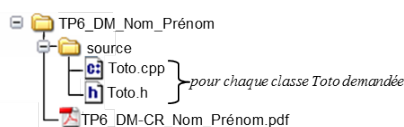


TP de C++ n° 6

Infrastructures de transport

Ce sujet comporte trois morceaux et forme le TP 6, qui se prolonge en un travail à la maison par groupe de deux. Le principe est que chaque membre du groupe prenne en charge un morceau (morceau A : aéroport ou morceau B : transport) : le travail à la maison (morceau C) consiste à faire fonctionner les deux morceaux ensemble sur des cas d'étude. Vous devez rendre tout ensemble sous la forme d'une archive, qui, une fois extraite, doit révéler l'arborescence suivante :



où aucun fichier ne doit comporter de fonction `main` et où le fichier `.pdf` doit être un compte-rendu présentant le travail réalisé et incluant un mode d'emploi et des tests, au format PDF. Cette archive doit être déposée au plus tard le vendredi 27 octobre 2017 à 20h00 sur l'Espace Numérique de Travail dédié au cours. Tout travail déposé en retard, contenant du code source ne compilant pas, ou non conforme à la consigne donnée ci-dessus (arborescence, noms de fichiers, format PDF, présence de `main`, ...) ne sera pas corrigé et recevra un zéro.

— : —

Le TP est volontairement peu guidé pour vous laisser mettre en œuvre une démarche complète. Le compte-rendu devra comporter une description de la conception du programme, des tests permettant de garantir le bon fonctionnement du programme, et des explications sur l'intégration des composants pour les cas d'étude.

Toutes les classes demandées doivent être munies de constructeurs, destructeurs, accesseurs, etc. Il faut employer `const` chaque fois que nécessaire.

A – Aéroport

On veut pouvoir représenter plusieurs types de `Terminal` :

- la Gare permet de faire circuler des trains ;
- l'`AéroportRegional` est un petit aéroport pouvant avoir des liaisons régulières avec un seul autre aéroport ;
- l'`AéroportInternational` est un moyen aéroport pouvant avoir des liaisons régulières avec 4 autres aéroports ;
- le `HubAéroport` est un hub, i.e. un grand aéroport pouvant avoir des liaisons régulières avec 12 autres aéroports ;
- le `HubMultimodal` est un hub muni en plus d'une gare.

Coder les classes nécessaires en implantant une forme d'héritage.

Un `Terminal` a une latitude et une longitude (en degrés), et des liaisons avec d'autres, i.e. leur est relié par une ligne régulière. Il a également un temps moyen de correspondance, i.e. le temps que doivent en moyenne attendre les passagers venant d'une liaison donnée et repartant par une autre.

À chaque terminal, on veut également pouvoir rattacher un flux de passagers, i.e. un nombre de passagers venant chaque jour et ayant une destination finale précise. La destination finale n'est peut-être pas accessible depuis le terminal (car on admet que les passagers puissent avoir à faire des correspondances).

On munira la classe `Terminal` d'une méthode `distance(double lat, double lng)` qui calcule la distance entre le terminal et le point de coordonnées données en argument.

B – Ligne

Une `Ligne` de transport est une classe template prenant en argument un `Moyen` de transport. Les `Moyens` peuvent être l'avion, l'avion électrique, le train. Chacun a une vitesse moyenne, une empreinte carbone au km, et une capacité (nombre de passagers transportés)

	capacité	vitesse	empreinte
train	1000 passagers	250 km/h	1,5 kg/km
avion électrique	100 passagers	400 km/h	6 kg/km
avion	150 passagers	700 km/h	18 kg/km

Une ligne de transport relie un `Terminal` (origine) à un autre (destination), et a une fréquence, i.e. un nombre de passage par jours.

Une fois cette classe écrite, écrire une classe `Voyage`. Un voyage a également une origine et une destination, mais a en plus une liste de lignes à emprunter (en faisant des correspondances). Faire en sorte que cette classe permette, à partir d'un flux de passagers, de calculer le temps de trajet et l'empreinte carbone correspondant à ce flux.

C – Études de cas

On s'intéresse au transport entre les villes suivantes : Paris, Bruxelles, Lyon, Rome, Naples. Chercher les coordonnées (latitude et longitude) de ces villes.

On veut étudier six scénarii :

- Scenario 1 : Paris est un hub, Rome est un aéroport international relié à Paris, les autres des aéroports régionaux (Lyon et Bruxelles reliés à Paris, Naples relié à Rome).
- Scenario 2 : Paris est un hub multimodal, Rome un aéroport international relié à Paris, Bruxelles une gare, Lyon et Naples des aéroports régionaux (mêmes liaisons).
- Scenario 3 : Paris et Rome sont des hubs multimodaux, les autres des gares (Lyon reliée à Paris et Rome, Bruxelles reliée à Paris, Naples reliée à Rome).
- Scenario 4 : Toutes les villes sont des aéroports internationaux reliés à tous les autres.
- Scenario 5 : Toutes les villes sont des gares reliées dans cet ordre : Bruxelles – Paris – Lyon – Rome – Naples.
- Scenario 6 : Paris est un hub, les autres des aéroports régionaux reliés à Paris.

Les flux de passagers sont les suivants (nombre de passagers par jour voulant aller d'un endroit à l'autre) :

	Bruxelles	Paris	Lyon	Rome	Naples
Bruxelles	—	13000	5500	4500	2000
Paris	12500	—	18000	10000	4000
Lyon	6000	19000	—	6500	2500
Rome	5000	9000	6000	—	11000
Naples	2500	5000	3000	10000	—

On souhaite fixer un temps de trajet maximum, et choisir le scénario minimisant l'empreinte carbone globale.

Faire cette étude pour les valeurs suivantes du temps de trajet minimum : 4h, 5h, 6h, 7h, 8h.