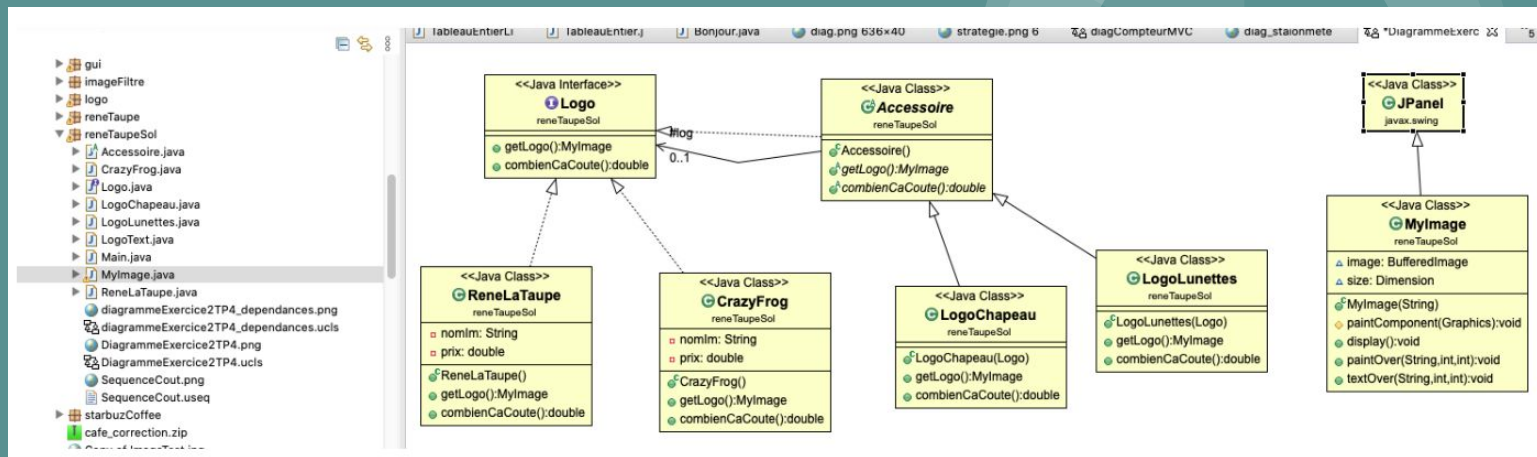


Rapport SAE Développement d'applications



DEMARQUE Amaury
FAURE Solène
HUGOT Benjamin
LIM HOUN TCHEN Aimé



Sommaire

1. Introduction
2. Itérations
3. Conclusion



1. Introduction

Objectif :

Reproduire un logiciel ressemblant au plugin ObjectAid sur Eclipse avec JavaFX

Travail à faire :

Concevoir le projet sous forme de diagrammes

Se répartir les différentes fonctionnalités à réaliser pour ensuite les regrouper



2.1 Itération 1

Fonctionnalités faites :

- Création d'une classe
- Affichage d'une classe dans un terminal

Fonctionnalités en cours de création:

- Création d'une classe à partir d'un fichier distant
- Début de l'interface graphique



2.2 Itération 2

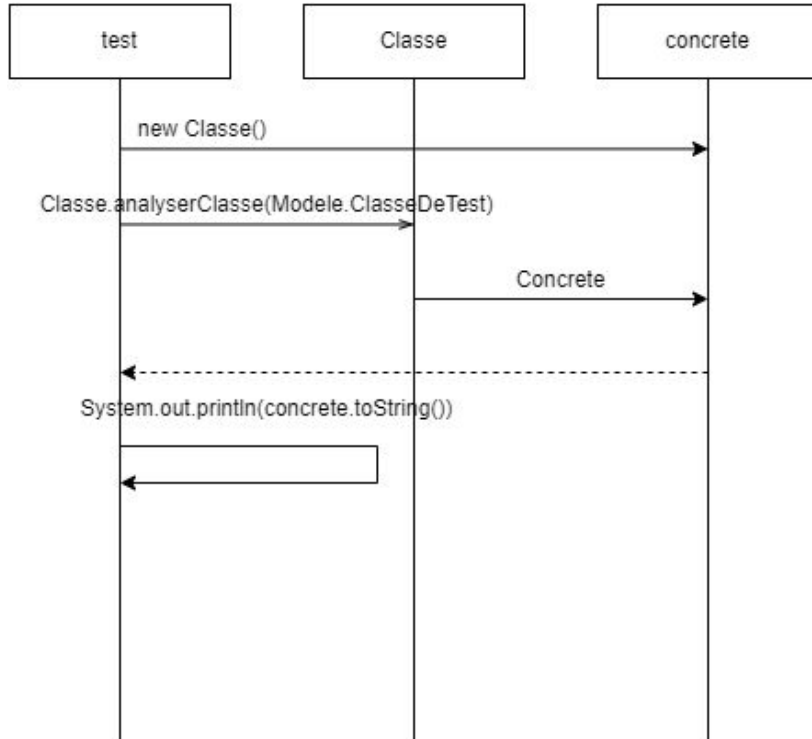
Fonctionnalités faites :

- Détection de la classe parent et des interface implémentées d'une classe
- Affichage d'une classe dans un terminal selon le type (Abstract, Interface ou Concrete)
- Possibilité de glisser et de déposer un fichier dans l'interface puis de le déplacer

Fonctionnalités en cours :

- Création d'une classe à partir d'un fichier

Fonctionnalité affichage Terminal.



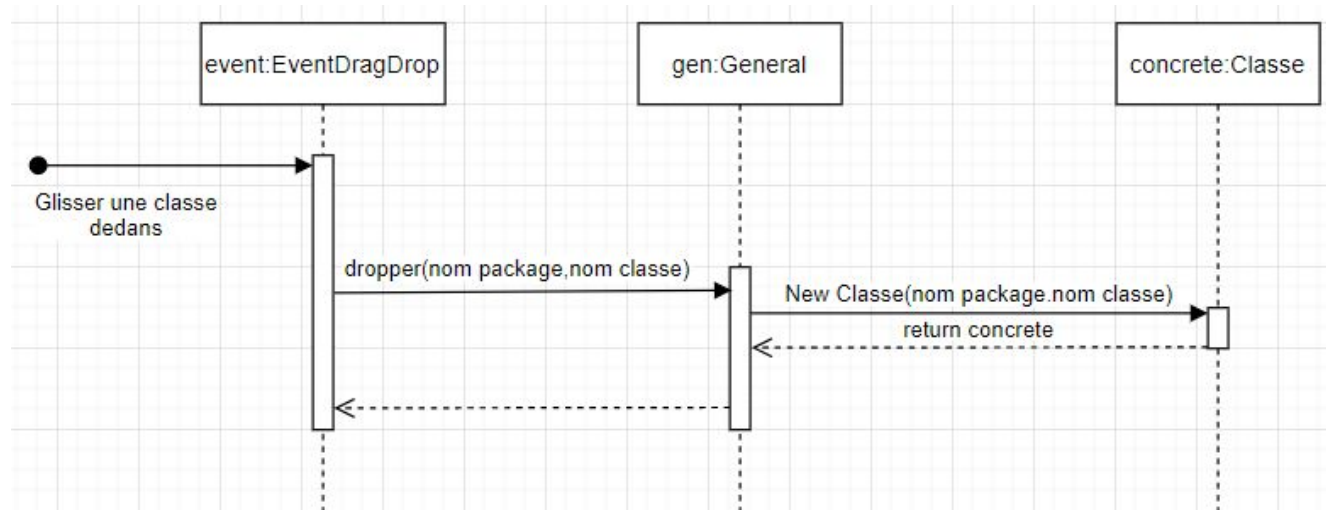
```
<<Java Class>>
ClasseConcreteDeTest
ClasseTest

-----
[private static b:String, protected final c:boolean, public a:int]
-----
[private ClasseConcreteDeTest(int), protected ClasseConcreteDeTest(int,String), public ClasseConcreteDeTest()]
-----
[private f2(int):int, protected f3(int,int,String):String, public f1():void]
-----

Parents de ClasseConcreteDeTest
<<Java Class>>
Object
java.lang

-----
[]
-----
[]
-----
[]
-----
```

Fonctionnalité Glisser et déposer.



Montre le nom pour le moment.



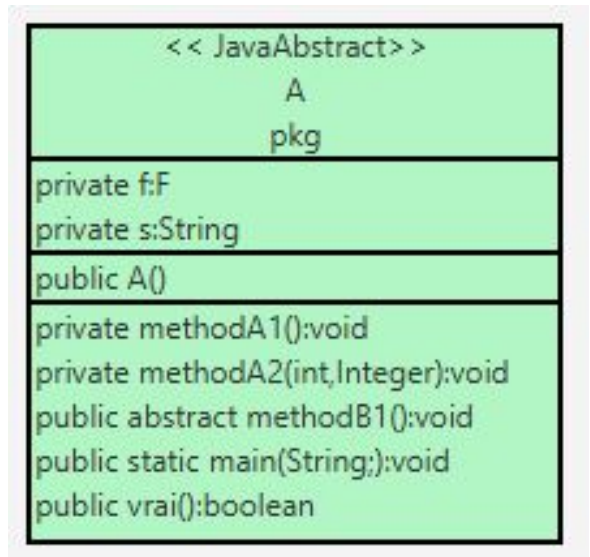
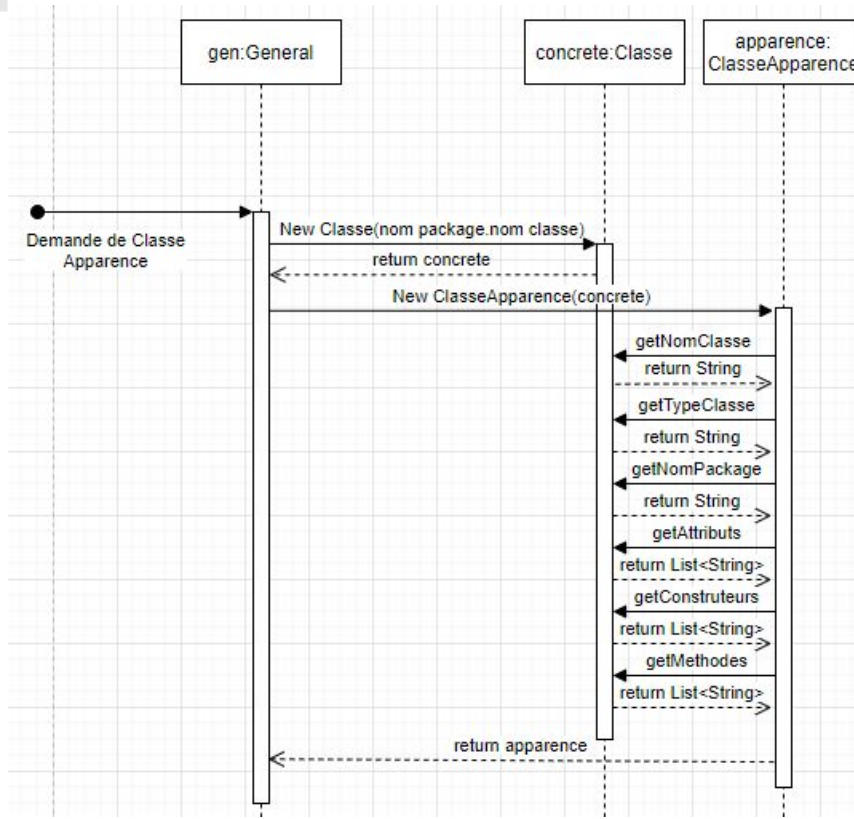


2.3 Itération 3

Fonctionnalités faites :

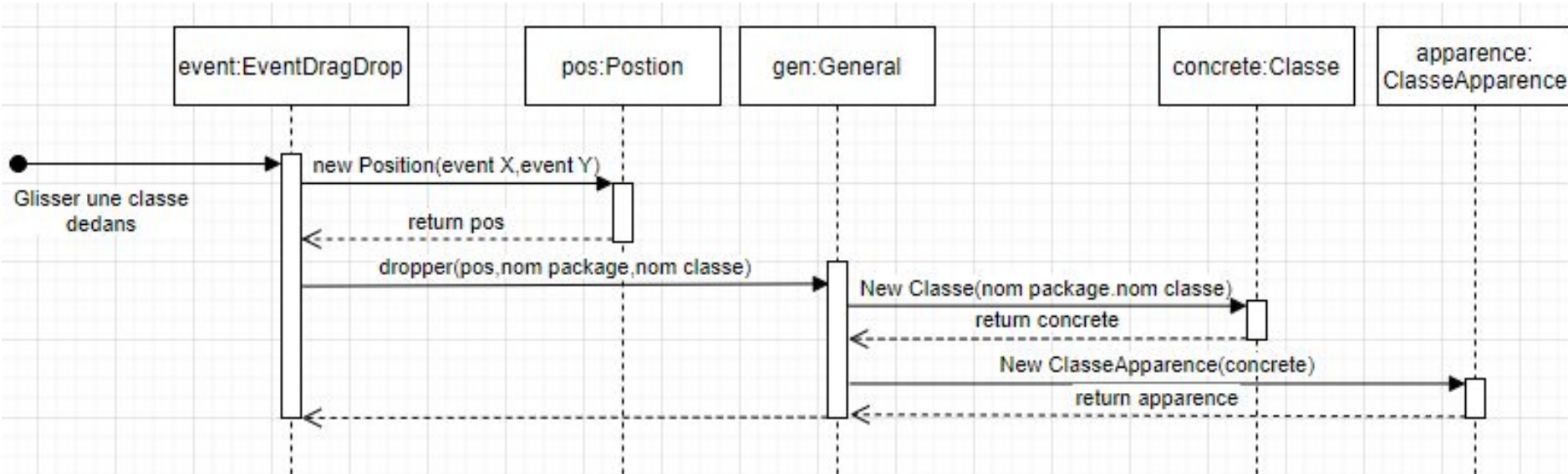
- Création de l'apparence d'une classe dans l'interface
- Actualiser Glisser Déposer avec l'apparence
- Revisite de l'affichage d'une classe dans le terminal

Fonctionnalité création d'apparence.





Fonctionnalité Glisser et Déposer (ajout Apparence).



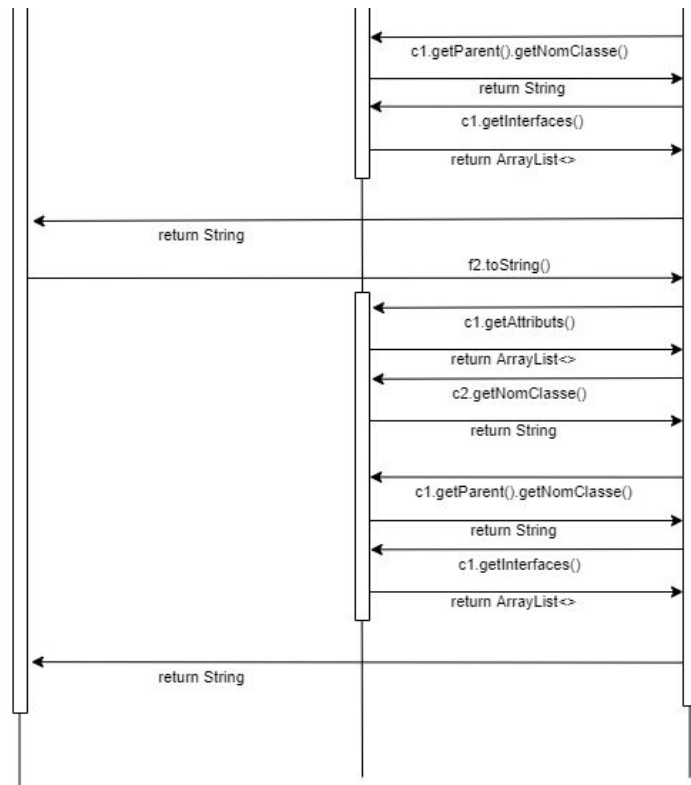
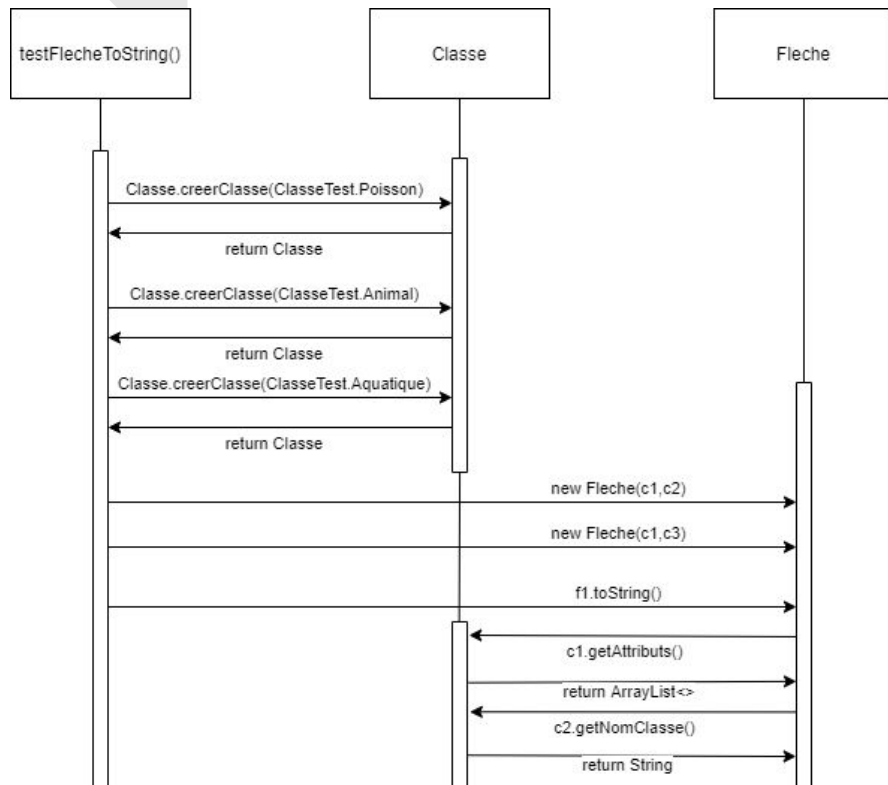


2.4 Itération 4

Fonctionnalités faites :

- Chargement d'une classe en dehors des dossier du projet (src/java)
- Copier/Coller une classe java dans le package ClasseChargees
- Création des flèches entre deux classes (Terminal toString())
- Ajout du menu Clic droit pour masquer ou afficher certaine lignes

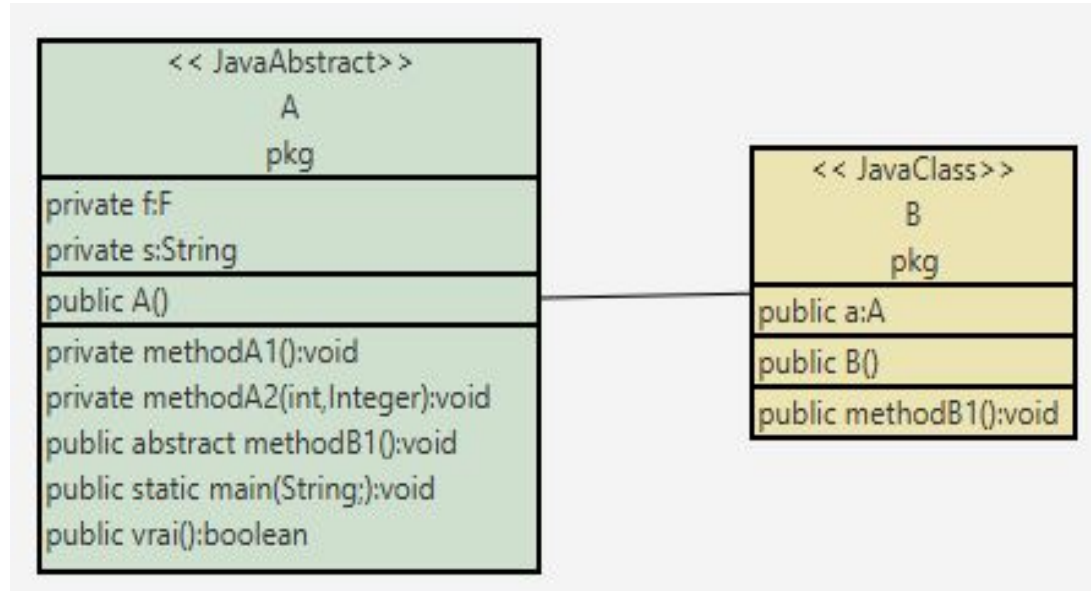
Fonctionnalité ajout de flèche.





Fonctionnalité ajout de flèche.

Exemple :



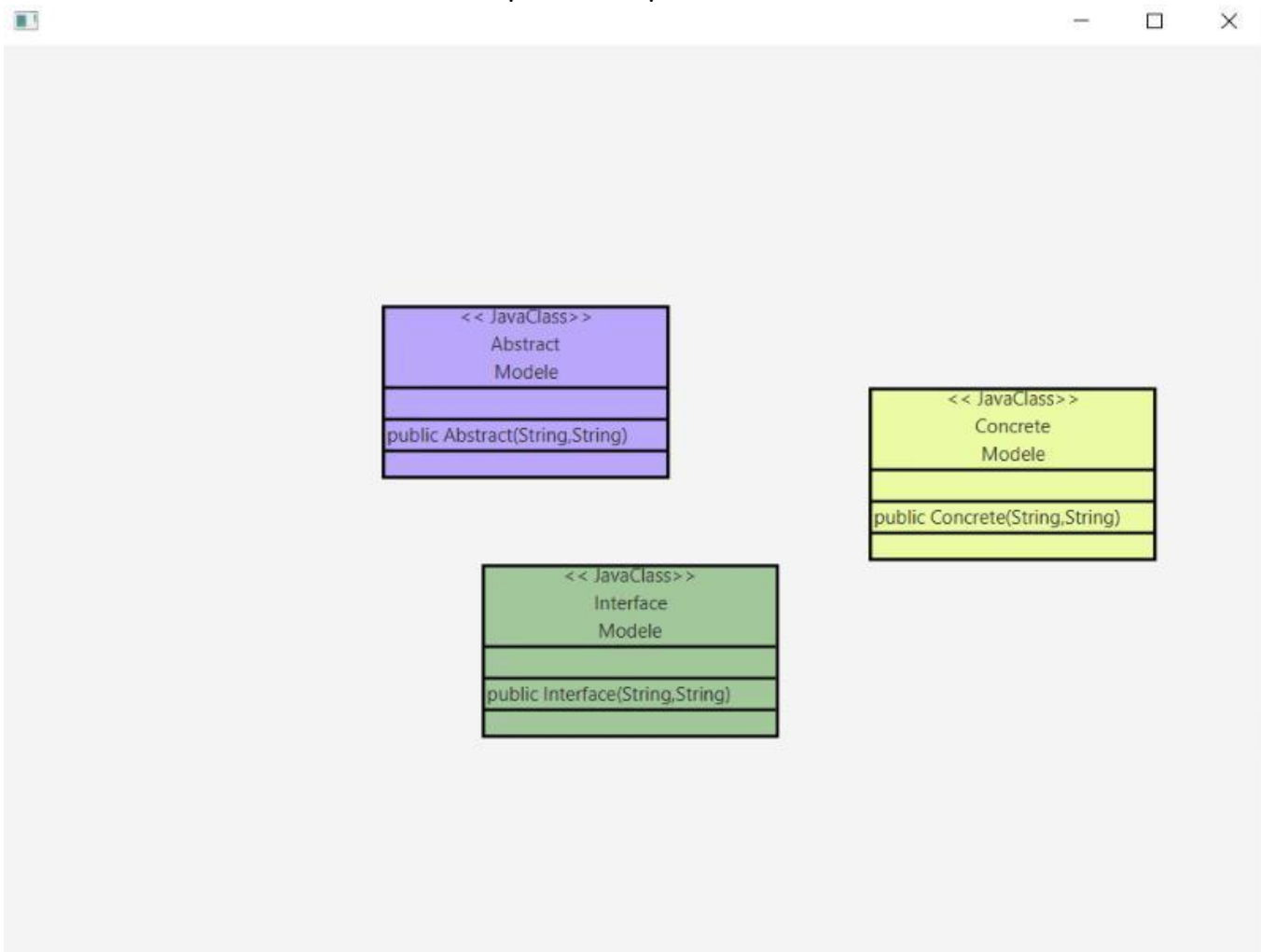


2.4 Itération 5

Fonctionnalités faites :

- Imprimer écran
- Compléter le menu clic droit

Exemple de capture d'écran





Fonctionnalité clic droit complet

```
<< JavaAbstract >>  
A  
pkg  
  
private f:F  
private s:String  
public A()  
  
private methodA1():void  
private methodA2(int,Integer):void  
public abstract methodB1():void  
public static main(String;):void  
public vrai():boolean
```

- ☐ Package
- ☐ Private
- ☐ Protected
- ☐ Public
- ☐ Static
- ☐ All
- ☐ None

```
<< JavaAbstract >>  
A  
  
  
  

```

- ☒ Package
- ☒ Private
- ☒ Protected
- ☒ Public
- ☒ Static
- ☒ All
- ☐ None

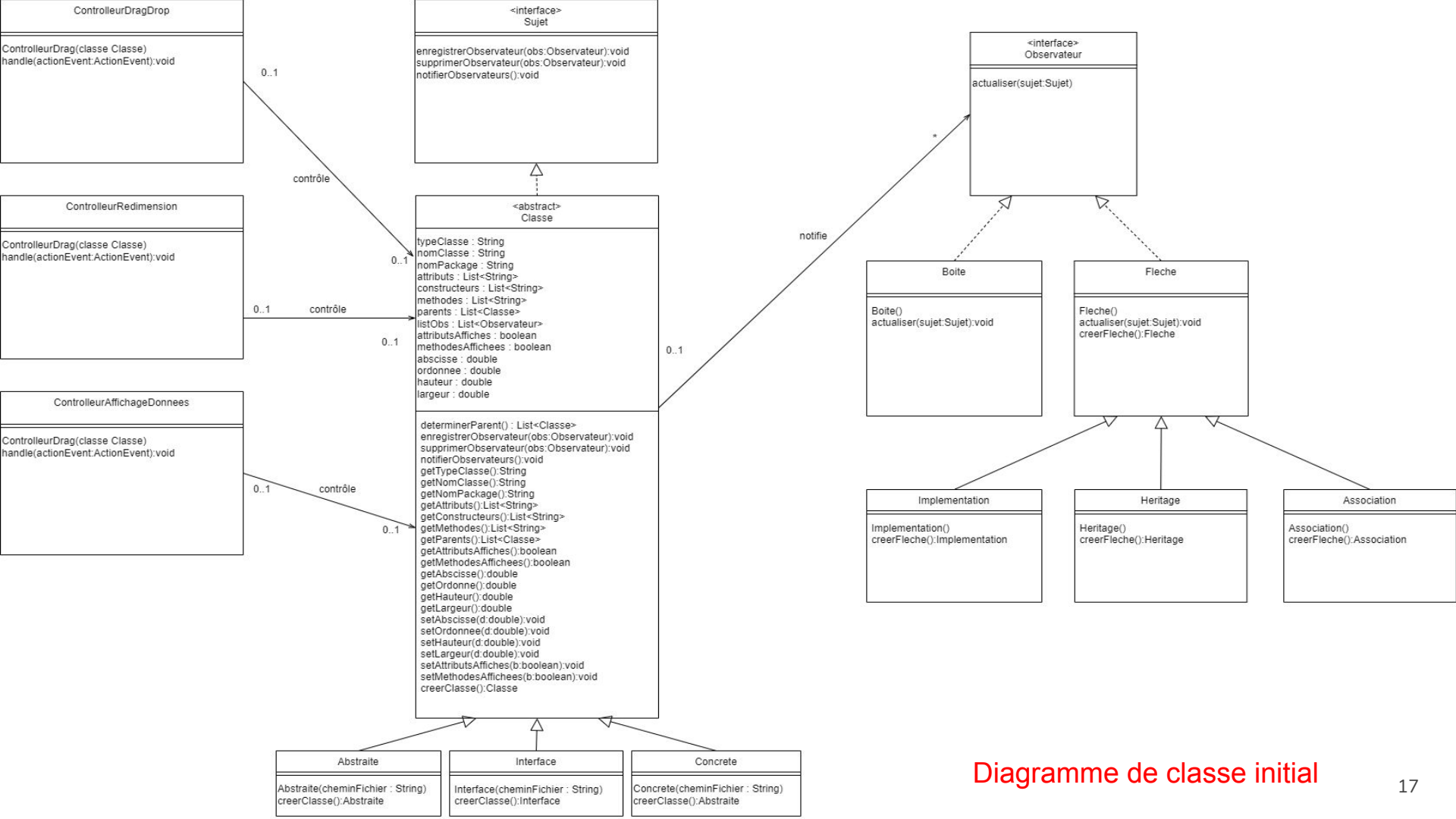


Diagramme de classe initial



3. Conclusion

Fonctionnalités principales réalisées :

- Afficher dans un terminal le contenu d'un objet Classe à partir d'un nom de package et de fichier
- Créer un objet Classe à partir d'un nom chemin de fichier au format .class ou d'un nom binaire (nomPackage.nomClasse) et déterminer ses parents (héritage, interfaces)
- Afficher dans une interface graphique les classes ajoutées en glissant-déposant et les associations de dépendances sous forme de flèches
- Afficher ou cacher à la demande les différentes informations (attributs, constructeurs, interfaces) constituant une classe à partir d'un menu clic droit
- Effectuer une capture d'écran de l'interface graphique en appuyant sur le bouton +