

Projet de pôle Algorithme II, Programmation Java, Méthodologie d'analyse, Théorie des graphes

EISTI, ING1

2010

Dans ce projet nous allons nous intéresser aux algorithmes basés sur des structures de graphes permettant d'implémenter certaines fonctionnalités dans le cadre d'une entreprise de transport logistique par voie ferrée.

Les deux fonctionnalités principales à implémenter sont :

1. La détermination du nombre de wagons nécessaires pour stocker des produits en fonction de leur incompatibilité. Vous devrez en déduire la composition de chacun des wagons.
2. Le calcul du chemin le plus court pour desservir le client destinataire des produits. Vous devrez en déduire l'itinéraire du convoi.

Au niveau du projet, il vous sera demandé de réaliser deux applications :

1. une application pour le client permettant la saisie du destinataire (ville de livraison du client) et des produits ainsi que leur incompatibilités. Les données seront exportées dans un fichier binaire contenant les objets Java correspondant à la saisie.
2. une application pour l'entreprise qui permettra de charger les données d'un fichier client et devra ensuite afficher la feuille de route des produits saisis, incluant le nombre et la composition des wagons nécessaires et l'itinéraire du convoi. La carte ferrovière sera lue à partir d'un fichier texte avec le format suivant (exemple) :

```
villes : 3
1 ville1
2 ville2
3 ville3
liaisons : 2
1 3 120
2 3 180
```

- les villes sont numérotées de 1 à $nbVilles$
- les liaisons sont des triplets
(numéro ville de départ, numéro ville d'arrivée, temps du trajet en minutes).
- les liaisons sont bidirectionnelles et toujours sous la forme $(i, j, temps)$ avec $i, j \in [1..nbVilles]$, $i < j$ et $temps \in \mathbb{N}$.

Les livrables seront les suivants :

11/03/2010 : Analyse statique UML et codage du problème avec la structure de graphe.

Vous rendrez un document PDF contenant :

1. les types abstraits permettant de modéliser :
 - (a) les produits et leurs incompatibilités
 - (b) la carte ferrovière

Pour ces deux types abstraits, inspirez vous du type graphe vu en cours d'algorithmique II. Exprimez les opérations de base sous forme axiomatiques et donnez le prototype des opérations d'extension nécessaire à votre application.

2. diagramme de classes UML des deux applications
3. contraintes OCL de votre diagramme de classes

06/05/2010 : Analyse dynamique UML et algorithmes sur les graphes.

Vous rendrez un document PDF contenant :

1. les algorithmes des opérations d'extensions définies dans le livrable 1.
2. cas d'utilisation des applications
3. diagramme de séquence du calcul du nombre de wagons et de leurs compositions
4. diagramme de séquence du calcul d'itinéraire
5. diagrammes d'états-transitions des classes s'y prêtant

03/06/2010 : Implémentation en Java.

Vous rendrez une archive de votre projet Java contenant de manière structurée :

1. le code source Java
2. les ressources : le plan du réseau ferrovière et un exemple de saisie client
3. un fichier de configuration Ant permettant au moins de :

- (a) compiler (règle compile)
- (b) fabriquer la documentation (règle doc)
- (c) générer les jars exécutables correspondant aux deux applications (règle jar) : transport-client.jar et transport-entreprise.jar
- (d) nettoyer (règles clean.bytecode, clean.doc, clean.jars, clean.all)
- (e) d'obtenir de l'aide sur l'utilisation du projet (règle help)