La programmation orientée objet avancée

Exercice 1 : Héritage, classes abstraites et interfaces

Implanter les classes Vehicule, Voiture et Train telles qu'elles ont été partiellement vues en cours.

1. Manipuler toutes les formes possibles de types pour les références et de type pour l'instance. Conclure quant aux manières possibles de référencer un objet.
2. Surcharger la méthode rouler() en lui ajoutant en paramètre un nombre de kilomètres.
3. Redéfinir la méthode rouler() dans les classes Train et Voiture. Illustrer cette redéfinition par le biais de tests.
4. Implanter la méthode rouler() sous forme de méthode abstraite dans la classe Vehicule. Conclure quant à la redéfinition précédente.
5. Définir une interface Pilotable qui permet de faire accélérer ou ralentir un Véhicule. Implanter cette interface.
6. Définir une interface Orientable qui permet de faire tourner à gauche ou à droite une un Véhicule. Implanter cette interface. Est-ce qu'un Train est orientable ?
7. Commander des instances de voitures et de train par leur interface Pilotable et Orientable.

Exercice 2 : Classes internes et anonymes

1. On désire maintenant prendre en compte la notion de moteur dans une voiture. Etant donné qu'un moteur n'a de sens que dans le cadre d'une voiture, on propose de l'implanter sous forme de classe interne (d'autres choix de modélisation seraient possibles). Implanter la classe interne Moteur.
2. Dans notre application, on désire créer une voiture unique dont l'implantation ne suit pas celle de la classe Voiture. Pour cela, on crée une classe anonyme, héritant de la classe Voiture.