

# TD Concepts objet et Java

2016-17

## Exercice 7

On souhaite mettre en place une classe paramétrée pour une structure de **pile** (LIFO (*last in first out*)) pouvant accueillir des objets d'un certain type. Une pile sera initialisée à l'aide d'un tableau du type approprié et de la taille voulue<sup>1</sup>. Un pointeur renseigne sur le premier élément nul (**null**) dans la pile. La pile doit permettre les types de requêtes suivants :

- l'interrogation sur son nombre d'éléments (**getTaille**) et sur son nombre d'éléments maximal, ou capacité (**getTailleMax**)
- la récupération de l'élément en haut de la pile et sa suppression (**pop**)
- l'ajout d'un élément en haut de la pile (**push**)
- son affichage dans un format lisible (**toString**)

1. Développez une classe répondant aux spécifications demandées. Testez les méthodes appropriées en affichant les états successifs de la pile.
2. Quels types d'erreurs peuvent survenir, et où ? Ces erreurs ont-elles un lien entre elles ? Enrichissez votre programme pour en réaliser le traitement au moyen du mécanisme d'**exceptions** (les exceptions devront être initialisées avec un message clair renseignant sur la nature de l'erreur), en distinguant aussi finement que nécessaire les différents types d'erreurs.
3. Aurait-il été approprié de déclarer ces exceptions comme dérivant de `java.lang.RuntimeException`, et pourquoi ?
4. On souhaite à présent pouvoir mettre en place une structure de pile dont les éléments sont triés : lors de l'ajout d'un élément, celui-ci sera inséré selon une notion d'ordre définie ; lors du retrait d'un élément, l'élément retourné reste celui qui se trouve sur le sommet de la pile. Quelle caractéristique devront avoir les types d'objets pouvant être mis dans la pile ?
5. Quelle doit être la signature appropriée pour une nouvelle classe répondant au besoin décrit ?
6. Proposez une implémentation d'une classe paramétrée **PileOrdonnee<T>**, et testez-la sur des exemples appropriés. Que se passe-t-il si une telle pile est créée pour des objets du mauvais type ?

---

1. Cela signifie qu'on pourra notamment invoquer le constructeur de la classe en lui transmettant un tableau anonyme du type et de la taille appropriés, ex : `new LaClasse(new Integer[20])` ;