1. Objet Simples

    1. Présentation du Code
    2. Compilation
    3. Mon Premier Objet: `Box`

Objet Simples

L'objectif de ce premier TME est de bien se familiariser avec les mécanismes base régissant le fonctionnement d'un objet C++. Plus particulièrement sur les constructeurs & destructeurs, et les contextes dans lesquels ils peuvent être appelés.

Présentation du Code

Comme il a été annoncé en cours un soin tout particulier devra être accordé à la présentation de votre code. A cet effet un exemple type d'indentation vous est fourni ci-après. Toujours pour une meilleure lisibilité, réduisez la taille des tabulations à 2 ou 4 caractères au lieu du defaut de 8. Enfin, n'insérez jamais le caractère *tabulation* dans votre code source, configurez votre éditeur favori pour que celui-çi substitue automatiquement des espaces blanc ordinaires à la place d'une tabulation.

Pour **`vim`**/**`gvim`**, ajouter dans votre `~/.vimrc` les lignes suivantes:

```
:set shiftwidth=2
:set expandtab

:autocmd FileType c,cpp set cindent
:autocmd FileType make set noexpandtab shiftwidth=8
```

Pour **`emacs`**, ajoutez dans votre `~/.emacs` les lignes suivantes:

```
;; CC-Mode configuration.
 (defun local-c-mode-hook ()
  (setq c-echo-syntactic-information-p t
                        c-basic-offset 2
                             tab-width 4
                      indent-tabs-mode nil)
  (c-set-offset 'topmost-intro          0)
  (c-set-offset 'inclass              '++)
  (c-set-offset 'access-label          '-)
  (c-set-offset 'comment-intro         '-)
  (c-set-offset 'arglist-cont-nonempty 'c-lineup-arglist-close-under-paren)
  (c-set-offset 'arglist-close         'c-lineup-arglist-close-under-paren)
  (c-set-offset 'defun-block-intro     '+)
  (c-set-offset 'statement-block-intro '+)
  (c-set-offset 'block-close            0)
  (c-set-offset 'case-label            '+)
  (c-set-offset 'statement-case-intro  '+))
 (add-hook 'c-mode-common-hook 'local-c-mode-hook)

 (setq auto-mode-alist
  (cons '("\\.\\(h\\|hh\\|hpp\\)\\'" . c++-mode)
   auto-mode-alist))
```

Compilation

Dans le cadre de ce premier TME, nous allons nous contenter de faire de la compilation monolithique. C'est à dire que tout le code de chaque exercice sera contenu dans un seul fichier source, par exemple `Box.cpp`. L'éxécutable sera créé par compilation de cet unique fichier.

Compilation monolithique d'un fichier source:

```
g++ -Wall -g -o box Box.cpp
```

Ici, l'éxécutable généré sera `box`. Attention l'argment `-o` doit immédiatement être suivi du nom de l'éxécutable (`box`).

Autres arguments donnés à `g++`:

- `-g` demande d'inclure le code nécessaire pour pouvoir utiliser un débogueur (`gdb` ou `ddd`).
- `-Wall` demande d'afficher tous les avertissements. Vous devrez toujours utiliser ce drapeau (`flag`) et à l'issue de la mise au point, votre programme ne devra générer aucun avertissement.

Mon Premier Objet: `Box`

Nous allons implémenter la classe `Box` présentée en cours. La classe `Box` comporte les cinq attributs suivants:

- `_name` : le petit nom de l'objet courant (de type `std::string`).
- `_x1, _y1, _x2, y2` : les coordonnées des deux angles de la boite (de type `int`).