

## Série 3B – Héritage et tableau d'objets

1(\*) Créez une classe "Pays" avec les attributs suivants: le nom du pays, sa superficie en kilomètre carré, sa population en millions d'habitant, et le nom de sa capitale. Codez ensuite les méthodes suivantes:

- le constructeur.
- une méthode qui indique (vrai/faux) si le pays fait partie des dix plus grand pays du monde (en superficie). Pour être considéré comme un des dix plus grand pays, il faut que sa superficie dépasse ou égal les 2 505 810 [km<sup>2</sup>].
- une méthode retournant la densité de population du pays (c'est à dire le nombre d'habitant par kilomètre carré) arrondie à l'entier le plus proche.
- une méthode toString pour la représentation en String d'un pays (et de tous ces attributs).

2(\*) Créez une classe Livre avec les attributs suivants: le titre du livre, l'année de parution, le genre (roman, poésie, bibliographie, policier, ...), le prix minimum de ce livre en magasin, le prix maximum de ce livre en magasin. Codez ensuite les méthodes suivantes:

- le constructeur.
- une méthode qui indique (vrai/faux) si le livre est une nouveauté. Un livre est considéré comme une
- nouveauté si son année de parution est équivalente à l'année en cours (une constante valant 2011).
- une méthode retournant le prix moyen du livre en magasin (moyenne arithmétique des deux prix, arrondie au 5 centimes).
- une méthode toString pour la représentation en String d'un livre (avec tous les champs du livre).

3(\*) Créez le diagramme des classes nécessaires pour la représentation d'une partie du règne animal. Modéliser au moins 3 ou 4 niveaux d'héritage (ex: Animal -> Mammifère -> espèce -> race) et une dizaine de classes.

4(\*\*) Créez le diagramme des classes nécessaire pour représenter des voitures, des bateaux et des avions. Les voitures auront comme fonctionnalité de rouler, les avions de voler et les bateaux de naviguer. Tous les véhicules auront comme attributs un nom et une vitesse maximum. Ajouter ensuite une classe pour y représenter une voiture amphibie.

5(\*) Créez une classe MyColor dérivée de Color qui ne fait que surcharger (override) la méthode toString pour un affichage de la couleur en français au format suivant : composante rouge : xxx, composante verte : xxx, composante bleue : xxx

6(\*\*) Faites le diagramme des classes et codez les classes en JAVA à partir des fonctionnalités souhaitées suivantes:

- On doit pouvoir créer des bateaux de transport. Un bateau possède:
  - o un nom
  - o une année de construction
  - o Un certain nombre de container (sous la forme d'un tableau de containers)

La compagnie possède aussi quelques bateaux particuliers qui ne peuvent transporter qu'un type de marchandise (un bateau ne transportant que des denrées alimentaires par exemple). Ces bateaux ont alors un champ de plus que les bateaux "classique": le type de marchandises transportées (alimentaire, textile, ...)

- On doit pouvoir créer des containers. Un container possède:
  - o un identifiant numérique de 8 chiffres (ex: 12223976)
  - o la valeur marchande des produits stockés dans le container (en CHF)

Une fois les classes faites, créez dans votre programme (dans le « main ») un tableau de quelques containers et ajoutez ce tableau de containers dans un bateau. Puis affichez le bateau et la liste des containers du bateau (grâce la méthode toString de la classe Bateau). Ajoutez ensuite les méthodes suivantes dans la classe Bateau :

- Une méthode calculant la somme des valeurs marchandes des containers transportés.
- Une méthode retournant le container du bateau ayant la plus forte valeur marchande.