Hygiène de code

# Indentation

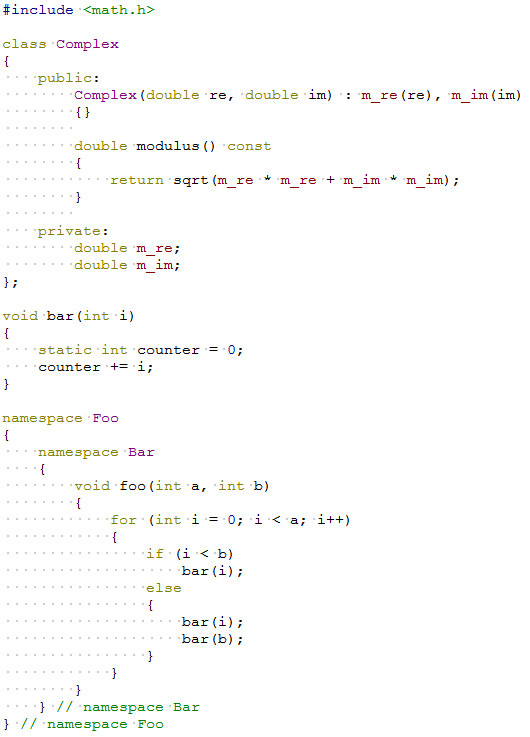


Figure - Juste pour l'indentation le reste est moche

Voici un exemple d'indentation d'un code :

- Les accolades :

Mises au retour à la ligne et au même niveau d'indentation lors de l'ouverture d'un scope :

* Fonction
* Classe
* if/else
* while/for/etc...

Tout ce qui est inclus dans les accolades est décalé d'une tabulation afin de pouvoir clairement identifier le début d'un scope et sa fin.

- Visibilité dans une classe

Les mots clés de visibilité dans la déclaration d'une classe sont donc décalés d'une tabulation par rapport aux accolades du scope de la classe.

Les déclarations sous les mots clés de visibilité sont également indentés d'une tabulation afin de bien identifier les différents niveaux de visibilité (public, protected, private).

## Variables, attributs et méthodes

Toutes les variables, les attributs, les méthodes, les classes, les fonctions sont nommées en anglais.

Les noms de variables doivent respecter une syntaxe particulière :

Pas d' "\_" dans les noms de variables (sauf pour les attributs voir : Dans une classe)

Les mots composant le nom de la variables prennent une majuscule (très utile avec QtCreator et le Ctrl + Effacer !!)

ex : double MyVerySexyVariable

Bien évidement les noms de variables doivent être le plus clair possible. Par exemple dans le cadre du projet un objet contenant la configuration d'un fichier wave : WaveformConfig

Ou la fonction d'affichage du signal sur le visualisateur en face avant : void DisplaySignal(Signal SignalToDisplay)

### Dans une classe :

Dans une classe les méthodes respectent les même règles que pour les fonctions.

Les attributs dans une classe sont nommés comme les variables mais on rajoute devant le préfixe "m\_" synonyme de "my". Ce préfixe permet de bien différencier les attributs des variables locales et permettent de clarifier les noms des variables passées en paramètre.

Concernant les noms des méthodes pour les getters/setters (ou accesseurs) :

Un get porte le nom de l'attribut sans le préfixe "m\_".

Un set porte le nom de l'attribut auquel on ajoute le préfixe "set"

Toutes les classes seront séparés en 2 fichiers : .h et un .cpp  
le .h contiendra la déclaration de la classes et toute les inclusions nécessaire à la classe  
le .cpp contient l'implémentation de tout les méthodes de la classe.

De plus les noms des classes sera préfixé par trois lettres correspondant au projet auquel la classe appartient.

ex : GBF\_Signal, GBF\_Generator, GBF\_...

Enfin le nom des fichiers portera le même nom que la classe.

exemple : Dans lequel j'ai oublier les préfixe GBF\_ !!!!

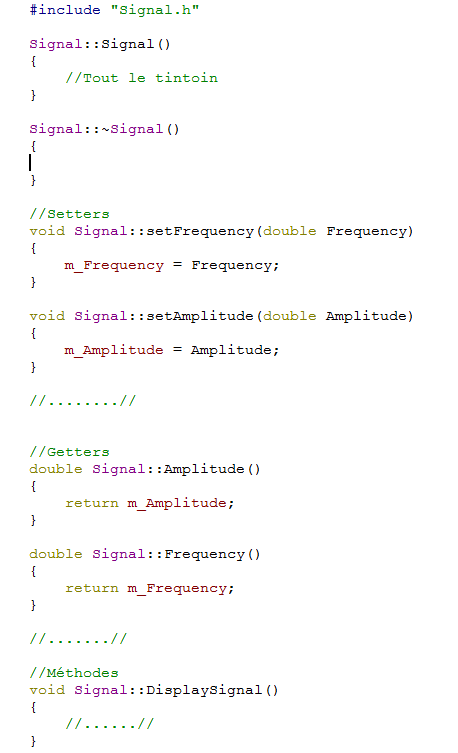


Figure - Signal.cpp

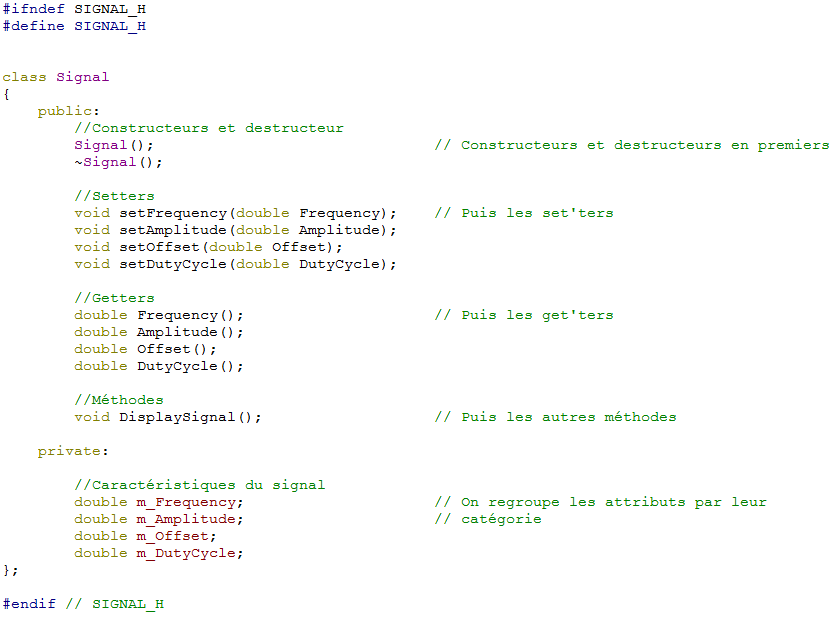


Figure - Signal.h