

Rapport de laboratoire

**Département de génie logiciel et des TI**

|  |  |
| --- | --- |
| **No de laboratoire** | Laboratoire 2 |
| **Étudiant(s)** |  |
| **Code(s) permanent(s)** |  |
| **Cours** | LOG121 |
| **Session** | E2015 |
| **Groupe** | 01 |
| **Professeur** |  |
| **Chargés de laboratoire** | **Patrice Boucher** |
| **Date de remise** | 14 juin 2015 |

# 1 Introduction

Dans le cadre de ce laboratoire, nous avons utilisé le premier laboratoire et apporté plusieurs modifications à celui-ci. Le premier objectif du laboratoire est d’encadrer toutes les formes d’un rectangle pointillé. Par la suite, nous devons permettre à l’utilisateur de choisir une méthode de tri. Cette méthode de tri peut soit classer les formes par numéro de séquence, par l’aire de la forme, le type de forme, les trois en ordre croissant ou décroissant ainsi que par distance maximale entre deux points des formes. La méthode retenue pour effectuer ce laboratoire répond à tous les critères d’évaluation. Ce rapport contient deux sections principales soit la conception et les décisions d’implémentation. La première section contient aussi les choix et responsabilités des classes, un diagramme des classes, une description de la conception, les faiblesses de la conception et le diagramme de séquence.

# 2 Conception

## 2.1 Choix et responsabilités des classes

Discutez du choix des classes et de la répartition des tâches. Vous pouvez utiliser un tableau qui résume vos classes importantes.

* *Responsabilités : utilisez des phrases descriptives, ne listez pas de méthodes*
* *Dépendances : les classes de votre application nécessaires au fonctionnement de la classe en question.*

|  |  |  |
| --- | --- | --- |
| *Classe* | *Responsabilités* | *Dépendances* |
|  |  |  |
|  |  |  |

## 2.2 Diagramme des classes

Présentez votre conception logicielle complète appuyée d'un ou plusieurs diagrammes de classes UML.

|  |
| --- |
| **Remarques :**   * *Vous pouvez réaliser le UML avec le logiciel de votre choix ou à la main si c'est fait avec beaucoup de soin (règle et écriture très claire). Si vous utilisez des logiciels qui génèrent automatiquement les UML à partir du code, révisez attentivement l'information, dont les types de relations (dépendance, association, agrégation, composition), lesquels sont souvent inappropriés.* * *N'accompagnez pas le schéma avec un texte explicatif dans cette sous-section, le texte explicatif du schéma est dans la sous-section suivante.* * *N'indiquez pas systématiquement tous les attributs et les méthodes des classes. Privilégier l’information qui démontre le fonctionnement de l’application. Par exemple, certains accesseurs/mutateurs peuvent être implicites.* * *Attention à la résolution : s'il faut zoomer pour comprendre un texte ou un détail, ce dernier doit rester net.* * *Attention aux flèches coupées : si vous séparez votre UML en plusieurs figures, et que des flèches sont coupées entre les figures, indiquez la destination ou la provenance de la flèche coupée.* |

## 2.3 Description de la conception

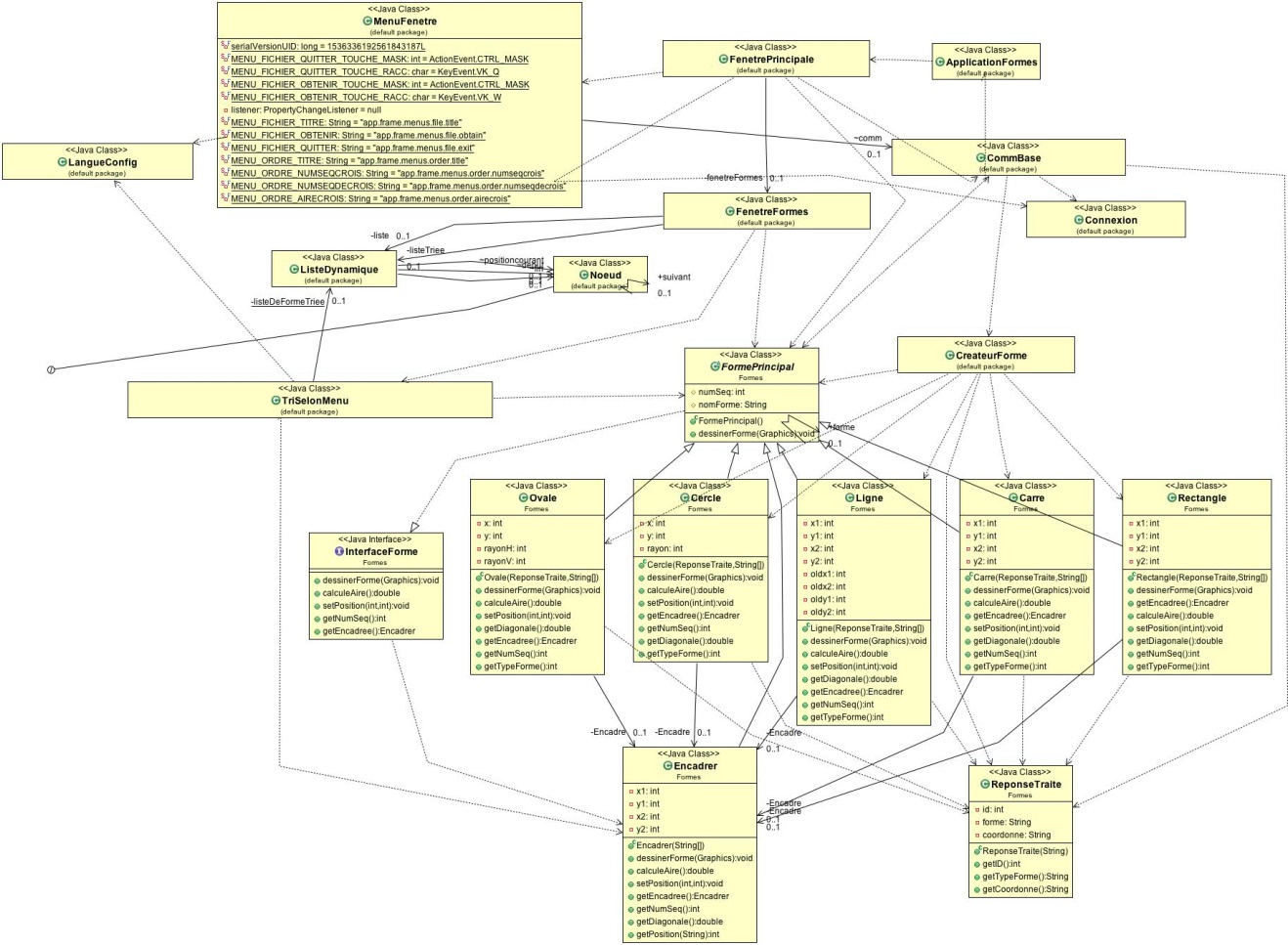
Ce texte explique et justifie la façon dont est fournie chaque fonction du logiciel (1 page maximum).

## 2.4 Faiblesses de la conception

Décrivez les faiblesses de votre conception et des solutions possibles pour remédier à ces points faibles. (maximum 1 page)

|  |
| --- |
| **Remarques :**   * *Essayez d'identifier les problèmes les plus importants.* * *Il faut absolument trouver des problèmes et des améliorations possibles – inutile de mentionner que votre programme est bien fait ;-)* |

## 2.5 Diagramme de séquence (uml)



### 2.5.1. Réception et création des formes

### 2.5.2. Tri des formes selon une option de votre choix

# 3 Décisions d'implémentation

Les décisions d'implémentation concernent la programmation du logiciel conçu à la section précédente, ce qui signifie des décisions liées aux algorithmes, structures de données et librairies utilisées.

|  |
| --- |
| **Remarques :**   * *Choisissez des décisions qui démontrent l'originalité de votre implémentation.* * *Évitez les décisions implicites, telles que celles directement liées aux contraintes de développement. Par exemple, « décider » de respecter l'énoncé est implicite. Une décision pertinente comporte plusieurs possibilités concurrentes pouvant chacune respecter les contraintes de l'énoncé.* |

## 3.1 Décision 1 : titre de la décision d'implémentation

Décrivez, en subdivisant bien votre texte selon chaque aspect :

* **Contexte**: le problème résolu**.**
* **Solution 1**: votre solution d'implémentation 1
* **Solution 2** : votre solution d'implémentation 2
* **Choix de la solution et justification :** quant à l'efficacité, la qualité (clarté, convenance, complétude, cohésion, cohérence), l'extensibilité et/ou l'ergonomie du logiciel, etc.

|  |
| --- |
| **Remarques :**   * *Utilisez au besoin des diagrammes et des pseudo-codes afin de bien décrire vos solutions. Soyez précis.* * *Indiquez des références complètes. Par exemple, si vous citez une méthode, citez dans quelle classe (et même dans quelle librairie selon le cas) elle se trouve.* |

## 3.2 Décision 2 : titre de la décision d'implémentation

# 4 Conclusion

1. Faites un rappel des objectifs du logiciel: ces objectifs ont-ils été atteints?
2. Synthétisez ce qui a été réalisé, les points forts et les points faibles de votre conception.
3. Commentez, rappelez, les améliorations possibles, la portée de votre travail pour d'autres applications et les développements futurs envisageables.

|  |
| --- |
| * *Ne discutez pas du contexte académique et de détails se rapportant au travail d'équipe (si vous éprouvez un problème dans votre équipe, contactez plutôt un chargé de laboratoire).* |

# 5 Références

* Mettez, au besoin, les références