

Université de Cergy-Pontoise

RAPPORT

pour le projet Génie Logiciel
Licence d'Informatique deuxième année

sur le sujet

Projet : UML Converter

rédigé par

FOURGS Alexandre
&
ASTORI Florian



Mai 2015

Table des matières

1	Introduction	2
1.1	Cadre	2
1.2	Pourquoi ce sujet ?	2
2	Spécification	2
2.1	Dessiner son UML	2
2.2	Convertir son UML en code Java	2
2.3	Ecrire un code Java	3
2.4	Convertir un code Java en diagramme UML	3
3	Réalisation	3
3.1	Conception	3
3.2	Outils utilisé(s)	3
3.3	Sur le plan personnel	4
4	Manuel Utilisateur	5
4.1	Comment dessiner son UML	5
4.2	Convertir son UML en code Java ou l'exporter en image	5
4.3	Convertir un code Java en diagramme UML	6
5	Déroulement du projet	7
5.1	Calendrier	7
5.2	De l'idée, au résultat	7
6	Conclusion	7

Remerciements

Les auteurs du projet voudraient remercier M.Liu !

1 Introduction

1.1 Cadre

Dans le cadre du cours de Génie Logiciel durant notre seconde année de licence d'informatique, nous avons été amenés à réaliser un projet de programmation en Java. Etant en binôme, nous avons eu le choix entre 5 projets, nous avons choisis le sujet UML.

1.2 Pourquoi ce sujet ?

Choix Pourquoi ce sujet en particulier ?

- Pour son utilité.
- Un sujet qui concerne un outil que l'on utilise.

Un UML est un outil puissant nous permettant de concevoir et de mieux comprendre le fonctionnement d'un logiciel. Lorsque nous avons un projet en tête, durant la phase de conception nous sommes souvent amenés à passer du temps sur l'UML de notre projet. Ceci dit, le temps que nous passons sur l'UML est du temps en moins sur la réalisation.

A travers ce projet, nous avons vu un outils permettant de concevoir un logiciel tout en le réalisant, car vous n'aurez pas à prendre du temps à retranscrire votre UML en langage de programmation, cet outil le fera pour vous. Tout l'enjeu de ce logiciel se joue ici, le temps et la facilité à transformer votre idée en un début de logiciel, et c'est particulièrement pour cela que nous avons choisis ce sujet.

De plus, au vue de notre situation d'étudiant en informatique et découvrant le monde de la programmation orientée objet depuis peu, nous avons eut l'occasion de faire des UML et de travailler dessus. Cette seconde raison nous a aussi poussé à choisir ce sujet. En effet, nous pensons nous servir de ce logiciel dans le futur.

2 Spécification

Avant de commencer Avant toute action sur le logiciel, l'utilisateur doit créer un nouveau projet.

2.1 Dessiner son UML

L'utilisateur peut, si il le souhaite, dessiner un diagramme UML. Pour cela, plusieurs boutons lui sont proposés. Il peut créer différents packages pour ranger ses classes, il peut également créer des classes, classes abstraites et interfaces qu'il pourra, à leur création ou plus tard, incorporer dans le package de son choix. Dans le moteur du logiciel, il faut savoir que des données sont tout simplement créées et stockées. Initialement, elles sont placées dans le package par défaut. Une fois ses classes, classes abstraites et interfaces créées, il pourra les associer entre elles comme par exemple lier deux classes avec un héritage ou encore une composition. Il peut également, si il le souhaite, implémenter une classe avec une interface.

Si il se trompe, l'utilisateur peut supprimer un lien mais également éditer voir supprimer une classe. Il n'aura pas besoin de supprimer les packages, seuls les packages ayant au moins une classe seront affichés.

2.2 Convertir son UML en code Java

Lors de la conversion de l'UML en code Java, les données qui ont été créées lors de la création des différents packages, classes ... sont "lu" a partir de l'objet "Project" qui contient toutes ces données. Par la suite, il suffit de récupérer, dans ces données, le code correspondant à leur représentation en code Java.

2.3 Ecrire un code Java

Une fois son projet converti en code, l'utilisateur peut commencer un début de code simple sur ces données. On parle ici de code simple car il ne pourra créer directement ni package ni classe à travers l'éditeur de fichier incorporer dans le logiciel.

2.4 Convertir un code Java en diagramme UML

A travers ce dernier outils, l'utilisateur à la possibilité d'effectuer une conversion inverse, il lui suffit de sélectionner le dossier qui contient les packages et les fichiers .java de son projet pour que le moteur du logiciel lise ce code et le décompose afin d'en extraire les données et les représenter sous forme d'UML.

3 Réalisation

3.1 Conception

Au départ de notre projet, nous avons dû penser à chaque données existantes en Java : les classes, les interfaces, les classes abstraites mais aussi les packages.

Mais ce sont surtout les autres données que peuvent contenir ces données qui sont importantes : les attributs, les méthodes, le type de ces données en fonction de la donnée supérieur (Exemple : Une classe abstraite peut contenir des méthodes abstraites), etc...

C'est pourquoi nous avons listé toute ces données et mis en place une hiérarchie de ces données à travers l'UML de conception de notre projet.

De plus, nous avons commencé à réfléchir à des solutions pour la création et le traitement de ces données (affichage, modifications).

La phase de conception nous a aussi amené à effectuer des maquettes de l'interface homme machine. Nous avons tout deux réaliser une maquette de l'interface, nous les avons comparés et nous nous sommes mis d'accord sur une interface qui liés chacunes de nos idées.

3.2 Outils utilisé(s)

En plus de nos idées, voici les principaux outils que nous avons utilisés afin de réaliser notre projet.

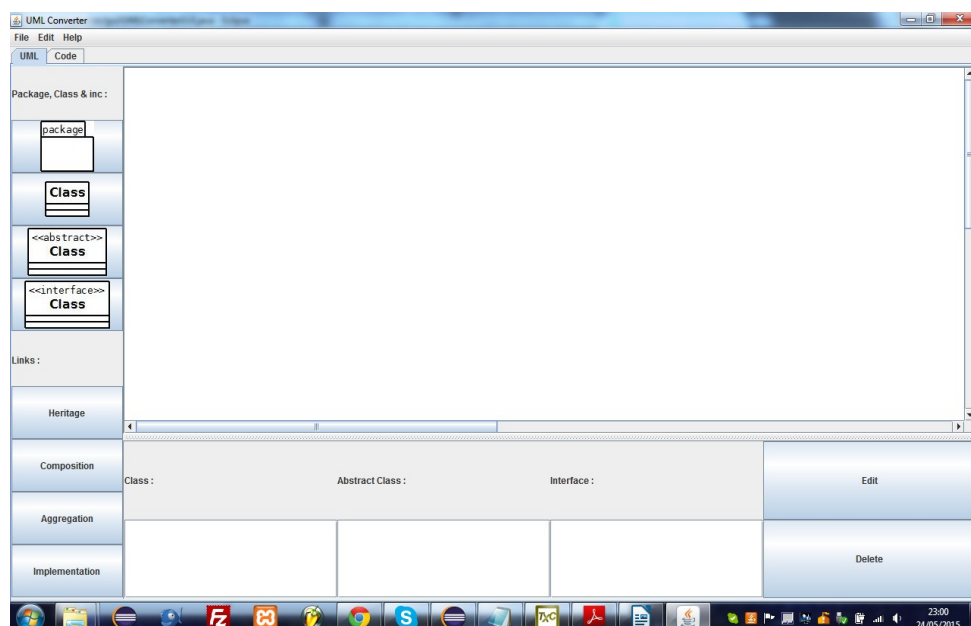
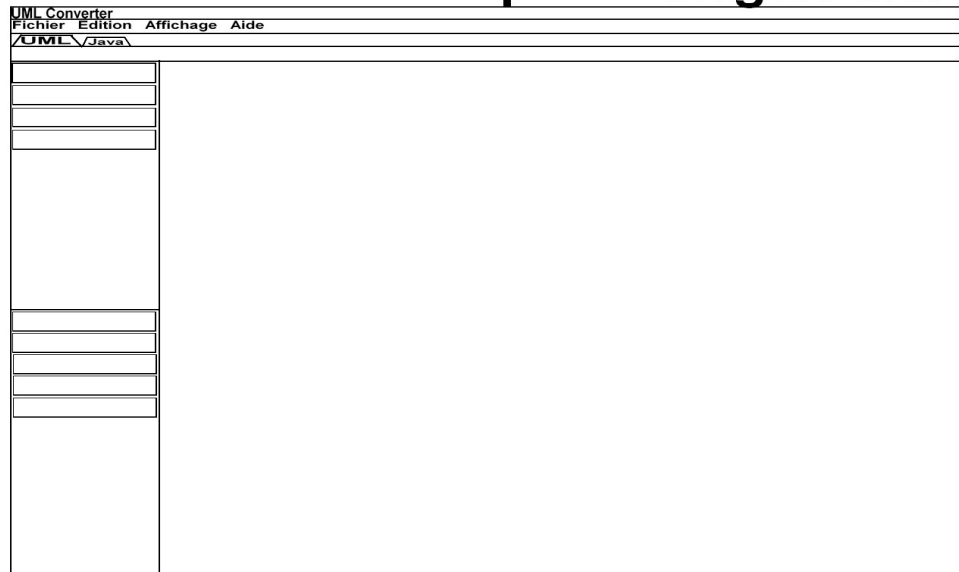
Outils de développement :

1. Java : Langage de programmation orientée objet pour réaliser des logiciels.
2. Eclipse : Logiciel permettant d'écrire notre logiciel, de le compiler et de l'exécuter.
3. Latex : Langage de programmation pour le traitement de texte.

Autres Outils :

1. PhotoFiltre : Logiciel de retouche d'images pour les images dans le logiciel et pour la maquette.

UMLConverter maquette Onglet UML



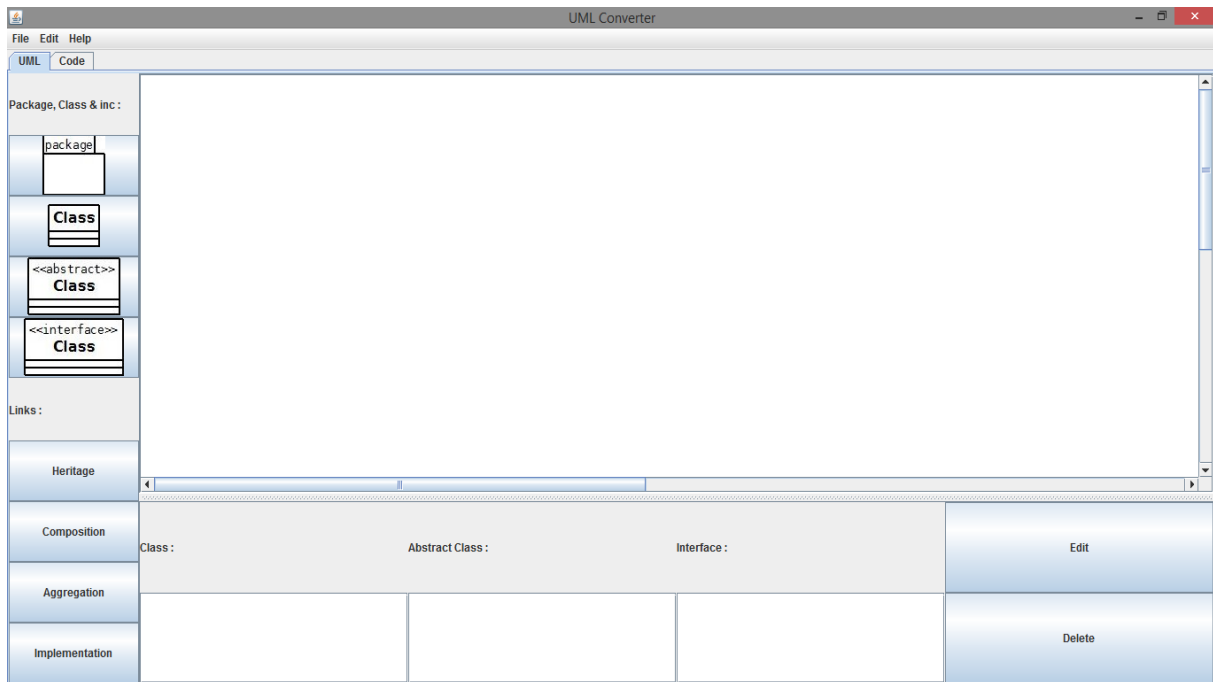
3.3 Sur le plan personnel

1. Alexandre Fourgs : J'ai vraiment été passionné par ce projet, lorsque j'ai vu les choix de sujets en binôme, ce sujet fut mon premier choix sans hésitation. J'ai pris énormément de plaisir à concevoir ce logiciel et à le réaliser. J'ai appris énormément de choses, et surtout, j'ai vraiment pris goût à la programmation orientée objet. Ce projet m'aura aussi permis d'enrichir mes connaissances sur le fonctionnement des langages de programmation orientée objet, mais aussi sur Java.
2. Florian Astori : Bien que ce projet m'intéressait fortement également, j'ai été beaucoup moins passionné que mon mon binôme, même si il est vrai que j'ai pu enrichir mes connaissances en programmation orientée objet, comprenant un peu mieux son fonctionnement et son utilisation.

4 Manuel Utilisateur

Dés l'ouverture l'utilisateur doit aller dans "File" et sélectionner "New Projet".

4.1 Comment dessiner son UML

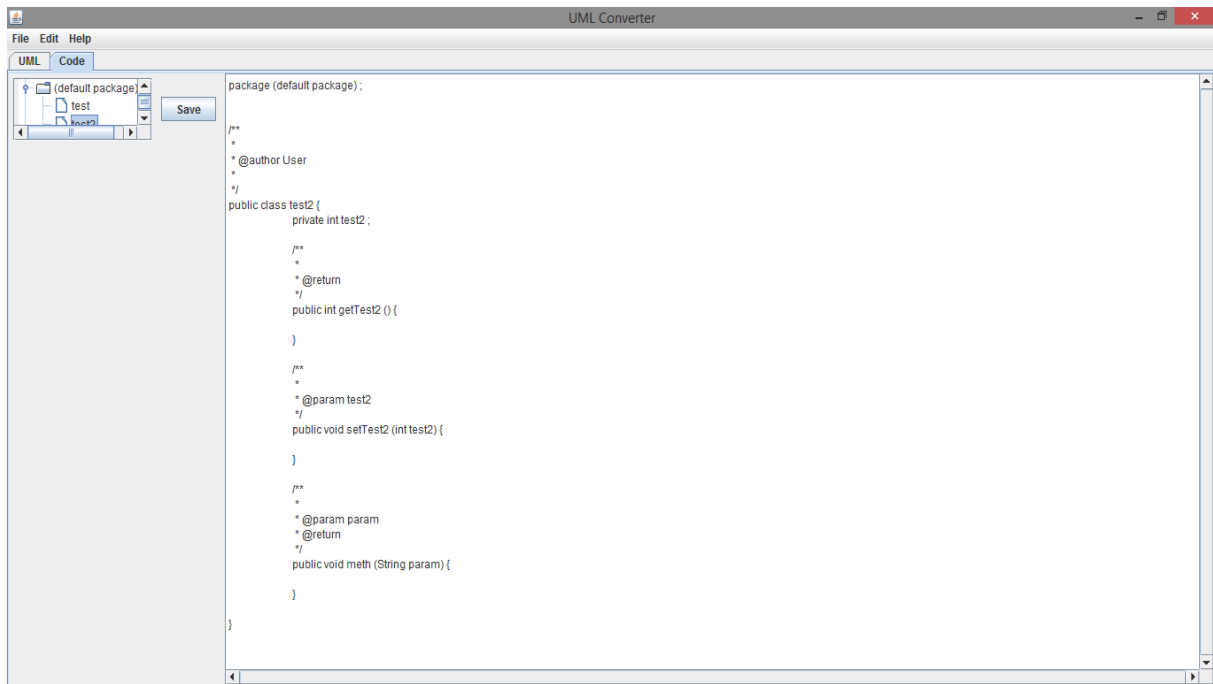


L'utilisateur peut créer des classes, classes abstraites et interfaces à l'aide des boutons présents sur le côté dans la partie "Package, classes & inc". Il peut également créer à l'aide du bouton "Package" un ou plusieurs packages qui serviront à ranger ses classes créées précédemment ou par la suite. Dans la cas où les classes sont créées avant les packages, l'utilisateur dispose d'un menu sous son UML qui lui permettra de supprimer ou d'éditer ses classes et de les placer, à l'aide d'un menu déroulant, dans les packages existants. Si il crée ses classes après ses packages, il pourra les ranger dans les packages voulus lors de la création à l'aide d'un menu déroulant. Le menu sous son UML sera également présent si il souhaite se corriger.

4.2 Convertir son UML en code Java ou l'exporter en image

Après avoir fini son UML, l'utilisateur peut exporter son UML en image ou en convertir en code Java. Pour cela, il lui suffit de sélectionner son choix dans le menu "File".

4.3 Convertir un code Java en diagramme UML



Si l'utilisateur souhaite faire l'opération inverse à celle prévue de base, c'est-à-dire générer un diagramme UML à partir d'un code Java, il doit sauvegarder son code, ouvrir une nouvelle fois le logiciel, créer un nouveau projet et ensuite choisir l'option ".java to UML" dans le menu "File".

5 Déroulement du projet

5.1 Calendrier

1. Janvier : Choix du sujet. Nous avons commencé à concevoir et réaliser la partie donnée de notre logiciel.
2. Février : Conception et début de réalisation de l'interface homme machine. Nous avons réaliser différentes maquettes de l'interface graphique, et nous avons commencé à réaliser le squelette de cette interface. Nous avons placé les différents composant de l'interface.
3. Mars : Réalisation poussé de l'interface homme machine. Nous avons commencé à implémenter les différentes actions dans l'interface, ainsi, nous pouvions créer des classes, des interfaces ou encore des packages et commencer à les remplir avec d'autres données.
4. Avril : Débuggage de certaines fonctionnalités de l'interface, réalisation d'outils simple et début de conception de d'autre outils. En avril, nous avons pu arriver à une interface graphique quasiment opérationnel. Nous avons pu mettre en place des outils comme la sauvegarde de projet, le chargement ou encore l'outil permettant de convertir l'UML en fichiers .java contenant le code représentatif de l'UML. Nous avons pu aussi commencer à réfléchir aux autres outils que notre logiciel contiendrait.
5. Mai : Dernier débbuggage de l'interface, réalisation des outils complexes et rendu du projet. Nous avons conçu et réaliser un convertisseur inversé permettant de convertir un projet (fichiers .java) en un UML ainsi qu'un petit éditeur de texte permettant d'écrire le début de votre projet. Le 24 mai, nous avons rendu notre projet.

5.2 De l'idée, au résultat

Alexandre Fourgs : Il a été en charge de réaliser toute la partie donnée, c'est-à-dire la création des classes représentatives des classes, classes abstraites, etc... Mais aussi des données comme les attributs, les méthodes et toutes les données liées au projet. Il a aussi réaliser le moteur du logiciel, les convertisseurs (UML > Java et Java > UML) ainsi que de la réalisation de l'interface homme machine et l'implémentation des différentes actions lors des interactions avec les différents composants de l'interface graphique.

Florian Astori : Il a travaillé sur la partie graphique du logiciel, c'est-à-dire tous les affichages et calculs nécessaires à la bonne utilisation du logiciel avec la partie UML.

6 Conclusion

Il nous a été demandé de réaliser un logiciel permettant avant de dessiner un diagramme UML et de le convertir en langage Java. Ce premier point a pu être réaliser avec succès. Nous avons par ailleurs réaliser le second point qui consistait à convertir un code Java en diagramme UML.

Ce logiciel peut être amélioré avec la mise en place d'un IDE performant, un position des éléments du diagramme UML à l'aide de la souris, communément appelé "Drag'n drop".