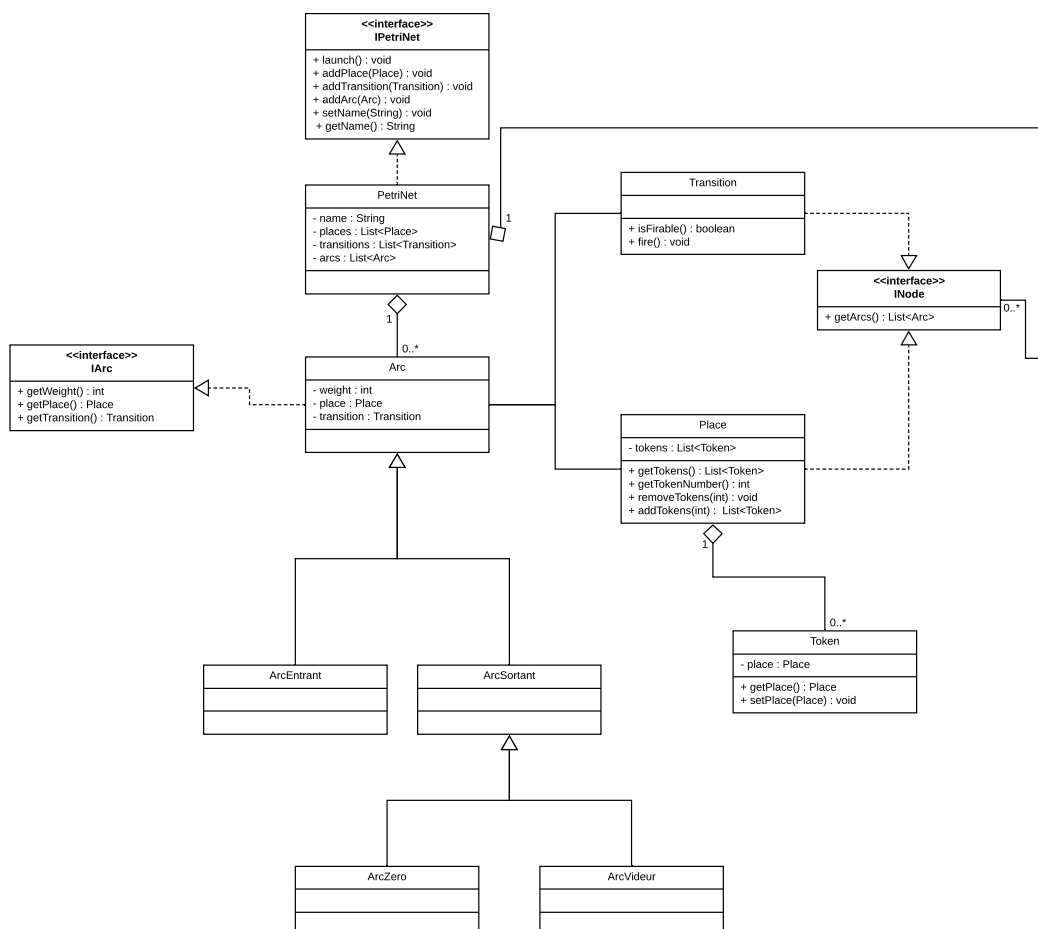




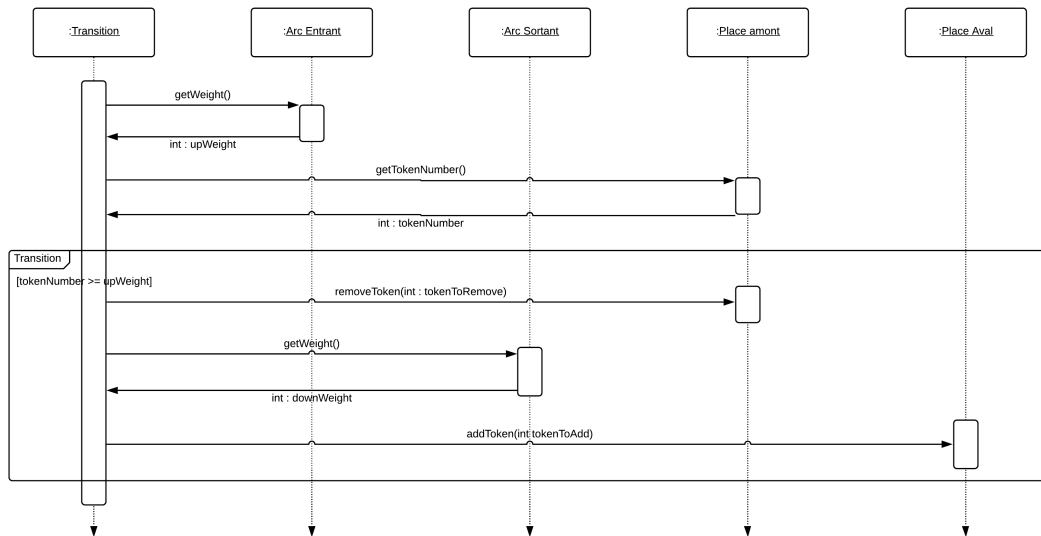
## FIL ROUGE- LIVRABLE 1

Ladislav DELLINGER, Alfred PICHARD

### 1 Diagramme de classes détaillé



## 2 Diagramme de séquence de la fonction `fire()`



## 3 Commentaires sur les choix faits

Nous avons identifié les principaux composants d'un réseau de Pétri, afin d'en structurer l'implémentation d'une simulation en Java. Ainsi, on sépare en différentes classes les composants Place, Transition et Arc.

Cependant, on peut remarquer des similitudes entre les Places et les Transitions, d'où l'idée de créer une interface `INodes` qui les regroupe.

De même, les différences entre les types d'Arcs et leur caractère entrant, sortant, ... nous a poussé à créer des sous classes pour mieux les séparer. Enfin, nous avons pris la décision d'écrire une classe `Token` dans une optique d'utilisation future possible ce réseau de Pétri. Ainsi, les tokens ne sont plus considérés comme des attributs des Places, et pourront représenter quelque chose de concret à l'avenir. En ce qui concerne l'interface `IPetriNet`, elle assure la modularité de notre simulateur.

Le diagramme de séquence représente le déroulement d'une itération de la fonction `fire()` de la classe `Transition`. Tout d'abord, on vérifie que l'opération est possible en récupérant la valeur du poids de l'arc ascendant, et en le comparant au nombre de token dans la Place du dessus. Ensuite, si les conditions sont vérifiées on effectue le transfert des jetons vers la Transition, puis de la Transition à la Place suivante, en adéquation avec le poids de l'Arc correspondant.