Génération de documentation technique

en s'appuyant sur un programme simple suivi avec Git-SCM (Git)

Si6 – Développement d'applications

version en date du 2015-01-30 17:55:15 +0100, révision 325872c

Grégory David

Lycée André Malraux
3 rue de Beau Soleil
72700 Allonnes

Table des matières

1 Couverture pédagogique

Activités support de l'acquisition des compétences

D1.1 – Analyse de la demande
- A1.1.1 Analyse du cahier des charges d'un service à produire

D4.1 – Conception et réalisation d'une solution applicative
- A4.1.7 Développement, utilisation ou adaptation de composants logiciels
- A4.1.9 Rédaction d'une documentation technique

D5.2 – Gestion des compétences
- A5.2.1 Exploitation des référentiels, normes et standards adoptés par le prestataire informatique
- A5.2.2 Veille technologique

Savoirs-faire

- Élaborer un jeu d'essai
- Valider et documenter une application
- Utiliser des outils de travail collaboratif

Savoirs associés

- Architectures applicatives : concepts de base
- Techniques de présentation des données et des documents
- Bonnes pratiques de documentation d'une application

2 Travail à faire

1. Créer le dépôt Git sur la plateforme GitHub.com (GitHub)
2. Écrire le code source permettant de répondre aux objectifs définis dans la section 3
3. Suivre le workflow Git afin d'assurer l'*indexation*, la *validation* et le *poussage* vers le dépôt central sur GitHub
4. Rédiger la documentation technique en respectant la syntaxe des commentaires Doxygen[1]
5. Refaire depuis l'item 2 tant que nécessaire
6. Générer la documentation en HTML à l'aide du compilateur Doxygen

3 Cahier des charges fonctionnel

Le programme doit permettre de réaliser les calculs définis dans la Table 1 page suivante, et être écrit en C++.

opérateur	fonction	int32_t	int64_t	float	double
+	type ajouter(type operandeA, type operandeB)	✓	✓	✓	✓
−	type soustraire(type operandeA, type operandeB)	✓	✓	✓	✓
×	type multiplier(type operandeA, type operandeB)	✓	✓	✓	✓
÷	type diviser(type operandeA, type operandeB)	✓	✓	✓	✓
mod	type modulo(type operandeA, type operandeB)	✓	✓	-	-

Table 1 – Définition des fonctions nécessaires au programme, voir exemple d'implémentation dans le Listing 1

Vous disposez du compilateur `g++` pour effectuer la compilation du code source en un fichier binaire exécutable. Voici un exemple de compilation du fichier `main.cpp` suivi de son exécution :

```
lambda@linux:~/repertoireProjet> ls
main.cpp
lambda@linux:~/repertoireProjet> g++ -o calculatrice main.cpp
lambda@linux:~/repertoireProjet> ./calculatrice
```

La documentation devra être générée en HTML à l'aide de Doxygen et son utilitaire de simplification de configuration : `doxywizard`[2]. Vous écrirez la documentation selon l'usage des commentaires Doxygen choisi parmi ceux proposés dans la documentation de Doxygen : http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html

Listing 1 – Exemple de documentation de fonctions avec la syntaxe Doxygen

```
/// \brief Fonction d'addition de deux nombre de type int64_t
///
/// La fonction d'addition est polymorphe, c'est à dire qu'elle
/// édpend du type des èparamtres fournis. Celle-ci correspond
/// au type int64_t.
///
/// \param operandeA un entier 64bits
/// \param operandeB un entier 64bits
/// \return La somme des deux entiers 64bits fournis en èparamtres
/// \sa ajouter(int32_t, int32_t), ajouter(double, double), ajouter(float, float)
///
int64_t ajouter(int64_t operandeA, int64_t operandeB)
{
  return operandeA + operandeB;
}

/// \brief Fonction d'addition de deux nombre de type int32_t
///
/// La fonction d'addition est polymorphe, c'est à dire qu'elle
/// édpend du type des èparamtres fournis. Celle-ci correspond
/// au type int32_t.
///
/// \param operandeA un entier 32bits
/// \param operandeB un entier 32bits
/// \return La somme des deux entiers 64bits fournis en èparamtres
/// \sa ajouter(int64_t, int64_t), ajouter(double, double), ajouter(float, float)
///
int32_t ajouter(int32_t operandeA, int32_t operandeB)
{
  return operandeA + operandeB;
}
```

A Annexes

A.1 Documentation générée en HTML

La configuration de Doxygen pour la sortie HTML :

Listing 2 – Extrait de la configuration HTML de Doxygen du fichier `Doxyfile`

```
GENERATE_HTML          = YES
HTML_OUTPUT            = html
HTML_FILE_EXTENSION    = .html
HTML_HEADER            =
HTML_FOOTER            =
HTML_STYLESHEET        =
HTML_EXTRA_STYLESHEET  =
HTML_EXTRA_FILES       =
HTML_COLORSTYLE_HUE    = 0
HTML_COLORSTYLE_SAT    = 0
HTML_COLORSTYLE_GAMMA  = 80
HTML_TIMESTAMP         = YES
HTML_DYNAMIC_SECTIONS  = NO
HTML_INDEX_NUM_ENTRIES = 100
GENERATE_HTMLHELP      = NO
```

La documentation ainsi générée est un ensemble de fichiers HTML, CSS, PNG et JavaScript. La lecture de la documentation passe alors obligatoirement par un navigateur Web.

Listing 3 – Contenu du répertoire de la documentation HTML

```
bc_s.png
bdwn.png
closed.png
doxygen.css
doxygen.png
dynsections.js
exemple_8cpp.html
exemple_8cpp.js
files.html
files.js
ftv2blank.png
ftv2doc.png
ftv2folderclosed.png
ftv2folderopen.png
ftv2lastnode.png
ftv2link.png
ftv2mlastnode.png
ftv2mnode.png
ftv2node.png
ftv2plastnode.png
ftv2pnode.png
ftv2splitbar.png
ftv2vertline.png
globals_func.html
globals.html
graph_legend.html
graph_legend.md5
graph_legend.png
index.html
jquery.js
nav_f.png
nav_g.png
nav_h.png
navtree.css
navtreeindex0.js
navtree.js
open.png
resize.js
search
```

```
40 sync_off.png
41 sync_on.png
42 tab_a.png
43 tab_b.png
44 tab_h.png
45 tabs.css
46 tab_s.png
```

Référence du fichier exemple.cpp

Aller au code source de ce fichier.

Fonctions

int64_t	**ajouter** (int64_t operandeA, int64_t operandeB) Fonction d'addition de deux nombre de type int64_t. Plus de détails...
int32_t	**ajouter** (int32_t operandeA, int32_t operandeB) Fonction d'addition de deux nombre de type int32_t. Plus de détails...

Documentation des fonctions

int64_t ajouter (int64_t operandeA,
int64_t operandeB
)

Fonction d'addition de deux nombre de type int64_t.

La fonction d'addition est polymorphe, c'est à dire qu'elle dépend du type des paramètres fournis. Celle-ci correspond au type int64_t.

Paramètres

operandeA un entier 64bits

operandeB un entier 64bits

Renvoie

La somme des deux entiers 64bits fournis en paramètres

Voir également

ajouter(int32_t, int32_t), ajouter(double, double), ajouter(float, float)

```
13 {
14   return operandeA + operandeB;
15 }
```

int32_t ajouter (int32_t operandeA,
int32_t operandeB
)

Fonction d'addition de deux nombre de type int32_t.

La fonction d'addition est polymorphe, c'est à dire qu'elle dépend du type des paramètres fournis. Celle-ci correspond au type int32_t.

Paramètres

operandeA un entier 32bits

operandeB un entier 32bits

Renvoie

La somme des deux entiers 64bits fournis en paramètres

Voir également

ajouter(int64_t, int64_t), ajouter(double, double), ajouter(float, float)

```
29 {
30   return operandeA + operandeB;
31 }
```

A.2 Documentation générée en LaTeX

La configuration de Doxygen pour la sortie LaTeX :

Listing 4 – Extrait de la configuration LaTeXde Doxygen du fichier `Doxyfile`

```
GENERATE_LATEX          = YES
LATEX_OUTPUT            = latex
LATEX_CMD_NAME          = latex
COMPACT_LATEX           = NO
LATEX_HEADER            =
LATEX_FOOTER            =
LATEX_EXTRA_FILES       =
LATEX_BATCHMODE         = YES
LATEX_HIDE_INDICES      = NO
LATEX_SOURCE_CODE       = NO
LATEX_BIB_STYLE         = plain
PERLMOD_LATEX           = NO
```

La documentation ainsi générée est un ensemble de fichiers LaTeXet un fichier de construction : `Makefile`. La lecture de la documentation ne peut se faire que si l'on compile les fichiers LaTeXen une sortie PDF par exemple.

Listing 5 – Contenu du répertoire de la documentation LaTeX

```
doxygen.sty
exemple_8cpp.tex
files.tex
Makefile
refman.aux
refman.idx
refman.ilg
refman.ind
refman.log
refman.out
refman.pdf
refman.tex
refman.toc
```

Listing 6 – Instruction pour compiler la documentation LaTeXen PDF

```
lambda@linux:~/repertoireProjet> cd doc/latex
lambda@linux:~/repertoireProjet> make
lambda@linux:~/repertoireProjet> evince refman.pdf
```

Chapitre 2

Documentation des fichiers

2.1 Référence du fichier exemple.cpp

Fonctions

- int64_t ajouter (int64_t operandeA, int64_t operandeB)
  Fonction d'addition de deux nombre de type int64_t.
- int32_t ajouter (int32_t operandeA, int32_t operandeB)
  Fonction d'addition de deux nombre de type int32_t.

2.1.1 Documentation des fonctions

2.1.1.1 int64_t ajouter (int64_t *operandeA*, int64_t *operandeB*)

Fonction d'addition de deux nombre de type int64_t.

La fonction d'addition est polymorphe, c'est à dire qu'elle dépend du type des paramètres fournis. Celle-ci correspond au type int64_t.

Paramètres

operandeA	un entier 64bits
operandeB	un entier 64bits

Renvoie

La somme des deux entiers 64bits fournis en paramètres

Voir également

ajouter(int32_t, int32_t), ajouter(double, double), ajouter(float, float)

```
13 {
14   return operandeA + operandeB;
15 }
```

2.1.1.2 int32_t ajouter (int32_t *operandeA*, int32_t *operandeB*)

Fonction d'addition de deux nombre de type int32_t.

La fonction d'addition est polymorphe, c'est à dire qu'elle dépend du type des paramètres fournis. Celle-ci correspond au type int32_t.

Paramètres

operandeA	un entier 32bits
operandeB	un entier 32bits

Renvoie

La somme des deux entiers 64bits fournis en paramètres

Voir également

ajouter(int64_t, int64_t), ajouter(double, double), ajouter(float, float)

```
29 {
30   return operandeA + operandeB;
31 }
```

Références

Webographiques

[1] *Doxygen : Main Page*. URL : http://www.stack.nl/~dimitri/doxygen/index.html (visité le 30/01/2015) (cf. p. 3).

[2] *Doxygen Manual : Doxywizard usage*. URL : http://www.stack.nl/~dimitri/doxygen/manual/doxywizard_usage.html (visité le 30/01/2015) (cf. p. 4).

Glossaire

Doxygen est un générateur de documentation sous licence libre capable de produire une documentation logicielle à partir du code source d'un programme. Pour cela, il tient compte de la grammaire du langage dans lequel est écrit le code source, ainsi que des commentaires s'ils sont écrits dans un format particulier. 3, 5, 8

workflow Un workflow, anglicisme pour flux de travaux, est la représentation d'une suite de tâches ou opérations effectuées par une personne, un groupe de personnes, un organisme, etc. Le terme flow (flux) renvoie au passage du produit, du document, de l'information, etc., d'une étape à l'autre. Le terme officiellement recommandé par la Commission générale de terminologie et de néologie est « flux de travaux ». Le grand dictionnaire terminologique propose « flux de travaux », « flux des travaux », ou « automatisation des processus ». On trouve aussi l'expression « flux opérationnel » chez des professionnels. 3

Acronymes

Git Git-SCM. 1, 3

GitHub GitHub.com. 3