensemble et 0 sinon

   private:

        int _contenu [MAX]; // contenu de l'ensemble
        int _nbElts;        // nombre d'éléments de l'ensemble

};

#endif
```

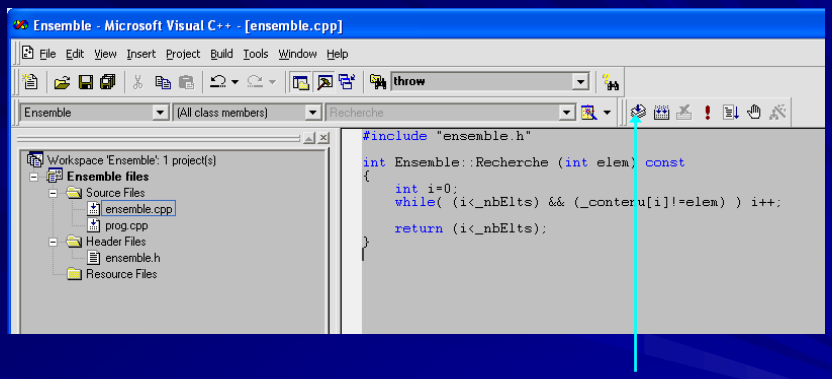
- Taper le code suivant dans le fichier ensemble.h

Edition du fichier ensemble.cpp



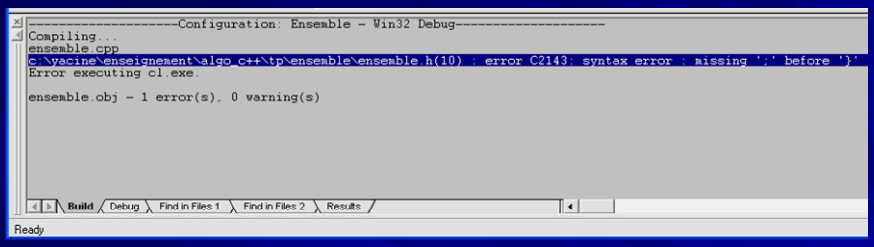
- Refaire la même chose avec le fichier ensemble.cpp
- Taper le code ci-dessus

Compilation du fichier ensemble.cpp



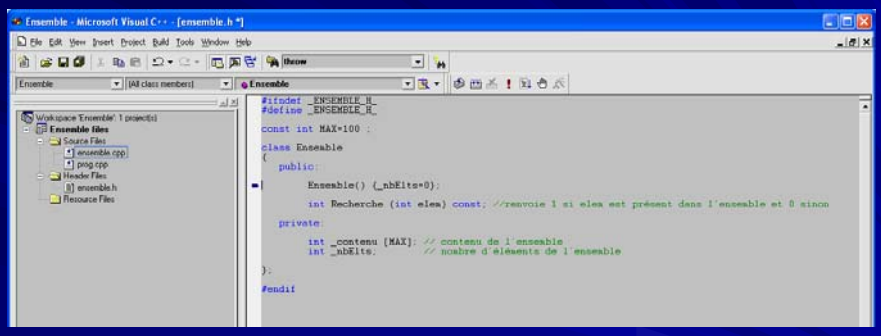
Lancer la compilation
du fichier ensemble.cpp
en cliquant sur ce bouton

Analyse des messages d'erreur



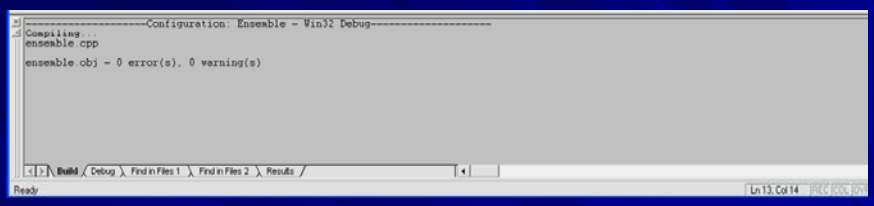
- La fenêtre des messages d'erreur affiche alors le message ci-dessus (manque un ; dans le fichier ensemble.h)
- Double-cliquer sur ce message

Correction de l'erreur



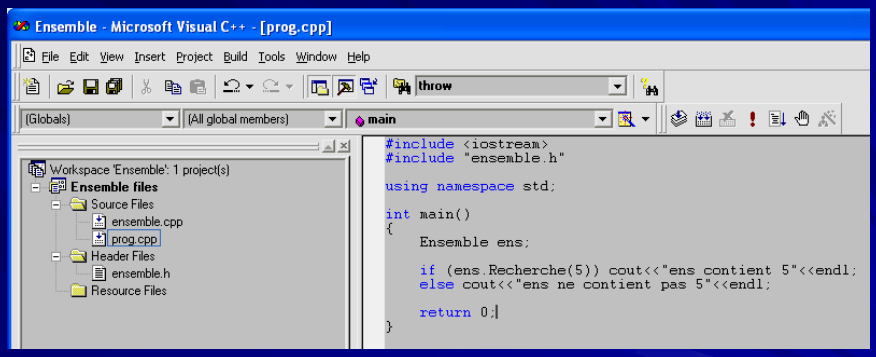
- Visual ouvre alors le fichier concerné et positionne directement le curseur sur la ligne de l'erreur
- Ajouter le ; manquant avant l'accolade fermante

Nouvelle compilation



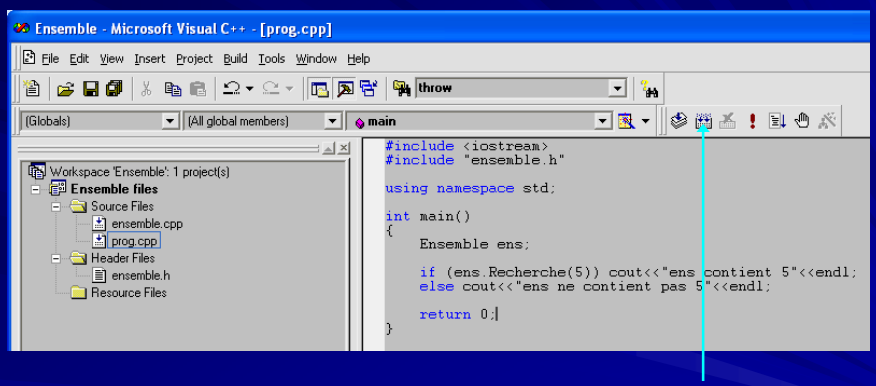
- Double-cliquer sur le fichier ensemble.cpp pour le faire revenir à l'éditeur
- Relancer sa compilation
- On obtient alors le message ci-dessus (pas d'erreurs)

Edition du fichier prog.cpp



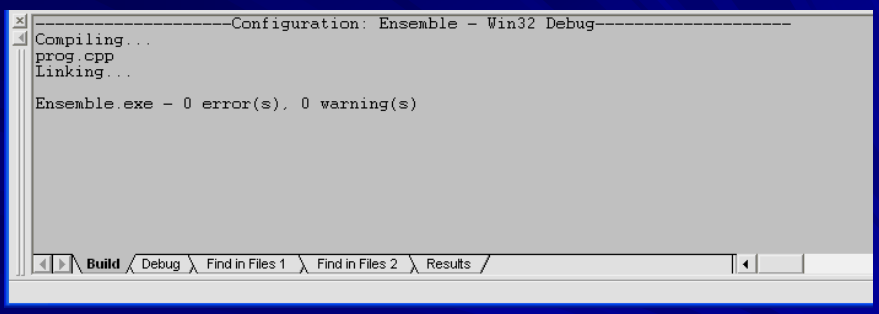
- Double-cliquer sur le fichier prog.cpp pour le créer
- Taper le code ci-dessus

Construction de l'exécutable



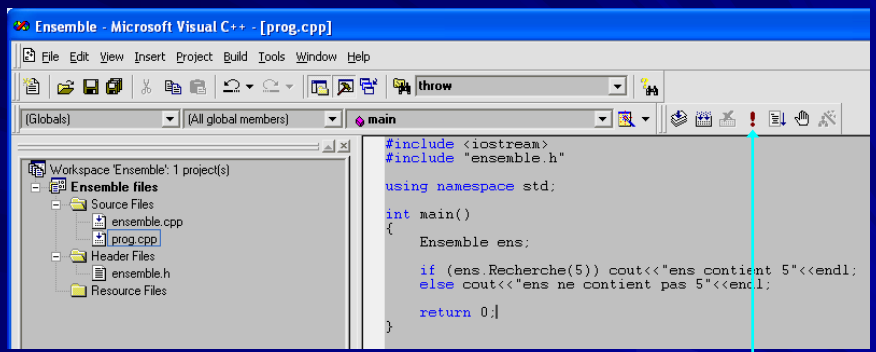
Cliquez sur ce bouton. Cela aura pour effet de recompiler tous les fichiers modifiés et de lancer l'édition de liens pour créer l'exécutable

Construction de l'exécutable



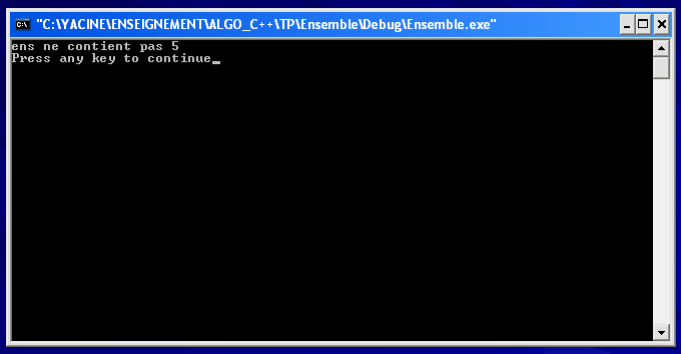
- On obtient l'affichage ci-dessus dans la fenêtre des messages

Exécution du programme



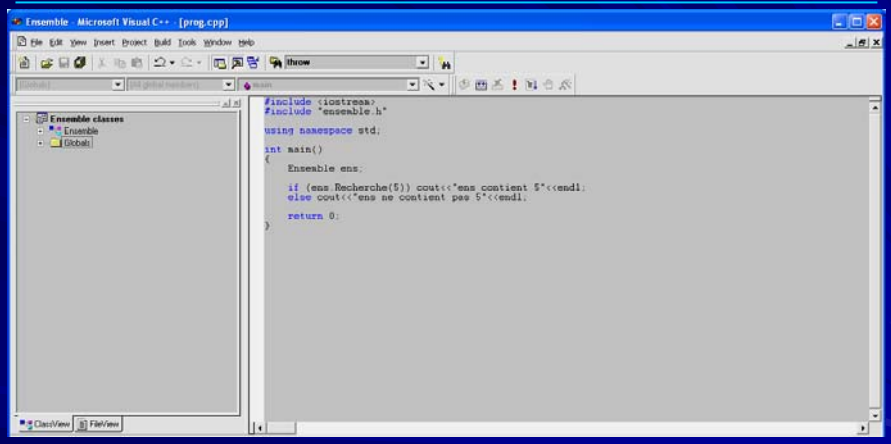
Cliquez sur ce bouton pour lancer l'exécution du programme

Exécution du programme



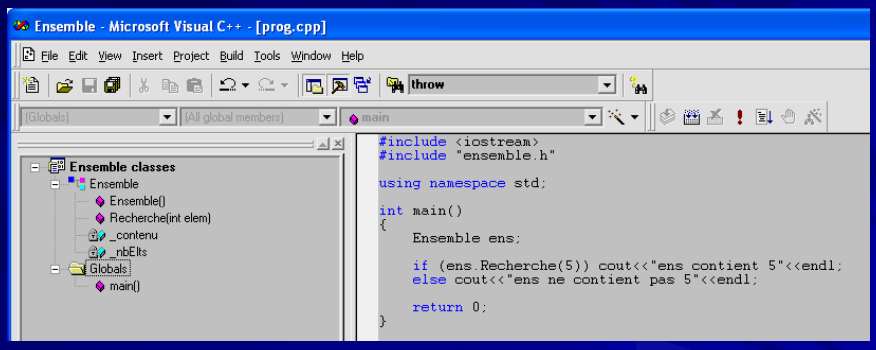
- Une nouvelle fenêtre apparaît dans laquelle le programme s'exécute
- Appuyez sur une touche du clavier pour quitter la fenêtre d'exécution

Vue par classes



- Cliquer sur l'onglet ClassView pour passer à la vue des classes
- Cliquer sur les symboles + à gauche de la classe Ensemble et Globals

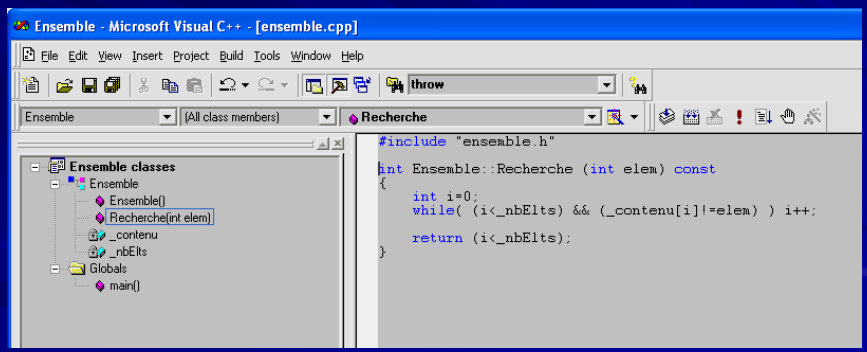
Vue par classes



- On obtient alors la vue ci-dessus
- On reconnaît le constructeur, la méthode Recherche, les données privées de la classe Ensemble ainsi que la fonction externe main

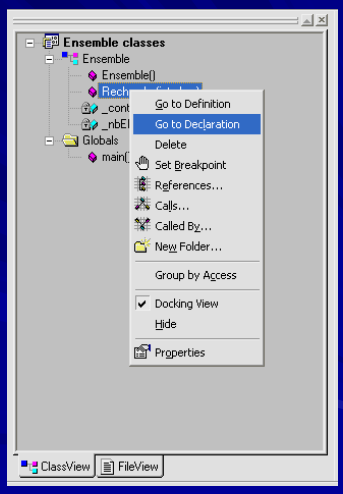
Utilisation de la vue par classes

- Pour accéder à la définition d'une méthode il suffit de double-cliquer dessus
- double-cliquer sur la méthode Recherche



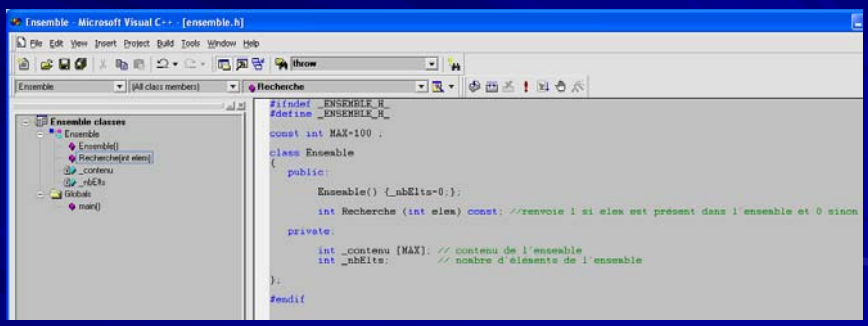
Utilisation de la vue par classes

- Pour accéder à la déclaration d'une méthode il faut cliquer dessus avec le bouton droit puis choisir Go to Declaration dans le menu qui apparaît



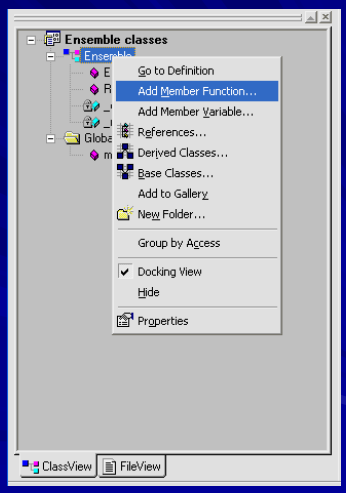
Utilisation de la vue par classes

- Pour accéder à la classe double cliquer dessus



Ajout d'une méthode taille

- Nous allons ajouter une méthode taille
- Cliquer avec le bouton droit sur la classe Ensemble
- Dans le menu qui apparaît choisir Add Member Function

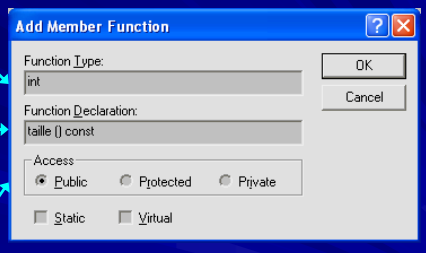


Ajout d'une méthode taille

Taper ici le type de retour

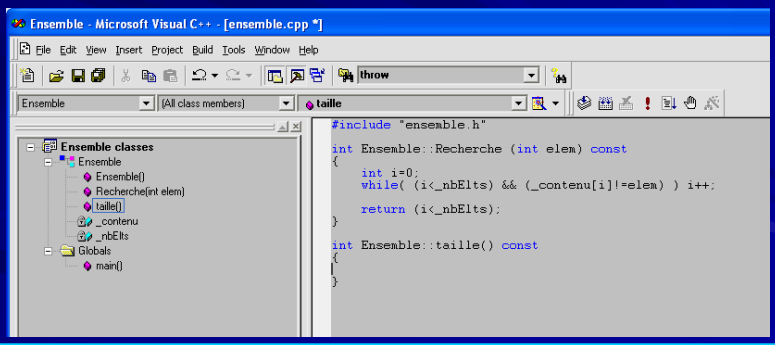
Taper ici le reste de l'entête

Choisir ici la méthode d'accés public



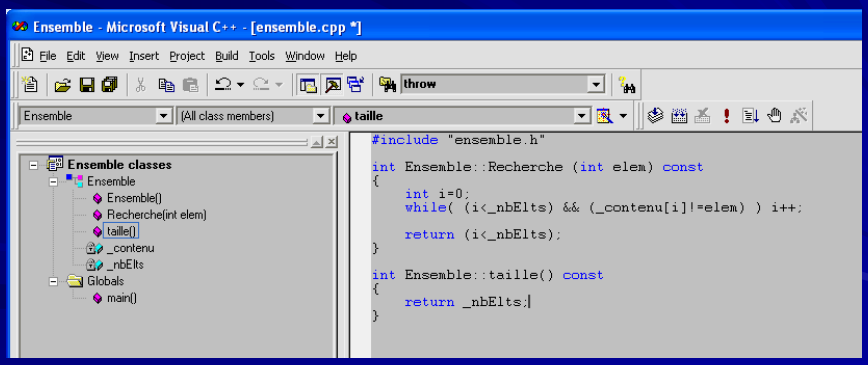
Ajout d'une méthode taille

- Visual a automatiquement rajouté l'entête de la nouvelle méthode dans le fichier .h et dans le fichier .cpp
- De plus, il nous a positionné directement au bon endroit dans le fichier .cpp pour écrire le corps de la nouvelle méthode
- Bien sûr il aurait toujours été possible de le faire nous même de façon manuelle en éditant directement les 2 fichiers .h et .cpp



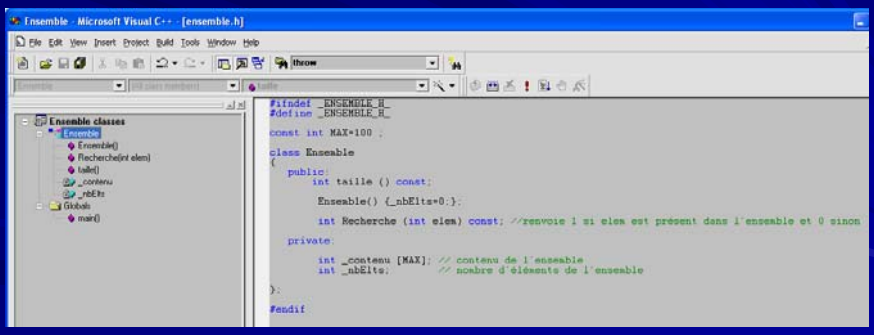
Ajout d'une méthode taille

- Taper le code ci-dessous pour la méthode taille



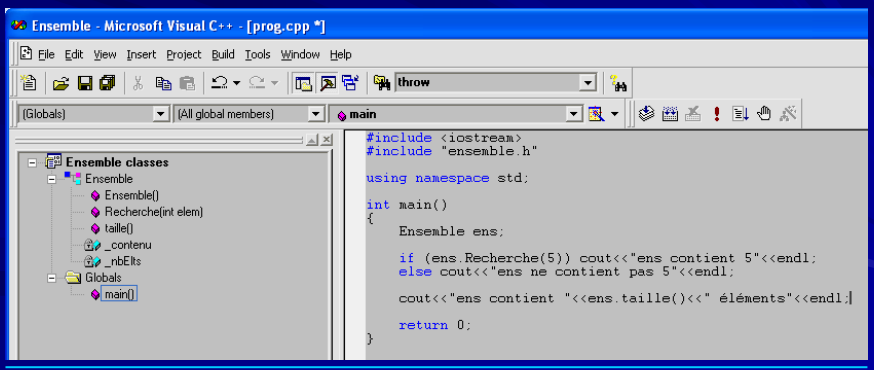
Ajout d'une méthode taille

- double cliquer sur la classe ensemble pour vérifier que l'entête de la méthode taille a bien été rajouté dans le fichier .h



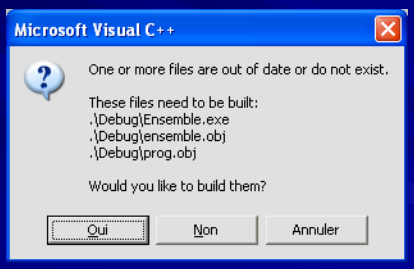
Modification de la fonction main

- double cliquer sur la fonction main et complétez le code comme ci-dessous



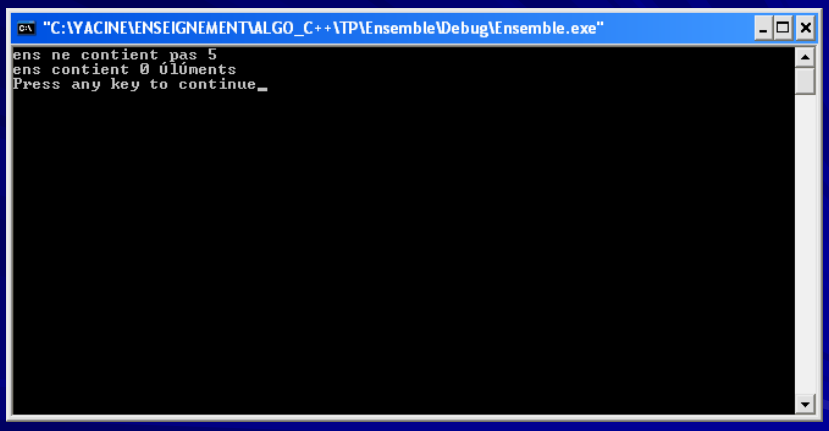
Exécution

- Cliquer sur ce bouton ! pour lancer la compilation, l'édition de liens et l'exécution en une seule opération



- Confirmez en cliquant sur Oui

Résultat de l'exécution



- Remarque : il est possible que les lettres accentuées n'apparaissent pas correctement.

Aide

- Il est toujours possible d'accéder à une aide contextuelle
- Positionner le curseur sur le mot-clef ou la fonction sur laquelle on désire avoir de l'aide
- Appuyez sur la touche F1

Travail autonome

Classe Ensemble : Partie A

Complétez les fichiers ensemble.h et ensemble.C
de manière à répondre aux questions suivantes :

1. Redéfinir l'opérateur **<<** pour l'affichage d'un ensemble.
2. Redéfinir l'opérateur **<<** entre un ensemble et un entier de manière à pouvoir écrire **e<<x**, ce qui aura pour effet de rajouter l'entier **x** à l'ensemble **e** (s'il n'y est pas déjà).
3. Testez dans le programme principal.

Classe Ensemble : Partie B

On souhaite rendre la classe Ensemble **dynamique**, c'est-à-dire que le tableau "_contenu" soit alloué de façon dynamique par le constructeur de la classe (la taille de l'ensemble est passée en paramètre au constructeur).

1. Quelles doivent être maintenant les données membres de la classe ? Justifiez.
2. Récrire le constructeur de manière à ce qu'il prenne en paramètre la taille de l'ensemble et qu'il fasse l'allocation correspondante.
3. Écrire le destructeur (afficher un message témoin "objet détruit" dans le destructeur).
4. Écrire le constructeur par recopie. Testez dans le programme principal.
5. Redéfinir l'opérateur d'affectation. Attention à l'auto-affectation (e=e). Testez dans le programme principal.