

# Niveau 3

TP  
S2. Gestion de projet  
S2.2 - Documentation  
Outils de génération de documentation logicielle  
S4.10 - Qualité logicielle  
Documentation des programmes (cf. S2.2)

## Les commentaires d'Andy

### TABLE DES MATIÈRES

1 - Présentation.....	2
2 - dOxygen.....	2
3 - Qt.....	2
4 - Le programme.....	3
5 - Documentation.....	3
6 - Consolidation.....	3

Commentaire :

**/@@@ @FILE MAIN.CPP**

**@**

**@BRIEF DERNIÈRE ÉCHÉANCE ?**

**@**

**@AUTHOR ANDY**

**@/**

écris ton  
commentaire  
dans le cadre  
(au seutre  
noir)

# 1 - PRÉSENTATION

Le but ultime de ce TP est la prise en main de la chaîne de documentation automatique **dOxygen**.

Ce logiciel permet de générer une documentation standardisée en fonction des commentaires que vous aurez inscrits directement dans votre code.

De plus, nous allons aussi introduire une arborescence de directories spécifique pour chaque projet que nous auront à utiliser par la suite.

La plupart des manipulations et des indications sur la façon de faire sont fournies dans le document annexe « Documenter son code avec **dOxyGen** ».

Il vous est donc essentiel de l'avoir à vos côtés en permanence !

# 2 - DOXYGEN

- 1) Validez que la chaîne de documentation **dOxygen** est bien installée sur votre poste. Pour cela, recherchez le directory correspondant sur le disque C. Vous y trouverez la commande en ligne **doxygen.exe** ainsi que le wizard graphique **doxywizard.exe**.
- 2) Dans un premier temps, lancez le wizard. Configurez-le pour générer la documentation de l'exemple qui vous est fourni dans le dossier **Test**. La documentation devra être générée dans le directory **Test/Doc**.  
Pour valider le fonctionnement de **dOxygen**, ouvrez la page **index.html** créée.
- 3) Créez l'arborescence d'un projet C++ comme indiquée dans le document annexe.
- 4) Générez le fichier de configuration de **dOxygen** comme indiqué au chapitre « CONFIGURER LA DOCUMENTATION » puis placez ce fichier dans le directory dédié à la documentation. Profitez-en pour créer un logo propre à votre projet, et placez-là au bon endroit de l'arborescence.
- 5) Au moyen du wizard, chargez ce fichier et paramétrez les options minimales de configuration comme indiquées.
- 6) Passez en mode expert et validez la configuration avancée.

Voilà, votre fichier de configuration est prêt !

Vous pourrez le réutiliser pour les projets suivants... après y apporter des modifications minimales !

# 3 - QT

- 7) Après avoir lancé Qt Creator, créez un nouveau projet tel qu'indiqué au chapitre « PRÉPARATION D'UN PROJET C++ AVEC QT ». N'oubliez pas la structure des sous-dossiers et de configurer les options de **shadow build** et d'**include** !

Il est possible de générer la documentation directement dans Qt, de deux façons différentes. Pour ce TP, on utilisera exclusivement la génération durant la phase de compilation pour que la configuration de Qt ne soit pas modifiée d'un groupe à l'autre. Par la suite, vous pourrez intégrer la génération dans les outils externes de Qt.

- 8) Ajoutez une étape de compilation supplémentaire à votre projet comme montré au chapitre « DURANT LA PHASE DE COMPILATION ».
- 9) Compilez votre projet, qui est encore vide de tout code. Vous devriez obtenir un site de documentation standard, ne montrant que le fichier `main.cpp` !

## 4 - LE PROGRAMME

- 10) Andy Fikulté a emprunté à Guy Mauve une somme de 2 500 € (prêt sans intérêts). Pour rembourser son ami, il prévoit de lui remettre 110 € par mois. Mais comme cela ne correspond pas à un nombre pile-poil de mois, il se demande quel sera le montant à rembourser le dernier mois. Écrivez **l'algorithme** permettant de résoudre le problème posé. Faites-le valider par l'enseignant.

## 5 - DOCUMENTATION

- 11) Nous allons commencer par la description du projet. Préparez un nouveau fichier de type texte, nommé `description.txt`, au moyen de l'éditeur que vous souhaitez. Complétez-le avec les indications données au chapitre « LE DESCRIPTIF DU PROJET ». Vous devrez adapter ces informations en fonction de chaque projet !
- 12) Ajoutez le commentaire d'en-tête du fichier cpp.
- 13) Ajoutez le commentaire de la fonction `main()`, sur le modèle fourni au chapitre des méthodes.
- 14) **Codez le programme** en appliquant les normes de programmation de la section **SN-IR**. Vous respecterez les noms de variables, les indentations et vous commenterez chaque entrée tel qu'indiqué dans le document annexe.
- 15) Finalement, vous validerez la documentation générée lors de la compilation.

## 6 - CONSOLIDATION

- 16) Reprenez deux des exercices réalisés durant les TP précédents et adaptez-les de façon à respecter :
  - la structure des directories ;
  - les règles de nommages des variables ;
  - les indentations et structures itératives et conditionnelles ;
  - les commentaires : la description du programme, les commentaires d'en-tête de fichier, de fonction et des variables.