

Serie 2 TP C++ (suite)

Exercice 3: Les variables static

1. Créer une classe Livre avec les attributs suivant: Auteur, titre, id
2. Créer une classe Etagère avec les attributs suivants: livre1, livre2, livre3, count qui calcule le nombre des livres dans la bibliothèque, il va nous aider à affecter des ids uniques pour chaque livre.
3. Créer les constructeurs par défaut et d'initialisation pour les deux classes.
4. Créer les setters public pour auteur et titre, ainsi que le setter de l'id privé
5. la fonction afficher pour les deux classes
6. Créer Les setters pour les trois livres à l'étagère.
7. Créer un programme de test.

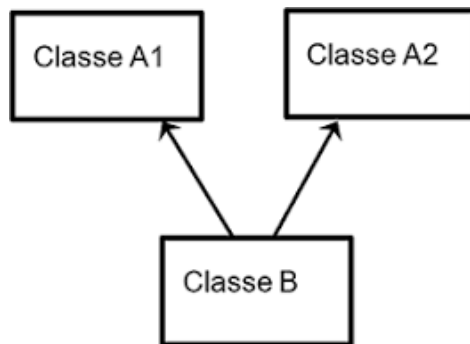
Exercice 4: L'Heritage simple

Les différents types d'accès dans l'heritage:

		Statut dans la classe de base	Statut dans la classe dérivée
mode de dérivation	public	public	public
		protected	protected
		private	inaccessible
	protected	public	protected
		protected	protected
		private	inaccessible
	private	public	private
		protected	private
		private	inaccessible

1. Créer une classe personnel comprenant 3 données membres (nom, position, salaire), un constructeur initialisant les données membres et une fonction affiche_personnel qui affiche les informations sur le personnel.
2. Créer une classe patron dérivant publiquement de la classe personnel et ayant 2 données membres (bonus_annuel, marque de voiture de fonction) , un constructeur initialisant les données membres et une fonction affiche_patron qui affiche les informations sur le patron (entant que membre du personnel et patron). Créer un programme permettant de tester ces classes.

Exercice 5: L'Héritage multiple



a) Créer une classe objet assure comprenant :

- les données membres : montant (float), type (chaîne de caractères)
- un constructeur initialisant les données membres
- une fonction affiche qui affiche les informations sur l'objet assure

b) Créer une classe ordinateur dérivant publiquement de la classe objet assure et comprenant :

- les données membres : ram (int), hd(int), ...
- un constructeur initialisant les données membres
- une fonction affiche qui affiche les informations sur l'ordinateur et qui appelle la fonction affiche de la classe objet assure

c) Créer une classe bagage dérivant publiquement de la classe objet assuré et comprenant :

- les données membres : type (chaîne de caractères), poids (float)
- un constructeur initialisant les données membres
- une fonction affiche affichant les informations sur le bagage et appelant la fonction affiche de la classe objet assure.

d) Créer une classe ordinateur portable dérivant publiquement des classes ordinateur et bagage et comprenant :

- les données membres poids (float) et epaisseur (float)
- un constructeur initialisant les données membres
- une fonction affiche qui affiche les informations sur l'ordinateur portable et qui appelle les fonctions affiche des classes bagage et ordinateur

e) Créer un programme qui crée un objet ordinateur portable et qui affiche les informations sur cet objet

Remarque : l'ordinateur portable doit bénéficier des 2 assurances, l'une contre le vol (en tant qu'ordinateur) et l'autre contre la perte (en tant que bagage).

Correction: <https://github.com/hm43/ExercicesTpCpp.git>