

# Implantation d'un graphe

# 00 | Objectif

L'objectif de cette session de TP est de créer une structure de graphe d'information en appliquant les principes d'implantation issus du génie logiciel.

La configuration d'Eclipse doit être la même que pour les sessions précédentes.

**Ce sujet de travaux pratiques sera réalisé durant  
DEUX SEANCES  
de travaux pratiques.**

# 01 | Exercices

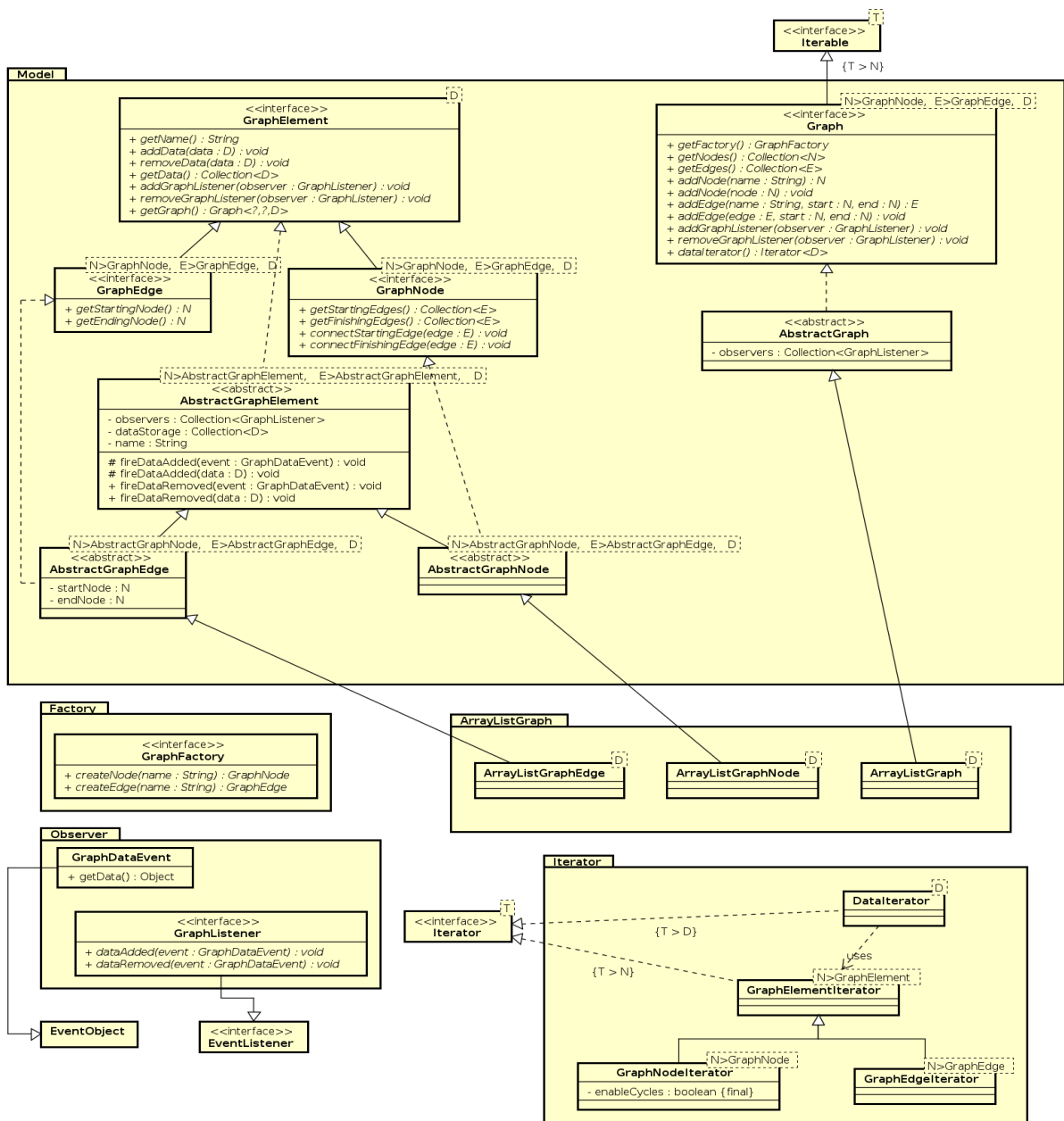
## 01.1 Réalisation d'une implantation d'un graphe

Un graphe est une structure de données constituée de nœuds (nodes) et d'arcs (edges). Les arcs relient les nœuds entre eux. Un arc ne peut relier que deux nœuds (ou un seul nœud si l'arc les deux extrémités de l'arc sont connectées à ce nœud).

Dans le cadre de ce TP, nous ne considérerons que le graphe est dirigé, c'est-à-dire que ses arcs ont un sens (du nœud de départ, jusqu'au nœud destination) et ne peuvent être traversés que dans ce sens.

Un graphe est une structure pouvant être utilisée dans le stockage d'information particulier. Toutefois, dans le cadre de ce TP, nous vous demandons de permettre le stockage de données d'un type prédéfini (T) dans les nœuds ou les arcs.

Le diagramme de classes UML ci-dessous vous donne la structure et les relations entre les types à implémenter à la fin du TP.



powered by Astah

Vous devrez respecter les bases des principes de la programmation inspirées des bonnes pratiques du génie logiciel, à savoir :

1. Utiliser des noms clairs et en anglais, car votre code doit pouvoir être lu par tout développeur, quelque soit son langage maternel.
2. Corriger le code pour qu'il n'y ait plus d'erreur ni de warning car chaque erreur ou warning indique une anomalie en relation avec les bonnes pratiques de programmation.
3. Ecrire une documentation claire et en anglais (vous aurez des warnings pour vous aider).
4. Appliquer le plus possible les design patterns :

[https://fr.wikipedia.org/wiki/Patron\\_de\\_conception](https://fr.wikipedia.org/wiki/Patron_de_conception)

**Travail à faire:**

### A) Structure d'arbre générique

- Écrire les interfaces `Graph`, `GraphNode` et `GraphEdge` représentant un graphe et les composants d'un graph. Ces types doivent prendre 3 paramètres génériques :
  - (a) `N` : représente le type des nœuds
  - (b) `E` : représente le type des arcs
  - (c) `D` : représente le type des données stockées dans les nœuds et/ou les arcs
- Écrire les classes abstraites correspondantes à chacune des trois interfaces implantées ci-dessus.
- Écrire une implantation concrète pour chacune des trois classes abstraites en utilisant des tableaux Java.

### B) Design Pattern : Factory

- Le patron de conception « Factory » permet d'externaliser la création d'objets (opérateur *new*) dans une classe particulière appelée *Factory*.
- Définissez l'interface `GraphFactory`.
- Créez une implantation de cette `GraphFactory` dans `ArrayListGraph` et utilisez là pour créer les nœuds et les arcs.

### C) Design Pattern : Observateur

- Écrire l'interface `GraphDataEvent` comme un objet contenant des informations correspondant à une modification des données dans le graphe.
- Écrire l'interface `GraphListener` représentant un observateur sur un graphe.
- Intégrer le patron de conception « Observateur » dans la structure de graphe. Les observateurs peuvent observer un nœud, un arc ou le graphe dans sa totalité.

### E) Design Pattern : Itérateur

- Deux implantation du patron de conception « Itérateur » sont nécessaire : une pour itérer sur les nœuds du graphe, et une seconde pour itérer sur les données stockées dans le graphe.
- Attention : le graphe peut être cyclique, c'est-à-dire qu'il contient des cycles dans les chemins pouvant être parcourus de nœud en nœud. Il faut donc éviter les boucles infinies lors des itérations.
- Créer l'itérateur sur les nœuds d'un graphe.
- Créer l'itérateur sur les données stockées dans un graphe.

### G) Programme de test

- Ecrire une fonction « main » réalisant les actions suivantes :
  - (a) création d'un graphe de nombres entiers
  - (b) ajout de nombres entiers aléatoires dans les nœuds et les arcs.
  - (c) Affichage des nœuds du graphe

(d) Affichage des entiers stockés dans le graphe.

**A LA FIN DES DEUX SEANCES DE TP, ENVOYEZ UN  
FICHIER ZIP CONTENANT LE CODE JAVA QUE VOUS  
AVEZ PRODUIT DURANT LE TP EN UTILISANT LA PLATE-  
FORME MOODLE.UTBM.FR**