

TP 4 : Java en mode console - Exceptions - E/S

Ce TP est à effectuer avec votre éditeur de texte préféré sous linux + une console.

Des robots :

Un Robot est caractérisé par un nom, et un état (ON/OFF). On peut allumer et éteindre un Robot, et lui faire afficher son nom.

1. Écrivez la classe `Robot`, et compilez la.
2. Si ce n'est pas fait, ajoutez une méthode `main` dans la classe `Robot` afin de tester votre classe.
3. Mettez à présent votre classe dans un répertoire `robots`. Compilez à nouveau en vous plaçant au niveau du répertoire (i.e. `javac robots/Robot.java`). Que constatez vous ? Afin de résoudre ce problème, indiquez le nom du package (i.e. du répertoire) dans votre classe.
4. Exécutez votre programme en utilisant la commande `java`.

On ajoute des Piles :

Un robot est maintenant composé d'une liste de `NB` piles d'un type donné. Une pile est caractérisée par son type (`C`, `D`, `AA` ou `AAA`) et sa capacité (un entier). Vous définirez en particulier un type `BatteryType` permettant de représenter les différents types de piles possibles.

À sa création, un robot indiquera le type de piles qu'il utilise ainsi que le nombre nécessaire à son fonctionnement. Pour fonctionner, un robot doit donc posséder `NB` piles (du bon type) de capacité > 0 .

1. Ajoutez la classe `Battery` dans le package `robots`. En particulier, vous définirez un constructeur permettant de construire n'importe quelle piles (type et capacité), ainsi que la constante de classe `DEFAULT_CAPACITY = 10`
2. Dans votre classe `Robot`, ajoutez un constructeur prenant en paramètres son nom, un type de pile, un nombre de piles et la capacité de chaque pile. Ce constructeur crée un robot avec toutes ses piles.
3. Ajoutez également une méthode permettant de remplacer la pile située en position k , par une autre pile passée en paramètre.
4. Ajoutez enfin une méthode permettant de savoir si un robot peut fonctionner ou non, qui retourne la liste des positions des piles HS.
5. Modifiez les méthodes de la classe robot de manière à ce que les piles s'usent d'une unité à chaque fois que le robot affiche son nom.

Gestion d'Exceptions :

On souhaite gérer 3 types d'exceptions :

- `InvalidBatteryCapacityException` : si on tente de créer une pile de capacité négative ou nulle,
- `InvalidBatteryTypeException` : si l'on tente d'ajouter à un robot une pile d'un autre type que celui qu'il accepte.
- `InvalidNbBatteryException` : si l'on tente de créer un robot avec un nombre de piles négatif ou nul.

1. Définissez les trois classes correspondantes comme des sous-classes de la classe `Exception`.
2. Testez votre constructeur de `Robot(String, BatteryType, int, int)`. Il devra en outre éventuellement transmettre au main une exception de type `InvalidNbBatteryException`.
3. Modifiez le code de manière à pouvoir récupérer, dans le constructeur de `Robot`, une exception de type `InvalidBatteryCapacityException`, et le cas échéant, créer des piles de capacité `DEFAULT_CAPACITY`.
4. Modifiez la méthode permettant de changer une pile de manière à pouvoir récupérer une exception de type `InvalidBatteryTypeException`, et le cas échéant, afficher un message d'erreur (sans effectuer le changement de pile).

Création d'un jar :

On souhaite créer une bibliothèque permettant de gérer les robots.

1. Créez une archive `robots.jar` contenant l'ensemble des fichiers bytecode (i.e. `.class`), permettant de manipuler des robots à piles. Mettez cette archive jar dans un répertoire `tmp`.
2. Créez un nouveau package `testjar`. Dans ce package, définissez une classe `Main` et une méthode `main` dans laquelle vous déclarez et utilisez des robots. N'oubliez pas d'importer les fichiers nécessaires à la manipulation des robots et des piles.
3. Compilez et exécutez votre classe `Main` en utilisant l'archive `robots.jar`.

Interactions avec l'utilisateur :

On souhaite pouvoir interagir avec l'utilisateur pour la création d'un robot. Pour cela, vous utiliserez un objet de type `Scanner`, branché sur `System.in`.

1. Définissez la méthode `static Robot getRobot()` permettant de créer un robot à partir de données saisies par l'utilisateur (nom du robot, type de piles, nombre de piles, et capacité des piles). Si l'utilisateur indique un nombre de piles négatif ou nul, l'exception sera récupérée, et la saisie effectuée à nouveau.
2. Testez cette fonction dans votre programme principal.