

Programmation par objets 2 - FLIN 606 - 5 mai 2010

8h30-10h30 - Tous documents autorisés

1 Généricité paramétrique : Transport

Nous disposons d'une classe représentant les colis dont voici un extrait :

```
class Colis
{
private:
    double poids;
    String origine;
    String destination;
    String identification;
    //.....
public:
    //.....
    virtual double poids(){return poids;}
    virtual String origine(){return origine;}
    virtual String destination(){return destination;}
    virtual String identification(){return identification;}
};
```

Nous disposons également d'une classe représentant les transports (camion, avion, etc.) de colis. Un transport a une cargaison qu'il livre à certaines destinations.

```
class Transport
{
private:
    vector<Colis> cargaison;
public:
    //.....
    virtual void embarque(Colis c); // ajoute le colis c à la cargaison
    virtual String listingCargaison()const; // retourne une chaîne de caractère contenant les identifi
    virtual vector<Colis> livrer(String destination); // retourne un vecteur contenant les colis à liv
};
```

Question 1 Complétez la classe *Transport* en ajoutant constructeur, destructeur et code des méthodes.

Question 2 Transformez la classe *Transport* en classe générique paramétrée (template class) par le type des éléments transportés.

Question 3 Quelle(s) contrainte(s) devrait respecter le type paramétrique ? Peut-on (1) représenter cette contrainte par une classe abstraite ? (2) assurer qu'un paramètre réel pour la classe générique (template) pour que la contrainte soit respectée ? Comparez l'approche de C++ sur ce problème avec celle de Java.

Question 4 Proposez une classe *Personne* très succincte qui vérifie la contrainte. Définissez un *main* dans lequel vous créez un transport de personnes, embarquez quelques individus et effectuez une livraison à la destination "Pôle Nord".

2 Héritage multiple : Véhicules

On veut représenter quelques classes représentant des véhicules :

- la classe générale *Véhicule*, qui comprend un attribut *marque*
- la classe *VéhiculeAérien* représentant les véhicules qui peuvent circuler dans l'air qui comprend un attribut *hauteurVol*
- la classe *VéhiculeAquatique* représentant les véhicules qui peuvent circuler sur et dans l'eau, qui comprend un attribut *profondeur*
- la classe *VéhiculeTerrestre* représentant les véhicules qui peuvent circuler sur la terre
- la classe *VéhiculeAmphibie* représentant les véhicules qui peuvent circuler sur la terre et sur l'eau, qui comprend un attribut *type*
- la classe *Hydravion* représentant les hydravions, qui est sous-classe (avec cet ordre de priorité) de *VéhiculeAérien*, *VéhiculeAmphibie*

Question 5 *Pour le cas de l'héritage virtuel puis de l'héritage répété (sur tous les liens) :*

- *Ecrivez les entêtes des classes*
- *Indiquez dans quel ordre seront appelés les constructeurs pour une instance de Hydravion*
- *Dessinez un schéma mémoire d'une instance de Hydravion*
- *Discutez de la conception de l'attribut marque.*
- *Comment traiteriez-vous un attribut vitesseMaximale qui indiquerait la vitesse maximale suivant le milieu de déplacement du véhicule : quelles sont les conséquences s'il était dans la classe Vehicule ? Dans certaines des sous-classes et lesquelles ?*

3 Héritage simple : Appartements

Nous nous plaçons dans l'hypothèse de la réalisation d'un logiciel destiné à une agence immobilière chargée de gérer des locations d'appartements.

Pour les besoins de cette gestion, les appartements sont décrits par une adresse, une année de construction, une superficie et un nombre de pièces. La valeur locative de base est calculée de manière générale avec la formule : $superficie * 5 * (1 + nbPieces / 10)$. Le loyer se calcule comme le produit de la valeur locative de base par un coefficient modérateur.

On distingue la sous-classe des appartements de luxe pour lesquels on enregistre le quartier. Le coefficient modérateur vaut 1.1 pour tous les appartements de luxe.

Une feuille de loyer comprend l'adresse, la valeur locative de base, le coefficient modérateur, et la valeur du loyer. A ces informations s'ajoute, dans le cas des appartements de luxe, le quartier.

Ecrivez le code des deux classes, **Appartement** et **AppartementDeLuxe** d'après les informations ci-dessus, avec au moins une méthode retournant le loyer et une méthode retournant une chaîne de caractères décrivant la feuille de loyer. Des méthodes auxiliaires peuvent être ajoutées (valeur locative de base, coefficient modérateur). N'écrivez pas les accesseurs que vous supposerez néanmoins exister. Les attributs seront privés.