

## Exercice 1.

1. Définir une classe *Etudiant* avec les attributs nom, prénom et note. Définir les getters et setters des attributs de la classe.
2. Définir une classe gérant une l'exception *NoteEtudiantIncorrecte* qui devra être levée quand la note de l'étudiant fournie au constructeur ou à *setNote* de *Etudiant* n'est pas comprise entre 0 et 20.
3. Faites une classe *TestNoteEtudiant* qui va générer puis gérer ladite exception.

## Exercice 2.

Une société recrute de nouveaux membres d'équipage. L'âge minimum pour être membre d'équipage est de 18 ans et l'âge maximum est de 22 ans. Une exception définie par l'utilisateur est lancée si l'âge est inférieur à 18 ans ou supérieur à 22 ans.

Créer une classe gérant cette exception ainsi qu'une classe de test.

## Exercice 3.

Réaliser une classe permettant de manipuler des entiers naturels (positifs ou nuls) et disposant :

- d'un constructeur à un argument de type `int` ; il génèrera une exception *ErrConst* si la valeur de son argument est négative ;
- de méthodes statiques de somme, de différence et de produit de deux naturels ; elles génèreront respectivement des exceptions *ErrSom*, *ErrDiff* et *ErrProd* lorsque le résultat ne sera pas représentable ; la limite des valeurs des naturels sera fixée à la plus grande valeur du type `int` ;
- une méthode d'accès *getN* fournissant sous forme d'un `int` la valeur de l'entier naturel.

On s'arrangera pour que toutes les classes exception dérivent d'une classe *ErrNat* et pour qu'elles permettent à un éventuel gestionnaire de récupérer les valeurs ayant provoqué l'exception.

Écrire deux exemples d'utilisation de la classe :

- l'un se contentant d'intercepter sans discernement les exceptions de type dérivé de *ErrNat*,
- l'autre qui explicite la nature de l'exception en affichant les informations disponibles.

Les deux exemples pourront figurer dans deux blocs *try* d'un même programme.