UNIVERSITÉ LIBRE DE BRUXELLES ULB

FACULTÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE

INFO-F105

Deuxième projet (C++)

Année académique 2019–2020

Résumé

Ceci est l'énoncé du quatrième projet du cours « Langages de programmation 1 ». On vous demande d'écrire un programme qui nous permet de modéliser une épidémie.

Introduction

Dans le modèle chaque agent peut être infecté par un pathogène et transmettre une infection, ainsi que récupérer. De plus, les agents seront stratifiés en groupes sociaux.

Modalités de réalisation

Pour réaliser le programme, trois classes sont essentielles. Si vous ressentez le besoin de créer des classes ou des méthodes supplémentaires, n'hésitez pas, mais adhérez pleinement à celles décrites.

Une classe *Personne*, pour représenter une personne.

Une classe *Pathogene*, qui représente un agent pathogène qui peut infecter quelqu'un.

Une classe *Groupe*, qui représentera un groupe, i.e. un ensemble de personnes.

La classe *Personne* contiendra les attributs suivants :

- Prénom
- Nom de famille
- âge
- pathogène infectant (en utilisant un pointeur sur un objet pathogène instancié)
- groupes sociaux associés (en utilisant un vecteur de pointeurs vers ces groupes)

Une personne aura également les méthodes suivantes ;

- *infection(pathogene)* : représentant l'infection par un pathogène spécifique.
- *infecter(personne)* : qui transmet l'agent pathogène infectieux à une autre personne.
- *recuperer()* : qui guérit une personne (i.e. plus d'agent pathogène associé).
- *afficher()* : qui affiche des informations sur la personne et son pathogène associé,le cas échéant.

Un pathogène a un type et un nom, un type peut être une bactérie ou un virus par exemple. De plus, chaque instance d'un pathogène aura un vecteur avec des pointeurs vers toutes les personnes qu'il infecte, et une méthode pour imprimer toutes les personnes infectées.

Un groupe social est un ensemble de personnes. Il sera implémenté à l'aide d'un vecteur de pointeurs vers les personnes de ce groupe. Une personne peut être présente dans plusieurs groupes sociaux. Un groupe social aura une méthode pour ajouter une personne, la supprimer et afficher toutes les personnes qu'il contient.

Les implémentations doivent être placées dans des fichiers cpp, les déclarations dans des fichiers hpp.

À titre complémentaire, on vous demande d'écrire un *main()* qui démontre votre implantation. Celui-ci initialise un certain nombre de groupes sociaux, de personnes et et de pathogènes, et démontre la fonctionnalité de votre code (vous pouvez choisir des valeurs où nécessaire). Ce fichier se nommera *main.cpp*.

Essayez de penser à tous les cas limites et effets de bord possibles et couvrez-les dans votre implantation.

Le projet doit être écrit en C++ strictement standard. Vous devez ainsi l'avoir compilé en respectant les directives indiquées sur l'UV (version du compilateur, options, etc.), sans susciter le moindre avertissement (*warning*).

Compilation

Votre code doit être accompagnése d'un makefile qui compile le projet.

ATTENTION !

Tous les *warnings* non résolus qui n'ont pas de justification explicite (par un commentaire dans le code, par exemple) seront sanctionnés d'un point sur la note finale. Et un code qui ne compile pas ne sera pas corrigé.

Indications complémentaires

Vous devez soumettre un fichier compressé `.zip` (donc pas de .rar ou autre) qui contient **un seul dossier**. Ce dossier doit s'appeler `<nom>_<prenom>` (pas de majuscules ni de caractères accentués), le fichier zip doit porter le même nom. Par exemple, Alan Turing doit soumettre un fichier `turing_alan.zip` qui contient le dossier `turing_alan` avec son projet.

Consignes pour la remise du projet

À respecter scrupuleusement !

1. Chaque fichier doit commencer par un commentaire indiquant **votre nom** et **votre section**.
2. Le fichier est à remettre exclusivement via l'UV.
3. Un fichier unique `.zip` doit être remis.
4. Votre code doit être **commenté**.
5. Date limite : le **vendredi 15 mai 2020** avant **8 heures** ;
   Après ce moment, les projets sont considérés comme **en retard** et ne seront plus acceptés.
6. Questions : nversbra@ulb.ac.be