

---

**Conception et programmation objets avancées**

---

**Sujet 6 : DP décorateur**

**Exercice 1** Bonbons d'Halloween

Un grand magasin a décidé, pour Halloween, de vendre des seaux et des bonbons de manière ludique pour occuper et distraire les enfants qui accompagnent leurs parents. À l'entrée du magasin, l'enfant choisit un récipient et le charge à sa convenance dans le magasin auprès de distributeurs devant lesquels il présente son seau. L'enfant peut charger son récipient autant qu'il le veut auprès de n'importe quel distributeur. Lors de cette opération, le prix du seau est mis à jour. Lors du passage en caisse, le seau est présenté et le prix affiché ainsi que la composition du seau (libellé).

Les seaux et recharges de bonbons disponibles sont :

- sachet sorcière : 0,5 €
- seau squelette : 1,5 €
- seau citrouille : 2,5 €
- recharge sucettes : 2 €
- recharge guimauves : 1,5 €
- recharge caramels : 1,75 €
- recharge pralines : 1,25 €

1. Si vous deviez créer une classe *SeauSquelette*, une classe *SeauCitrouille*, une classe *Sachet* puis toutes les combinaisons possibles de remplissage de bonbons, combien de classes devriez-vous créer ? Est-ce que cette solution est envisageable ?

2. Dessinez le diagramme de classe ré-utilisant le DP décorateur étudié en cours qui vous permet de modéliser votre problème. Quelles sont les deux méthodes que vous allez définir ?

3. Critiquez la solution proposée : quelle classe abstraite pouvez-vous remplacer par une interface ? Quel en est l'avantage ?

4. Entrez le diagramme de classe modifié (avec interface) dans votre IDE puis codez vos classes.

5. Créez une classe de test dans laquelle vous créerez un seau-citrouille avec double remplissage de guimauve et un remplissage de caramels. Faites afficher le libellé et le prix de votre seau.

6. Si le magasin installe en plus un distributeur de réglisses, que devez-vous faire ?

**Exercice 2** Reader et Writer

Les *reader* et *writer* en Java utilisent le design pattern décorateur.

1. Regardez la documentation des classes *InputStreamReader* et *BufferedReader*.

2. Écrivez une classe qui permet de décorer un *BufferedReader* en le chiffant selon la méthode de votre choix. Par exemple, dans la méthode ROT13, on effectue une rotation de 13 caractères sur les lettres. Les autres caractères (chiffres et symboles de ponctuation) ne sont pas changés.

3. Utilisez votre classe pour lire un mot de passe secret à l'écran, le chiffrer et l'afficher.

4. Écrivez une classe qui permet de décorer un *BufferedReader* en passant toutes les minuscules en majuscules.

5. Utilisez vos classes pour lire un mot de passe secret, le passer en majuscule et le chiffrer. Faites afficher le résultat.