



HLIN603

Feuille de TD/TP N°2 : Héritage multiple en C++

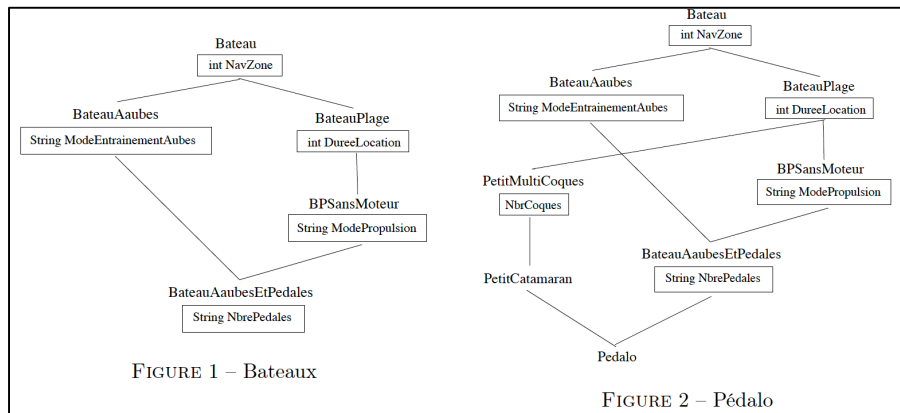
Exercice 1 : Hiérarchie d'héritage de comptes bancaires

Cet exercice reprend une partie de la hiérarchie d'héritage définie dans l'exercice 3 de la feuille de TD/TP N°1.

- 1) Proposez trois classes C++ représentant respectivement :
 - Une classe **CompteBancaire** disposant d'un attribut **solde** et d'un destructeur qui affiche la valeur du solde que la banque est supposée rendre au client lors de la fermeture du compte.
 - Une classe **CompteRemunere** représentant les comptes auxquels on sert un intérêt. En particulier, lors de la fermeture, le solde est augmenté de 10%.
 - Une classe **CompteDepot** représentant les comptes de dépôt classiques. Lors de la fermeture, on prélève sur ces comptes des frais de gestion de 100 euros.
 - Une classe **CompteDepotRemunere**.
- 2) Représentez une instance de chaque classe dans le cas de l'héritage non virtuel, puis dans le cas de l'héritage virtuel.
- 3) - Ajoutez une méthode **deposer** dans **CompteBancaire**, spécialisez-la dans le cas des **CompteRemunere** en ajoutant un intérêt de 1% à la somme déposée, et dans le cas des **CompteDepot** en retirant 1 euro de frais de gestion et en ajoutant 10 euros si le dépôt est supérieur à 1000 euros. Effectuez un dépôt sur une instance de **CompteDepotRemunere**.
 - Que se passe-t-il dans chacun des cas d'héritage (virtuel ou non) ?
 - Comment pouvez-vous résoudre le problème ?
- 4) Donnez l'ordre des opérations effectuées lorsqu'un compte est fermé (l'instance est "détruite" au sens de C++) dans chacun des cas d'héritage (virtuel ou non).
- 5) On suppose que **CompteDepotRemunere** possède une méthode **deposer**, et hérite "d'abord" de **CompteRemunere** puis de **CompteDepot**.
 - Ajoutez une classe **CompteDepotAvecCarteCredit** et une classe **CompteDepotRemunereEtAvecCarteCredit** qui hérite "d'abord" de **CompteDepotAvecCarteCredit**.
 - Dans le cas de l'héritage virtuel, que se passe-t-il lorsqu'on ferme un compte de la classe **CompteDepotRemunereEtAvecCarteCredit** ?

Exercice 2 : Hiérarchie d'héritage de bateaux

Construisez la hiérarchie de la Figure 1, d'abord sans faire d'héritage virtuel, puis avec de l'héritage virtuel.



Combien d'attributs possède une instance de **BateauAaubesEtPedales** dans chacun des deux modes d'héritage ? Comment pouvez-vous le vérifier ?

À partir de maintenant et pour toute la suite, on ne fait plus QUE de l'héritage virtuel.

Des effets secondaires de l'ordre d'appel des constructeurs et destructeurs.

- Construisez la hiérarchie de la Figure 2. Dans les cas d'héritage multiple, déclarez les classes mères dans l'ordre "gauche_à droite" du dessin. Pour simplifier, chaque classe ne contient, en dehors de ses attributs, qu'un constructeur sans paramètres et un destructeur.
- Dans le constructeur de la classe **BateauPlage** (resp. **BateauAaubes**), initialisez l'attribut **NavZone** avec la valeur 1 (resp. 100). Cet attribut représente la zone de navigation autorisée, c'est à dire la distance maximum d'éloignement du rivage (en kilomètres).
 - o Créez une instance de la classe **BateauAaubesEtPedales**, une de la classe **PetitCatamaran**, et une de la classe **Pedalo**.
 - o Affichez la valeur de **NavZone** pour ces trois objets. Est-ce que tout vous semble normal ?
 - o Mettez un affichage dans chaque constructeur. Est-ce plus clair ?

Tester la résolution des conflits

- Ajoutez les méthodes prévues dans la Figure 3.

