

TD n°6 – Héritage et polymorphisme

Ce TD a pour but de mettre en œuvre les notions d'héritage et polymorphisme en java en surchargeant/redéfinissant des méthodes.

Les points colorés

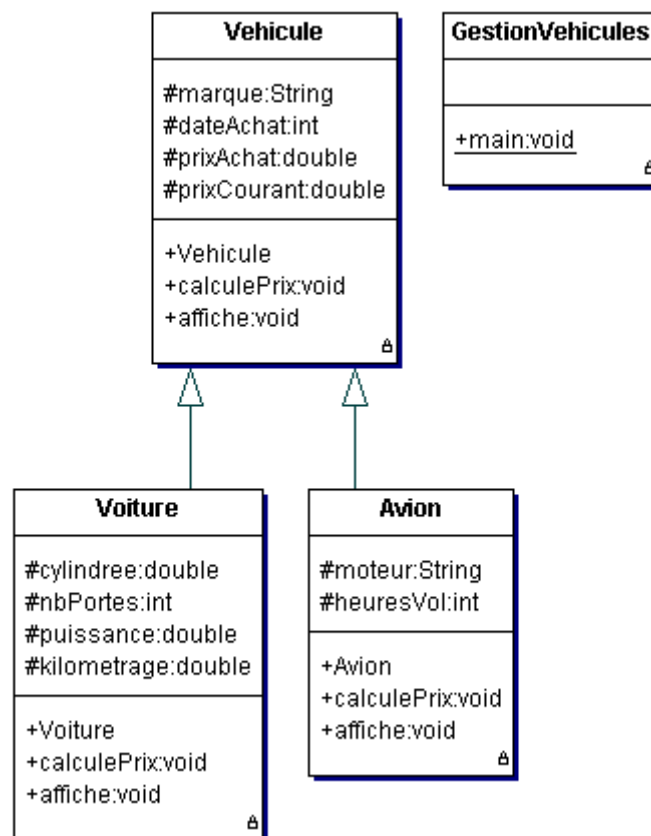
Reprendre la classe `ColoredPoint` du TD précédent et redéfinir les méthodes suivantes :

- `projX()` → renvoie un `ColoredPoint` projeté de la même couleur
- `projY()` → renvoie un `ColoredPoint` projeté de la même couleur

Surcharger les méthodes de projection dans la classe `ColoredPoint` afin de préciser en paramètre une nouvelle couleur pour le point projeté.

Ajouter dans le `main` des appels à ces nouvelles méthodes et vérifier l'exactitude des résultats.

Flotte de véhicules



Reproduire l'arborescence d'héritage ci-dessus.

1. La méthode `affiche()` permet d'afficher l'état de l'instance, c'est-à-dire la valeur de ses attributs. Cette méthode doit être redéfinie dans chaque classe.
2. La méthode `calculerPrix(int anneeActuelle)` dans la classe `Vehicule` fixe le prix courant égal au prix d'achat moins 1% par année (entre la date d'achat et la date actuelle).

- Redéfinissez cette méthode dans les deux classes dérivées `Voiture` et `Avion` de sorte à calculer le prix courant en fonction de certains critères, et mettre à jour l'attribut correspondant au prix courant.
 - Pour une voiture, le prix courant est égal au prix d'achat, moins :
 - 2% pour chaque année depuis l'achat jusqu'à la date actuelle
 - 5% pour chaque tranche de 10000 kms parcourus (on arrondit à la tranche la plus proche)
 - 10% s'il s'agit d'un véhicule de marque "Renault" ou "Fiat" (ou d'autres marques de votre choix)
 - et **plus** 20% s'il s'agit d'un véhicule de marque "Ferrari" ou "Porsche" (idem).
 - Pour un avion, le prix courant est égal au prix d'achat, moins :
 - 10 % pour chaque tranche de 100 heures de vol s'il s'agit d'un avion à hélices.
 - 10 % pour chaque tranche de 1000 heures de vol pour les autres types de moteurs.

Le prix doit rester positif (donc s'il est négatif, le mettre à 0). De même, le prix actuel ne doit pas être plus important que le prix d'achat.

3. Exécuter la classe `GestionVehicules` donnée dans le répertoire TD6 dans *SupportCours*. Vérifier l'exactitude des résultats obtenus.
4. Améliorez la méthode `main` ci-dessus en tenant compte du fait que tous les véhicules ont un type commun: `Vehicule`.
5. Améliorer la modélisation précédente en rendant *private* les attributs des différentes classes afin d'assurer une bonne encapsulation, et en ajoutant les méthodes *getters/setters* (accesseurs/mutateurs) afin de récupérer la valeur de chaque champ dans la classe correspondante.