Exercice 1 (polymorphisme)

Créer une classe `Individu` capable de répondre à la méthode `faitTesSalutations()` par l'affichage d'un message ordinaire (`Bonjour`).
Créer une sous-classe `IndividuFamilier` qui comme salutation affiche `Salut`.
Créer une sous-classes `IndividuTresPoli` qui comme salutation affiche `Bonjour mon/ma très chèr(e)`.
Créer une fonction recevant en paramètre la référence d'un individu et lui demande de faire ses salutations.
Tester sur différents individus...

Exercice 2 (polymorphisme)

Le jeu est constitué d'un plateau de 10 cases et d'un pion. On démarre sur la première case et l'utilisateur (tirant un dé par lui-même) avance son pion d'autant de cases que le dé l'indique. Si le pion tombe sur une case piège, le jeu est terminé et le joueur a perdu. Si le pion sort au-delà de la dernière case, le joueur a gagné.
Créer des classes `Case`, `Pion` et `Plateau` permettant de créer un plateau de 10 cases et de placer un pion sur la première case. On affichera le plateau (à l'aide de caractères). Permettre de déplacer le pion d'un nombre de cases déterminées. Simuler un jeu complet.
Créer une sous-classe de case `Piege` et créer un plateau dont la case numéro 5 est un piège. Changer ce qu'il faut pour que le jeu fonctionne (pion qui tombe sur la case piège, joueur perdant).

Exercice 3 (polymorphisme)

Créer une classe abstraite `Vehicule` (avec un attribut `marque`) et des sous-classes concrètes `Voiture` (attribut `nombreDePlaces`), `Camionette` (attribut `volumeUtile`) et les constructeurs, getters et opérateurs d'affichage adéquats.
Créer une classe `Parking` permettant de créer un parking de n emplacements pour véhicules. L'affichage d'un parking permettra de savoir quel emplacement est occupé ou non et pour chaque emplacement occupé, la marque et les caractéristiques du véhicule. Créer des méthodes permettant de parquer un véhicule ou vider un emplacement.
Tester.