

TD n° 8

Exceptions

Exercice 1 On considère la classe :

<pre>public class Universite{ private int nbEtu; //nombre d'étudiants inscrits e l'universite private int nbEtuS; //nombre d'étudiants inscrits en science private int nbEtuI; //nombre d'étudiants inscrits en informatique private int nbEtuM; //nombre d'étudiants inscrits en mathématiques }</pre>	<p>1</p> <p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>8</p>
---	---

1. Écrire un constructeur public `Universite(int nbEtu, int nbEtuS, int nbEtuI, etuM)`. Ce constructeur lèvera une exception de type `IllegalArgumentException` (qui est une sous classe de `RuntimeException`) lorsqu'il le faudra (par exemple si `nbEtu` est négatif ou nul, si `nbEtuS` est strictement supérieur à `nbEtu`, ect...).
2. Écrire une méthode public `void descriptif()` qui affiche le nombre d'étudiants inscrits à l'université, le pourcentage d'étudiants scientifiques inscrits en informatique et le nombre d'étudiants inscrits en mathématiques. Si `NbEtuS` est nul, la méthode devra capter l'exception levée par le programme. Cette exception est de type `ArithmeticException`.

Dans les exercices suivant on introduit certaines exceptions. On demande à chaque fois de les écrire.

Exercice 2 On suppose créées des classes `Etudiant` et `Professeur` possédant des attributs `String nom` et `String prenom`. On souhaite faire une classe `AnnuaireUniversite` admettant deux arguments `Etudiant[] etu` et `Professeur[] prof`.

1. Écrire une méthode `estMembreUniversite(String nom, String prenom)` qui affichera le nom et le prénom si la personne figure dans `etu` ou dans `prof`. Si ce n'est pas le cas cette méthode lèvera une exception de type `MembreNonTrouveException`.
2. Sur le même principe, on rajoutera des méthodes `estEtudiant(String nom, String prenom)` et `estProfesseur(String nom, String prenom)` qui lèveront des exceptions du type `EtudiantNonTrouveException` et `ProfesseurNonTrouveException`. Ces exceptions héritent de `MembreNonTrouveException`.

Exercice 3 On va modéliser une saisie de menu. Un menu comprend une liste d'entrées, une liste de plats et une liste de desserts. On considère la classe :

```
public class Menu{ 1
    private String[] entrees; //entrées proposées dans le menu 3
    private String[] plats; //plats proposés dans le menu 4
    private String[] desserts; //desserts proposés dans le menu 5
} 7
```

1. Écrire un constructeur prenant en arguments trois tableaux de chaînes de caractères. Si l'un de ces tableaux est de longueur strictement inférieure à 2 ou strictement supérieure à 9, le constructeur lèvera une exception de type `NEstPasBon` (afin de simplifier la modélisation, on suppose qu'il y a au plus 9 entrées, 9 plats et 9 desserts).
2. On souhaite définir une méthode `void choixMenu()` qui affiche dans un premier temps la liste des entrées puis demande à l'utilisateur celle qu'il choisie, fait la même chose avec les plats puis les desserts. Si pour l'un des choix l'utilisateur entre autre chose qu'un chiffre, une erreur de type `MauvaiseSaisieException` est levée. Si l'utilisateur entre, (pour le choix de l'entrée, du plat ou du dessert) un chiffre qui n'est pas compris entre 1 et la longueur du tableau correspondant, une exception du type `MauvaisChoixException` est levée. Afin de simplifier la situation on suppose que l'utilisateur entre toujours un caractère.
N'hésitez pas à définir des méthodes intermédiaires afin, notamment, de gérer les différents types d'exceptions. On suppose en particulier créée une méthode `static charToInt(char a)` qui convertit `a` en `int` si `a` est égal à '1', '2', ..., '9' et lève une exception de type `MauvaiseSaisieException` sinon.

Exercice 4 Si vous avez du temps.

On reprend la classe `Menu`. On suppose créées deux méthodes `Menu menuDeMidi()` et `Menu menuDuSoir()`. La méthode `menuDeMidi()` (respectivement `menuDuSoir()`) renvoie le menu obtenu en ne conservant que les cases paires (respectivement impaires) de `this.entrees`, `this.plats` et `this.desserts`.

Définir des méthodes `void choixMenuDeMidi()` et `void choixMenuDuSoir()` qui fonctionnent comme `void choixMenu()` mais qui lèvent une exception de type `PasLeBonMenuException` si l'utilisateur choisie un plat du soir dans le menu de midi ou réciproquement. Cette exception hérite de `MauvaisChoixException`.