

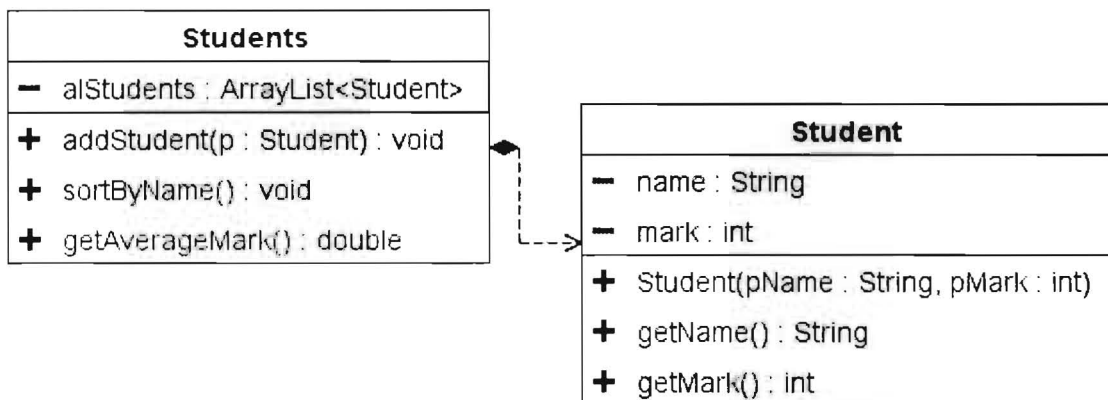
Code branche INFOR	Ministère de l'Education nationale, de l'Enfance et de la Jeunesse EXAMEN DE FIN D'ETUDES SECONDAIRES TECHNIQUES Régime technique – Session 2015/2016	
Épreuve écrite	Branche	Division / Section
Durée épreuve 3h	Informatique	GE
Date épreuve 3.6.2016		

Dans votre répertoire de travail (à définir par chaque Lycée), vous trouverez un sous-dossier nommé **EXAMEN_GE**. Renommez ce dossier en remplaçant le nom par votre code de l'examen (Exemple de notation : **LTXY_GE2_07**). Tous vos fichiers devront être sauvegardés à l'intérieur de ce dossier, qui par la suite sera appelé 'votre dossier'!

Question 1

15 points

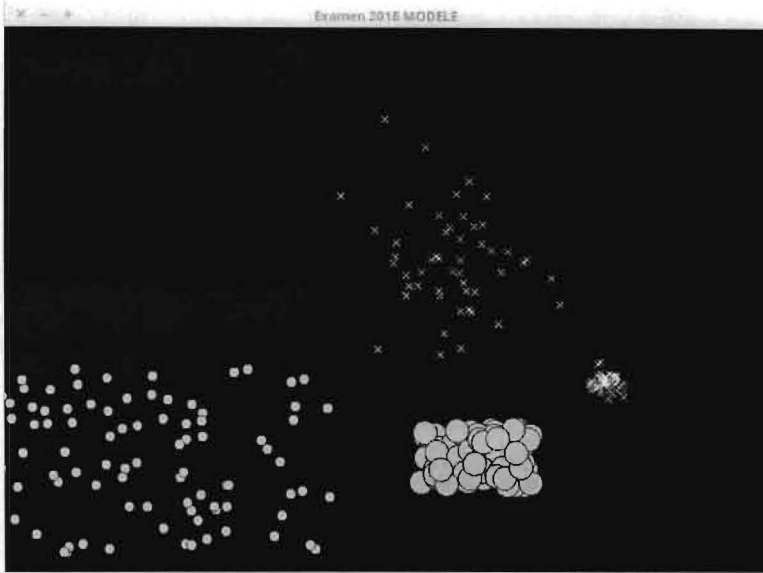
Dans **Unimozzer** ou **NetBeans**, ouvrez le projet « Question1 » qui contient déjà les deux classes `Student` et `Students` :



- Écrivez le code source de la méthode `sortByName` qui trie la liste `alStudents` à l'aide de l'algorithme de tri par sélection directe suivant le nom des étudiants dans l'ordre alphabétique croissant. (9 points)
- Écrivez le code de la méthode `getAverageMark` qui retourne la moyenne non arrondie de toutes les notes de liste `alStudents`. Si la liste est vide, cette méthode retourne la valeur `-1`. (6 points)



Dans la suite vous allez écrire l'application dénommée « FireWorks ». Il s'agit d'un petit programme capable de créer des animations simples simulant un feu d'artifice.



Vous trouvez une version exécutable du programme, nommé **Modele.jar**, dans le dossier « Question2 » de votre dossier. Avant de continuer, il est recommandé de lancer et de tester ce programme en cliquant le bouton gauche ainsi que le bouton droit à différents endroits de la surface noire !

Créez dans **NetBeans** un nouveau projet nommé **FireWorks** que vous sauvegardez dans votre dossier.

Réalisez cette application en vous basant sur la version exécutable fournie ainsi que sur le diagramme UML de la dernière page et tout en respectant les instructions et précisions données dans la suite.

La classe Shape (5 points)

- Les attributs `x` et `y` représentent les coordonnées du centre d'un shape.
- L'attribut `ttl` (*time to live*) représente le temps de vie restant d'un shape et est initialisé à 255.
- Implémentez le constructeur, les manipulateurs (*setters*) et accesseurs (*getters*) spécifiés dans le schéma UML.
- La méthode `draw` ne fait rien et reste donc vide.
- La méthode `move` décrémente l'attribut `ttl` d'une unité.

La classe Particle (10 points)

- Cette classe hérite de la classe `Shape` et sert à représenter une particule en mouvement.
- Les attributs `vx` et `vy` représentent la vitesse horizontale resp. verticale de la particule. `vx` est initialisé à une valeur décimale de l'intervalle $[-2 ; 2[$, `vy` à une valeur décimale de l'intervalle $[-4 ; -2[$. Le rayon est initialisé à 15.
- La méthode `draw` dessine une particule, c'est-à-dire un disque gris avec un bord noir, centré autour de ses coordonnées (x,y) .
- Dans la méthode `move`, `ttl` est décrémente d'une unité, `x` est incrémenté de `vx`, `y` est incrémenté de `vy`, puis `vy` est incrémenté de 0.10. Si la valeur `ttl` est un multiple de 10, alors le rayon est décrémente d'une unité sous condition de rester strictement positif.

La classe Cross (6 points)

- Cette classe hérite de la classe `Shape` et sert à représenter une croix (ayant la forme du signe X) en mouvement.
- La méthode `draw` dessine une croix (X) de 7x7 pixels, centrée autour (x,y). Sa couleur est blanche.
- Dans la méthode `move`, `ttl` est décrémenté d'une unité, `x` et `y` sont incrémentés de valeurs décimales aléatoires de l'intervalle `[-5 ; 5[`.

La classe Shapes (7 points)

- Cette classe représente les différentes formes et possède à cet effet une liste `alShapes`, à initialiser.
- La méthode `draw` sert à dessiner toutes les formes (ici : particules et croix) de la liste `alShapes`.
- La méthode `add` sert à ajouter un `shape` à la liste `alShapes`.
- La méthode `move` sert à déplacer toutes les formes contenues dans la liste `alShapes`.
- La méthode `cleanUp()` sert à enlever toutes les formes de la liste dont la valeur `ttl` est strictement négative.

La classe DrawPanel (4 points)

- Cette classe possède un attribut `shapes` et un manipulateur `setShapes`.
- La méthode `paintComponent` dessine d'abord l'arrière-plan noir, puis les différentes formes.

La classe MainFrame (13 points)

- Complétez le constructeur `MainFrame()` par les instructions pour initialiser les attributs `shapes` et `timer`. Le bouton `stepButton` est rendu invisible et le `timer` est démarré.
- Lorsque l'utilisateur clique avec le bouton gauche de la souris sur l'arrière-plan noir, alors 100 nouvelles particules sont créées à cet endroit.
- Lorsque l'utilisateur clique avec le bouton droit de la souris sur l'arrière-plan noir, alors 50 nouvelles croix sont créées à cet endroit.
- Toutes les 30 millisecondes, le chronomètre (via la méthode du `stepButton`) déplace les formes, supprime celles qui sont en fin de vie et rafraichit l'écran.

Schéma UML

