

1. Les annotations

Créer une classe Java.

Créer une méthode statique `oldMethod` qui affichera un warning indiquant que la méthode est dépréciée.

2. Automobile

- 2.1. Coder la classe abstraite `Automobile` caractérisée par un modèle, une puissance, une couleur et un espace. Ajouter une méthode `afficherCaractéristiques()`.
- 2.2. Coder les classes `AutomobileElectrique` et `AutomobileEssence`.
- 2.3. Coder la classe abstraite `Scooter` caractérisée par un modèle, une couleur et une puissance. Ajouter une méthode `afficherCaractéristiques()`.
- 2.4. Coder les classes `ScooterElectricité` et `ScooterEssence`.
- 2.5. Coder une interface `FabriqueVéhicule` permettant de fabriquer des `Automobiles` et des `Scooters`.
- 2.6. Coder les classes `FabriqueVéhiculeElectricité` et `FabriqueVéhiculeEssence` implémentant `FabriqueVéhicule`.

3. Calculatrice

Nous souhaitons créer une API permettant d'implémenter une calculatrice en Java.

- 3.1. Créer une interface **Operation** définissant les opérations *addition*, *soustraction*, *multiplication* et *division*. Les méthodes doivent prendre en paramètre deux *Object* et retourner un *Object*.
- 3.2. Créer une classe abstraite **Affichage** dont le rôle est d'afficher un objet *Object*.
- 3.3. Créer une classe **Calculatrice** héritant de *Affichage* et implémentant *Operation* dont le rôle est de calculer un entier.
- 3.4. Nous souhaitons aussi une mise à jour pour calculer des réels. Créer une nouvelle classe **CalculatriceReel** permettant de calculer des *Double*.

4. API PrimeFlix (suite TP1)

PrimeFlix souhaite mettre à disposition une API Java afin de faciliter le travail de l'équipe de développeurs.

4.1. Modifiez votre précédent code en rajoutant des interfaces et classes abstraites nécessaires à l'évolutivité du code :

4.1.1. Un producteur et un client peuvent tous deux être considérés comme un utilisateur de la plateforme

4.1.2. Une série et une playlist sont une collection de vidéos