

TD 1 — DEV 3.4

Benjamin Bordais

18 septembre 2026

1 *Setter et getters*

Pour chacune des deux classes ci-dessous, analyser quels seraient des *setters* et *getter* appropriés.

1. Afin de sécuriser l'accès à une application, différents utilisateurs sont authentifiés à l'aide d'un mot de passe. Définissez une classe `MotDePasse` qui permet de stocker un mot de passe par utilisateur.
2. Lorsque l'on définit une application sur laquelle des utilisateurs peuvent se connecter, on peut vouloir afficher le nombre d'utilisateurs en ligne. Définissez une classe `CompteurUtilisateur` qui se chargera de compter ce nombre d'utilisateurs.

2 Héritage inutile

Vous reprenez le code d'un logiciel pour une pizzeria dans lequel les différentes pizzas sont encodés en utilisant des classes qui héritent les unes des autres, en particulier pour calculer le prix. La classe `Pizza` représente le tarif de base, et chaque sous-classe ajoute son supplément, voir l'extrait de code ci-dessous.

```
1 public class Pizza {
2     // La pizza standard (Pate fine, base tomate) coute 8.0 euros
3     public double getPrix() {
4         return 8.0;
5     }
6 }
7
8 public class PizzaCroute extends Pizza {
9     @Override
10    public double getPrix() {
11        return super.getPrix() + 2.0;
12    }
13 }
14
15 public class PizzaCrouteBaseCreme extends PizzaCroute {
16     @Override
```

```

17     public double getPrix() {
18         return super.getPrix() + 1.5;
19     }
20 }
21 // ... et ainsi de suite
22

```

1. Supposons que l'on ait trois tailles pour chaque pizza (Petite, Moyenne, Grande) et une option supplément pesto. Combien de classes y aura t'il au total ?
2. Proposez un diagramme de classes pour modéliser cette situation qui n'utilise pas l'héritage. Ecrire une nouvelle classe Pizza (qui pourra utiliser d'autres classes que l'on suppose déjà écrites).

3 Diagramme de classes

1. On s'intéresse au fonctionnement d'une bibliothèque municipale. Cette bibliothèque contient de nombreux livres ; pour chacun d'entre eux, la bibliothèque garde son titre, son année de parution, son numéro ISBN, sa langue et son prix (le montant à rembourser en cas de perte de l'ouvrage). Un livre peut également avoir plusieurs sous-titres. Chaque livre est écrit par un ou plusieurs auteurs : un auteur a un nom et un prénom. Par ailleurs, un livre est édité par un éditeur, qui a un nom et un pays.
Proposez un diagramme de classe (assez haut niveau). Proposez ensuite un diagramme d'objet possible.
2. Construire le diagramme de classes (on s'intéresse notamment aux associations) du système de réservation de vols suivant (faites attention aux associations choisies) :
 - Des compagnies aériennes proposent différents vols.
 - Un vol est ouvert à la réservation et fermé sur ordre de la compagnie.
 - Un client peut réserver un ou plusieurs vols pour des passagers différents.
 - Une réservation concerne un seul vol et un seul passager.
 - Une réservation peut être annulée ou confirmée.
 - Un vol a un aéroport de départ et un aéroport d'arrivée.
 - Un vol a un jour et une heure de départ, et un jour et une heure d'arrivée
 - Un vol peut comporter des escales dans des aéroports.
 - Une escale a une heure d'arrivée et une heure de départ
 - Chaque aéroport dessert une ou plusieurs villes.
 - Une compagnie a un nom.
 - Chaque aéroport a un code spécifique
 - Un client a un nom, un prénom et une adresse.
 - Un passager a un nom, un prénom et un numéro de passeport.