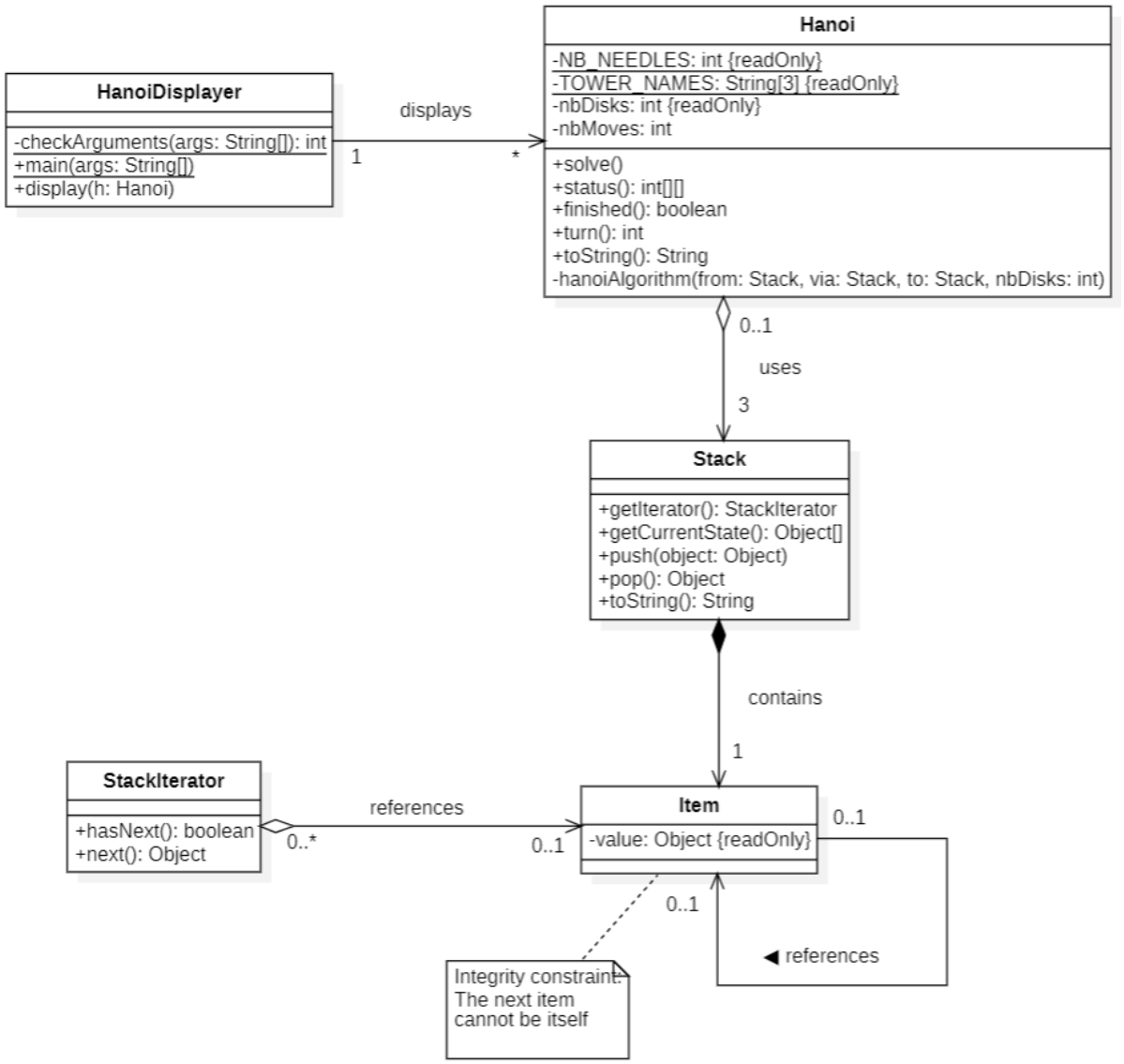
HEIG VD

Programmation orientée objet
Classe C – groupe F

Rapport du laboratoire n°7 : Tours de Hanoi

Auteurs : Kevin Farine, Timothée Van Hove

1 Diagramme de classe UML



2 Description des classes

2.1 Stack

Pour modéliser une pile, nous avons choisi de ne pas nous inspirer de l'implémentation de la stack Java, car elle hérite de la classe Vector. À la place, nous avons décidé de nous inspirer d'un laboratoire d'ASD du semestre passé en chainant les items de la pile qui ont chacun une référence sur l'item suivant. La classe Stack est composée de la classe interne privée Item et de la classe interne publique StackIterator. Notre pile redéfinit aussi la méthode toString(), nous permettant de renvoyer la représentation de chaque aiguille, comme demandé dans la donnée du labo.

2.2 Item

Cette classe permet de stocker la valeur d'un objet, et la référence sur l'objet suivant. Nous avons choisi de l'implémenter à l'intérieur de la classe Stack, car cela nous permet de représenter la composition. En effet, la destruction d'une stack induirait la destruction de l'item contenu à l'intérieur.

2.3 StackIterator

Cette classe permet d'itérer sur une pile avec les méthodes next() et hasNext(). Étant donné que la classe Item est une classe interne privée, nous étions obligés de rendre StackIterator aussi interne, mais avec une visibilité publique pour que les utilisateurs puissent l'utiliser. L'utilisateur ne pouvant pas instancier d'item, le constructeur de StackIterator est donc privé.

2.4 Hanoi

Nous avons décidé de contenir les 3 aiguilles (représentées par 3 Stack) dans un tableau, car nous devions implémenter la méthode status() qui renvoie un tableau 2d de int. Il est alors beaucoup plus aisé de factoriser le code dans une double boucle for imbriquée. Pour la représentation du problème à la console, nous avons redéfini la méthode toString(), puis simplement affiché la représentation graphique de chacune des aiguilles avec un formatage pour obtenir le même résultat que dans la donnée du labo.

2.5 Algorithme utilisé

Nous avons choisi la version récursive de l'algorithme que nous avions appris au cours d'ASD. Nous avons choisi cette version de l'algorithme, car elle est extrêmement simple à implémenter. Voici son fonctionnement :

1. Appel récursif pour déplacer les n-1 disques de la tour 1 vers la tour 2.
2. Déplacer le plus grand disque de la 1^{e} tour vers la tour 3.
3. Appel récursif pour déplacer les n-1 disques de la tour 2 vers la tour 3.

2.6 Réponse à la question

Pour résoudre le problème des tours de Hanoi, il faut effectuer $2^n - 1$ déplacements de disques. Pour déplacer 64 disques d'une tour à l'autre, il faut donc $2^{64} - 1$ mouvements.

A supposer qu'un mouvement de disque prend 1 seconde, il faut donc $2^{64} - 1$ secondes.

Dans une année, il y a :

$$60 * 60 * 24 * 365.25 = 31'557'600\ secondes$$

Il faut donc $\frac{2^{64}-1}{31'557'600} \approx 584'542'046'090$ années pour résoudre ce problème avec 64 disques. La légende précise que la résolution du problème prend place au commencement du monde, ce que nous interprétons comme la création de l'univers. Notre univers étant âgé de 13,8 milliards d'années, il reste donc environ 570,742 milliards d'années avant que l'univers ne disparaisse.

$$584'542'046'090 - 13'800'000'000 = 570'742'046'090\ années$$