TP Génie Logiciel

PRÉAMBULE

Les entiers ou les réels sont gérés d'une manière toute particulière lorsque l'on programme. En langage C par exemple, un `int` est la plupart du temps stockée sur 4 octets et un `float` sur 4 octets. Cela implique que leur taille est limitée. Ainsi un `int` signé ne peut dépasser la valeur 2 147 483 647 ($2^{32}/2$) s'il est signé et 4 294 967 294 (2^{32}) s'il est non signé. Les `float` eux sont stockés différemment des `int`. La structure de données est plus complexe. Un `float` dispose d'une mantisse et d'un exposant.

$1,2345 = 12345 \times 10^{-4}$ devient une fois stocké en `float` :

Bit de signe	Exposant								Mantisse																							
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1

Figure 1 Représentation du stockage en mémoire d'un `float` en langage C

Cela signifie que certains calculs ne sont pas possibles en utilisant ces types de données.

Ainsi le programme suivant :

```
int main (void) {
    float i = 123456789.0;
    printf ("i+1-i = %f\n", i+1-i);
    return EXIT_SUCCESS;
}
```

Renvoie un résultat incohérent

```
i+1-i = 0.000000
```

Pour la plupart des situations ces types de données suffisent et si nous avons besoin de nombre plus grand, il existe les types double et long. Cependant, comment faire lorsque la situation demande une précision numérique au 10^{-50} prêt ou bien lorsque le nombre significatif de chiffre d'un entier est de 100 ou 1 000.

benjamin.maurice@telecomnancy.net
jeanaymeric@gmail.com

SOMMAIRE

CAHIER DES CHARGE DU PROJET HUGENUMBERCALCULATOR

Comme vous l'aurez compris l'idée est de construire une calculatrice de très grands nombres.

Afin de permettre le stockage de ces très grands nombres (entiers et réels), ils vont être stockés en utilisant une liste chainée. Chacun des chiffres constituant le nombre sera stocké dans un des maillons de la chaine. Une petite illustration clarifiera les choses :

Voilà comment sera stocké l'entier 1 234 567 :

Figure 2 Illustration du stockage d'un grand nombre dans une liste chainée

Ce n'est ici qu'une ébauche de structure, il vous appartient bien entendu de définir la structure exacte vous permettant en plus des chiffres constituant le grand nombre, de stocker aussi son signe. De la même manière, les chiffres sont peut-être à stocker à l'envers dans la liste chaînée afin de faciliter la programmation des additions, soustraction, multiplication et division.

De plus, pour les réels, il pourrait être judicieux d'utiliser une structure plus complexe. En effet, pour stocker le nombre 10^{-100}, il n'est pas utile d'encombrer la mémoire en enregistrant tous les 0. Cependant, cette structure doit pouvoir permettre de réaliser correctement l'opération $1 - 10^{-100}$

FONCTIONS DE SERVICES

Fonctions principales

- FP1.1 : HugeNumberCalculator doit permettre de réaliser une addition entre deux nombres entiers signés sans limite de nombre de chiffres.
- FP1.1 : HugeNumberCalculator doit permettre de réaliser une addition entre deux nombres entiers non signés sans limite de nombre de chiffres.
- FP1.3 : HugeNumberCalculator doit permettre de réaliser une addition entre deux nombres réels signés ou non signés sans limite de nombre de chiffres.
- FP2.1 : HugeNumberCalculator doit permettre de réaliser une soustraction entre deux nombres entiers signés sans limite de nombre de chiffres.
- FP2.2 : HugeNumberCalculator doit permettre de réaliser une soustraction entre deux nombres entiers non signés sans limite de nombre de chiffres.
- FP2.3 : HugeNumberCalculator doit permettre de réaliser une soustraction entre deux nombres réels signés ou non signés sans limite de nombre de chiffres.
- FP3.1 : HugeNumberCalculator doit permettre de réaliser une multiplication entre deux nombres entiers signés sans limite de nombre de chiffres.
- FP3.2 : HugeNumberCalculator doit permettre de réaliser une multiplication entre deux nombres entiers non signés sans limite de nombre de chiffres.
- FP3.3 : HugeNumberCalculator doit permettre de réaliser une multiplication entre deux réels entiers signés ou non signés sans limite de nombre de chiffres.

- FP4.1 : HugeNumberCalculator doit permettre de réaliser une division euclidienne entre deux nombres entiers signés sans limite de nombre de chiffres.
- FP4.2 : HugeNumberCalculator doit permettre de réaliser une division euclidienne entre deux nombres entiers non signés sans limite de nombre de chiffres.

Fonctions contraintes

- FC1 : HugeNumberCalculator doit être écrit en langage C.
- FC2 : HugeNumberCalculator doit être accompagné d'une documentation Doxygen rédigée en anglais.
- FC3 : HugeNumberCalculator doit être constitué d'un code source écrit en anglais.
- FC4 : HugeNumberCalculator doit être modulaire. Les différentes bibliothèques doivent être le plus indépendantes possibles.
- FC5 : HugeNumberCalculator doit être un applicatif console.
- FC6 : HugeNumberCalculator doit respecter les standards d'écriture de code C.

MODÉLISATION UML DES FONCTIONS DE SERVICE

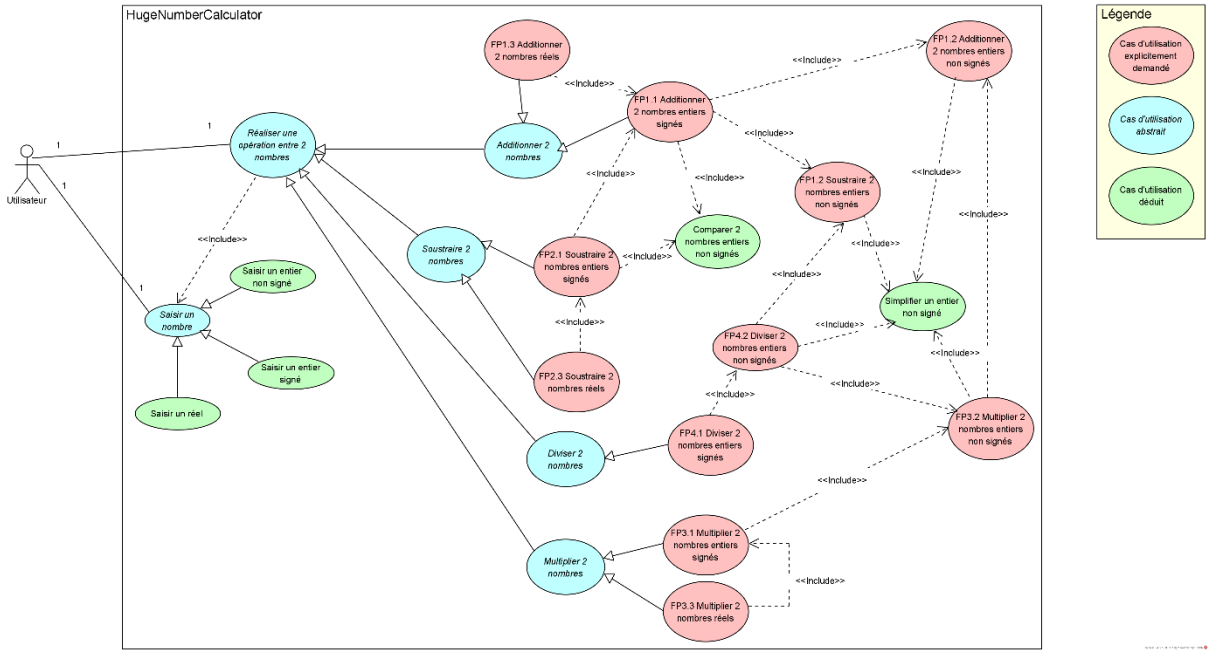

Figure 3 Diagramme de cas d'utilisation HugeNumberCalculator

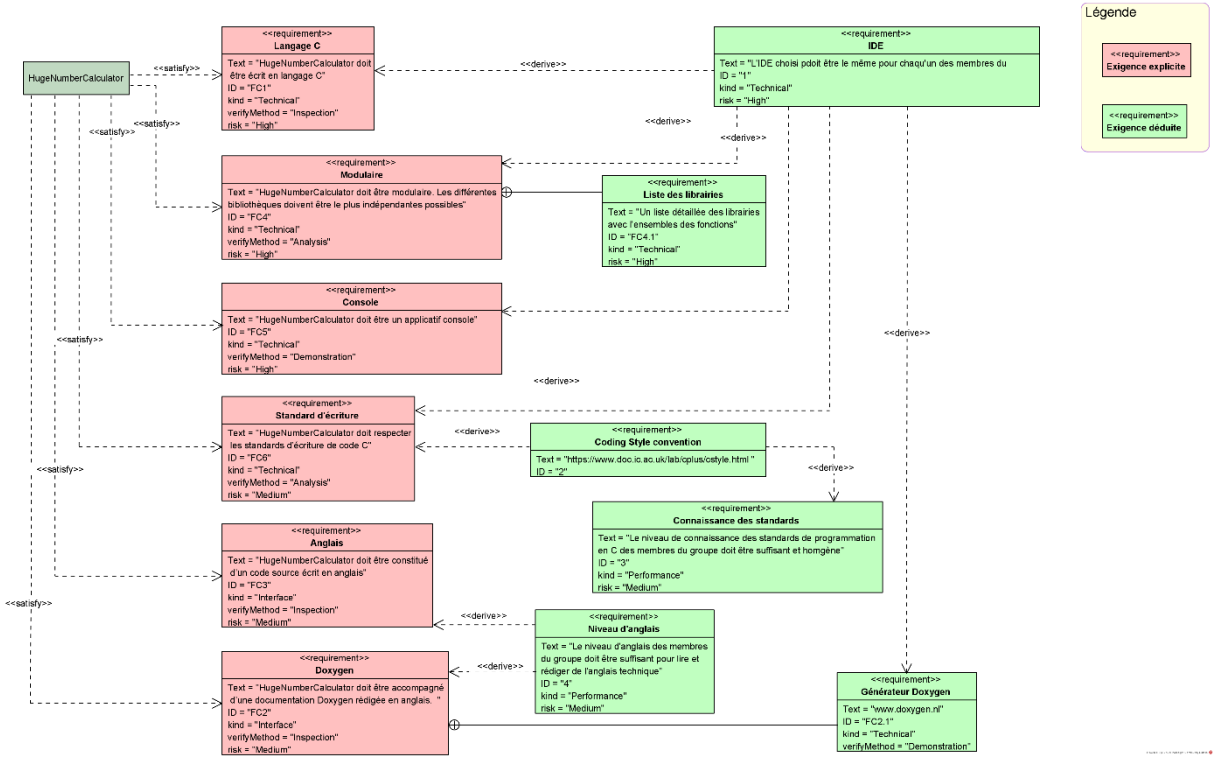

Figure 4 Diagramme d'exigences HugeNumberCalculator

Travail demandé pour le TP1

Une architecture complète et détaillée du programme HugeNumberCalculator. Elle doit contenir :

- La liste de tous les fichiers .h et .c
- Les relations envisagées entre les différents .h
- La signature des fonctions par .h
- Les nouveaux types de données définis

Une répartition des tâches. Cette répartition doit contenir :

- La méthode de travail envisagée
- Les outils utilisés :
    - Choix d'IDE
    - Choix d'outils de travail collaboratif
- La liste des tâches à réaliser et son affectation à un membre du groupe.

Ressources :

Les diagrammes de cas d'utilisation et d'exigences sont accessibles :

- Use Case Diagram.png
- Requirement Diagram.png
- HugeNumberCalculator.vpp.

Le logiciel utilisé pour construire ces diagrammes est Visual Paradigm Community Edition. Le logiciel est téléchargeable ici :

https://www.visual-paradigm.com/download/community.jsp

Attention à bien sélectionner la version Community Edition qui est gratuite, l'autre version offre certes plus de fonctionnalités, cependant elle est payante et se bloque au bout de 30 jours. Étant donné le temps dont vous disposé, nous vous déconseillons de télécharger pendant le TP ce logiciel. De plus, ce logiciel est un logiciel professionnel, aussi sa bonne utilisation demande un temps d'apprentissage assez important. Les images des deux diagrammes sont amplement suffisantes pour le réaliser. Cependant, à l'issu du TP, il peut être intéressant pour vous de vous familiariser avec cet outil de modélisation.