

Programmation C++

Parcours MF– ING2

TP4 : Héritage simple

Objectifs

– Apprehender la notion d'héritage par des exemples typiques d'utilisation.

1 – Le compteur borné

On va opérer une modification du cahier des charges de la classe Compteur : notre client désire maintenant pouvoir spécifier, lors de la création d'un compteur, une borne maximale que le compteur ne pourra pas dépasser. On va pour cela créer une classe dérivée de la classe précédente qu'on appellera CompteurBorne.

Définir un constructeur permettant de fournir une borne maximale éventuelle à un compteur, et mettre à jour la structure et les méthodes nécessaires pour prendre en compte cette notion de borne.

Modifier le programme de test pour qu'il effectue la même série d'opérations sur un compteur normal et sur un compteur borné.

2 - Le coursier

Nous allons simuler le travail d'un coursier dans une grande entreprise. Ce coursier passe tous les matins dans les bureaux et récupère le courrier à affranchir. Ce courrier est de deux types, avec une différence au niveau du processus d'affranchissement:

- les lettres, qui peuvent être ordinaires (0,5 euro) ou urgentes (+0,3 euro). On convient que la lettre représente 1 unité de volume.
- les colis, dont l'affranchissement dépend du volume (0,5 euro / unité de volume).

Le coursier place les courriers dans un grand sac (de capacité fixée à sa création). De retour dans son bureau, il place le sac dans une machine à affranchir (qui affranchit automatiquement les courriers contenus dans le sac en fonction de leur type). On demande de modéliser la tournée du coursier en

- créant un sac et quelques courriers;
- plaçant les courriers dans le sac;
- affranchissant le sac;
- affichant le tarif des différents courriers.

Dans un premier temps, nous implémentons le sac à l'aide d'un tableau. On supposera qu'on dispose de la fonction AleaN() qui renvoie un nombre réel aléatoire entre 0 et 1.

3 – Villes et capitales

Ecrire une classe Ville et une classe Capitale qui en hérite avec les instructions suivantes :

- Une ville est décrite par son nom et son nombre d'habitants.
- Le nom d'une ville ne peut pas varier ; Il doit être connu dès l'instanciation de l'objet.

Par la suite, ce nom servira de clé de recherche d'informations.

- Les villes sont classées en 3 catégories : grande (plus de 500 000 habitants), moyenne (entre 100 000 et 500 000) ou petite (moins de 100 000).
- La classe Ville doit fournir une méthode "obtenirInformations()" qui retourne les informations sous la forme d'une chaîne de caractères.
- Une capitale est une ville et contient l'information du pays où elle est située. Il faut donc surcharger la méthode obtenirInformations() pour rajouter cette dernière information.

Exemple:

v : Ville

```
v ← nouveau Ville("Les Pennes-Mirabeau",19022)
Ecrire( v.obtenirInformations() )
a pour effet d'afficher :
les pennes-mirabeau : 19022 habitants, petite ville.
Autre exemple:
c : Capitale
c ← nouveau Capitale("Paris",2153600,"France")
Ecrire( c.obtenirInformations() )
affiche :
paris : 2153600 habitants, grande ville. Capitale de France.
```

4 – Mots et termes

Ecrire une classe Mot avec les instructions suivantes:

- La classe Mot a un seul attribut : une chaîne de caractères.
- L’instanciation d’un mot est possible à partir d’une chaîne de caractères (attention: le mot doit être stocké en minuscules).
- On doit pouvoir accéder à cette chaîne de caractères.
- On doit pouvoir comparer le mot courant avec un autre mot et retourner -1 (resp. 0 ou 1) si le premier mot est de longueur inférieure (resp. égale ou supérieure) au deuxième
- On doit pouvoir afficher son contenu en l'encadrant d'astérisques “*”

Ecrire une classe Terme qui hérite de la classe Mot, avec les instructions suivantes:

- La classe Terme a un certain nombre de définitions, regroupées en un tableau.
- Elle possède sa propre méthode d'affichage qui retourne le terme accompagné de ses définitions (sur une seule ligne).
- Elle surcharge la méthode d'affichage de sa superclasse en rajoutant ligne par ligne l'ensemble de ses définitions.

Exemple: Si on rentre le code suivant :

```
t : Terme
t.ajouteDefinition("Ile indonesienne")
t.ajouteDefinition("Langage orienté objet")
alors la méthode d'affichage de t doit rendre la chaîne suivante :
java[2] ← "Ile indonésienne" "Langage orienté objet"
```

et l'appel de méthode Affiche() a pour effet d'afficher la chaîne suivante :

```
*****
```

```
* J a v a *
```

```
.....
```

```
1. Ile indonésienne
```

```
2. Langage orienté objet
```

5 – Habitations individuelles et professionnelles

Dans le cadre de l’informatisation d’une mairie, on veut automatiser le calcul des impôts locaux. On distingue deux catégories d’habitation : les habitations à usage professionnel et les maisons individuelles, l’impôt se calculant différemment selon le type d’habitation. Pour cela, on définit les classes HabitationProfessionnelle et HabitationIndividuelle et les caractéristiques communes à ces deux classes sont regroupées dans la classe Habitation.

5.1/ Ecriture de la classe Habitation :

La classe Habitation comprend les attributs : propriétaire du type chaîne de caractères et qui correspond au nom du propriétaire, adresse du type chaîne de caractères et qui correspond à l’adresse de l’habitation,

surface du type Réel et qui correspond à la surface de l'habitation et qui permet de calculer le montant de l'impôt. Elle possède aussi les méthodes :

- Un constructeur qui initialise le propriétaire, l'adresse et la surface
- Une fonction Impot() qui permet de calculer le montant de l'impôt que doit payer le propriétaire de l'habitation à raison de 2 euros par m2.
- Une procédure Affiche() qui permet d'afficher les trois attributs de la classe Habitation.

5.2/ Ecriture des classes HabitationIndividuelle et HabitationProfessionnelle :

Le calcul de l'impôt d'une maison individuelle est différent de celui d'une habitation, il se calcule en fonction de la surface habitable, du nombre de pièces et de la présence ou non d'une piscine. On compte 15 euros/pièce et 80 euros supplémentaire en cas de présence d'une piscine.

Par ailleurs, le calcul de l'impôt d'une habitation à usage professionnel est également différent de celui d'une habitation. Il se calcule en fonction de la surface occupée par le bâtiment et du nombre d'employés travaillant dans l'entreprise. On compte 150 euros supplémentaire par tranche de 10 employés.

Ecrire les classes HabitationIndividuelle et HabitationProfessionnelle. On redéfinira si nécessaire les méthodes Impot() et Affiche() pour afficher les nouveaux attributs, et pour recalculer l'impôt.