

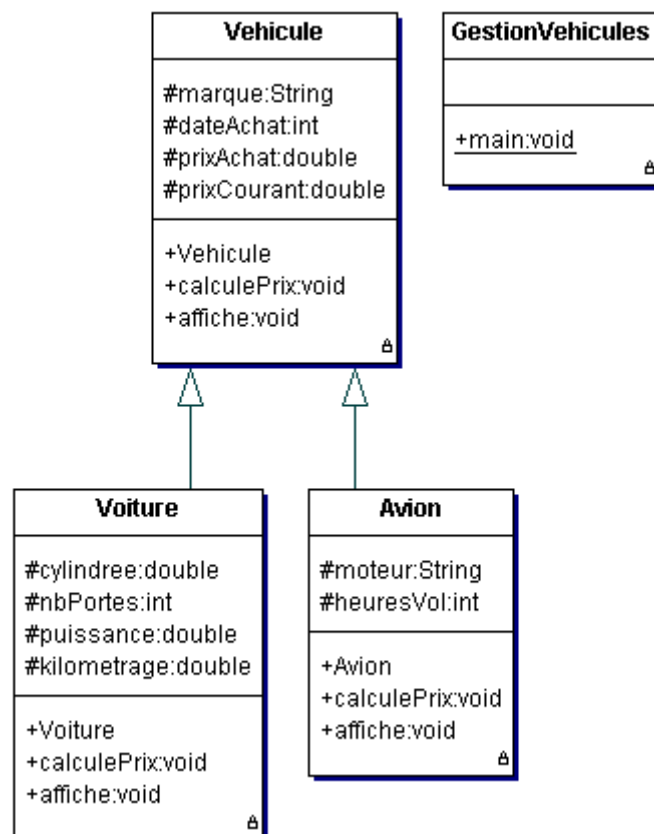
TD n°4 – Héritage

Les points colorés

Dans un nouveau package, créer la classe `ColoredPoint` vue en cours qui hérite de la classe `Point`. La compléter avec un accesseur de consultation de la couleur et une méthode `memmeCouleur` qui retourne *vrai* si le point coloré courant et celui passé en paramètre sont de la même couleur, *faux* sinon.

- Créer une classe pour la méthode `main`. Instancier des points colorés et tester les méthodes de la classe dérivée et de la classe de base.
- Redéfinir :
 - La méthode `equals` dans la classe `ColoredPoint` afin de tester l'égalité entre 2 points colorés. Les coordonnées ainsi que la couleur doivent être identiques.
→ Faire appel à `equals` de la classe `Point` et `memmeCouleur` de la classe `ColoredPoint`
 - la méthode `toString()` dans la classe `ColoredPoint` afin d'afficher en plus des coordonnées, la couleur du point.
→ Faire appel à `toString()` de la classe `Point`
 - `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.

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 `calculePrix(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 *SupportCours*. Vérifier l'exactitude des résultats obtenus.
4. 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.

Formes géométriques

A partir de la classe `Rectangle` disponible dans le répertoire TD5 sur *SupportCours*, écrire une classe `Carre` qui hérite de `Rectangle`. Cette nouvelle classe ne possède pas d'attribut spécifique.

- Ecrire un constructeur pour la classe `Carre` qui prend en paramètre la longueur d'un côté du carré. Comment faire pour faire correspondre cette caractéristique avec les attributs de la classe `Rectangle` ?
- Comme pour les points colorés, redéfinir dans `Carre` les méthodes de la classe `Rectangle` qui ne s'appliquent pas directement aux carrés.
- Créer une méthode `main` et instancier des carrés et des rectangles afin de tester les constructeurs et méthodes.