

## 5. (10 points)

**Rappels :**  $\triangleright$  `parseInt(String s)` est une méthode statique de la classe `Integer` qui renvoie l'entier correspondant à la chaîne de caractères `s` si elle est composée de chiffres;

$\triangleright$  `NumberFormatException` est une exception lancée lorsqu'une conversion d'une chaîne de caractères vers un type numérique ne peut se faire car la chaîne n'a pas le format approprié.

**5.1** **Ecrire** une classe `NombreNegatif` héritant de la classe `Exception` dont le constructeur fait afficher à l'écran "ce n'est pas une longueur".

**5.2** **Ecrire** une classe `VerificationEntier` héritant de la classe `NumberFormatException` dont le constructeur fait afficher à l'écran "ce n'est pas un entier".

**5.3** On considère une classe `Rectangle` dont les attributs entier sont `cote1` et `cote2` avec ses constructeurs et ses accesseurs. En utilisant la classe `Scanner`, **écrire** pour cette classe une méthode statique `saisieCorrecte`

- qui demande à l'utilisateur de saisir une valeur positive entière au clavier (considérée comme une chaîne de caractères)
- qui, si la chaîne saisie ne correspond pas à un entier, affiche le message d'erreur "ce n'est pas un entier" et recommence la saisie
- qui, sinon, si la chaîne saisie ne correspond pas à un entier positif, affiche un message d'erreur "ce n'est pas une longueur" et recommence la saisie
- qui, sinon renvoie cette valeur.

**5.4** Toujours pour la classe `Rectangle`, **écrire** une méthode `calculSurface` qui renvoie la surface de l'objet `Rectangle`

**5.5** Toujours pour la classe `Rectangle`, **écrire** une méthode statique qui

- qui construit un objet `rect Rectangle`
- qui demande à l'utilisateur de saisir 2 fois un entier positif et affecte respectivement ces valeurs aux attributs `cote1` et `cote2` de l'objet `rect`
- qui renvoie la surface de l'objet `rect`