TD3 B

Exercice 1 : Classes et Propriétés

Objectif : Revoir les classes, les constructeurs (primaire et secondaire), ainsi que les propriétés avec getters et setters en Kotlin.

1. Créer une classe **Voiture** avec les propriétés **marque, modèle, et année**.
2. Utiliser un constructeur primaire pour initialiser **marque** et **modèle**, et un constructeur secondaire pour initialiser également **année**.
3. Surdéfinir le getter de la propriété **année** pour empêcher qu'elle soit inférieure à 1886 (date d'invention de l'automobile).
4. Ajouter une méthode **démarrer** qui affiche un message.

Exercice 2 : Héritage, Surcharge, et Propriétés avec get/set

Objectif : Travailler sur l'héritage, la surdéfinition de méthodes et de getter/setter.

1. Créer une classe **Animal** avec une propriété **nom**, et une méthode **faireDuBruit()**.
2. Ajouter une méthode **description()** qui retourne la phrase "Je suis un animal".
3. Créer deux sous-classes, **Chien** et **Chat**, en surdéfinissant **faireDuBruit()** et **description()**.
4. Surdéfinir le getter de la propriété **nom** pour toujours retourner le nom en majuscules.

Exercice 3 : Constructeurs et Propriétés avec get/set

Objectif : Explorer les constructeurs primaires et secondaires, et la surdéfinition des getters/setters.

1. Créer une interface **Jouable** avec une méthode **jouer()**.
2. Créer une classe **Instrument** avec une propriété **volume** ayant un getter et un setter personnalisé pour s'assurer que le volume reste entre 0 et 100.
3. Utiliser un constructeur primaire pour initialiser volume, et un constructeur secondaire pour initialiser également une propriété type.
4. Créer deux sous-classes **Guitare** et **Piano**, et surcharger **jouer()**.

Exercice 4 : Constructeur Secondaire, Propriétés Nullable, et Smart Casts

Objectif : Gérer les types nullables, les constructeurs secondaires, et l'initialisation.

1. Créer une classe **Utilisateur** avec une propriété nullable **adresseEmail**: String?.
2. Ajouter un constructeur secondaire qui permet d'initialiser un utilisateur sans email.
3. Surdéfinir le getter pour renvoyer une valeur par défaut si l'email est null.
4. Implémenter une méthode pour envoyer un email si l'email est non-null, en utilisant les smart casts ou l'opérateur safe call (?.).

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/cnam/utc503/td3-b

Last update: **2025/08/12 02:35**