

# IFT2255 – Génie logiciel

## Devoir 3

Vincent Antaki  
Guillaume Poirier-Morency  
Émile Trottier

6 décembre 2014

### Résumé

L’objectif du travail est d’analyser et concevoir un système confronté à une série de changements.

Le projet a été séparé en trois paquetages Java. Pour minimiser l’implantation, les classes non-modifiées sont réutilisées depuis les paquetages précédents.

Le projet est accompagné d’une série de tests JUnit au lieu de spécifier une classe `Main`. C’est plus simple et cela nous permet de soumettre notre implantation à des cas d’utilisation plus réalistes.

## 1 Diagramme de classes initial

## 2 Diagramme de classe modifié

## 3 Diagramme de classe avec Client

## 4 Changements

Pour les deux sous-sections qui suivent :

- identifier les concepts pertinents
- préciser l’impact
- préciser si l’impact est localisé

### 4.1 Première requête de changement

Pour effectuer le changement, nous introduisons la classe `Raccourci` et `ElementRaccourciable`. Un raccourci possède un agrégat vers une instance d’`ElementRaccourciable`.

Il s’agit d’une version modifiée du patron décorateur utilisé pour `ElementDecorator`, car un agrégat le lie à l’élément qu’il pointe.

Les classes `Fichier` et `Dossier` seront impactés, car elles devront désormais hériter de la classe `ElementRaccourciable`.

Il faut aussi déplacer la fonction `taille` de la classe `Element` vers la nouvelle classe `ElementRaccourciable`, car un raccourci ne possède pas de `taille` à proprement parler. On évite notamment un problème cycle pour le calcul de la `taille` d’un dossier dans le cas où un dossier contiendrait un raccourci qui pointerait vers ce même dossier.

### 4.2 Deuxième requête de changement

Pour implanter ce changement, nous faisons implanter l’interface `DeleteObserver` à la classe `Raccourci`, de telle sorte que le raccourci est notifié lorsque son élément est détruit.



FIGURE 1 – Diagramme de classes initial



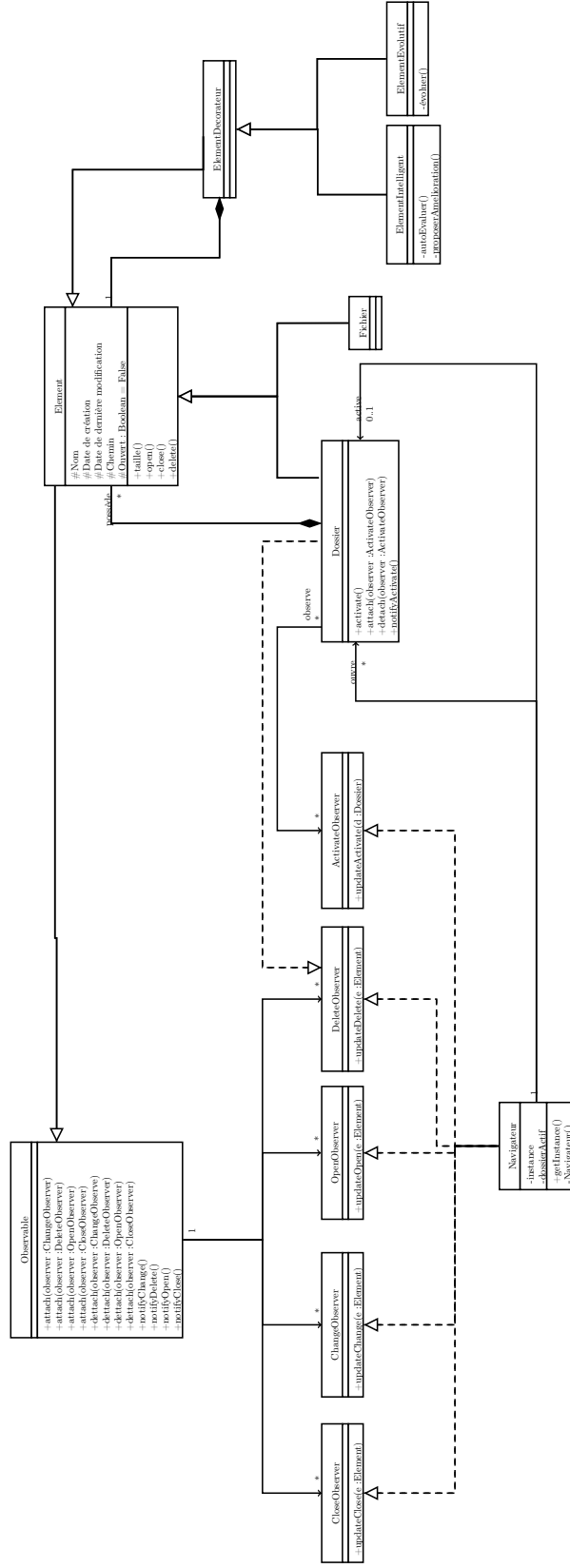


FIGURE 3 – Diagramme de classes

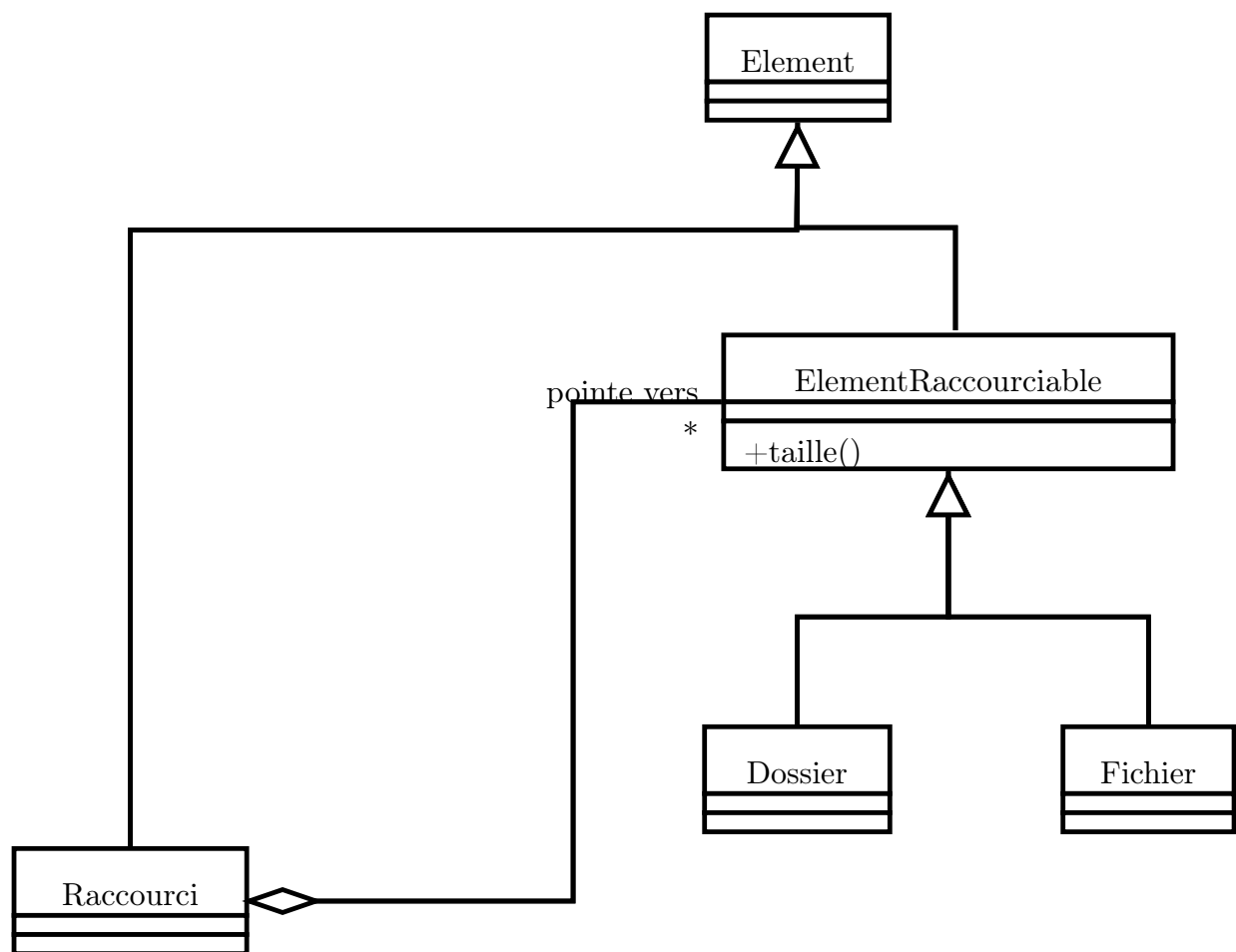


FIGURE 4 – Diagramme de classes pour les raccourcis

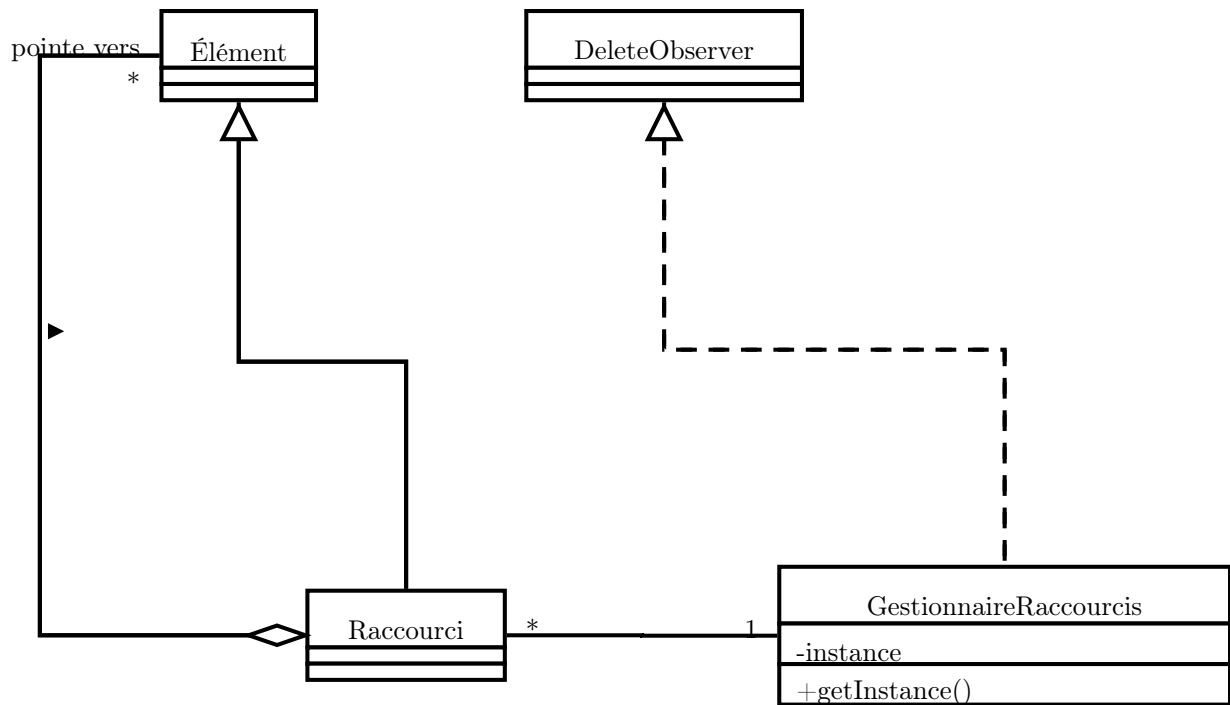


FIGURE 5 – Diagramme de classes pour le gestionnaire de raccourcis

Lorsqu'un raccourcis est créé, il attache le gestionnaire et tant que DeleteObserver de l'élément qu'il pointe. Lorsque l'élément pointé est supprimé, le gestionnaire capture l'événement et est en mesure de supprimer le raccourcis approprié.

L'implantation a pour ensemble d'impact initial la **Element** et l'interface **DeleteObserver**.