

Universite Paris Diderot

Langages à Objets Avancés: TP4 : Héritage + UML

November 4, 2012

1 Mise en bouche

1.1 POO: héritage

soit la classe suivante:

```
Class Base
```

```
{
```

```
public
```

```
virtual void print() { cout << "Base" << endl;
```

```
}
```

derivez deux classes de Base . Pour chacune d'entre elles, définissez print() de telle sorte qu'elle affiche le nom de la classe.

Creez des objets de ces classes et appelez print().

Affectez des pointeurs d'objets de classes derivees a des pointeurs Base* et appelez print() via des pointeurs.

1.2 POO: Classe Abstraite

Declarer une Classe Shape .

aucun objet de cette classe ne peut être cree.

Cette classe possede les methodes: void print(), bool is_closed(), int area()

Derivez deux classes de Shape : cercle et polygone de Shape. Prenez soin de définir chaque methode.

2 UML

Dans cette section on se propose d'implementer un mini reseau social.

Un utilisateur est défini par son nom, prénom, adresse, date de naissance, profession ...

Un utilisateur peut ou non faire partie d'un ou plusieurs groupes .

Un groupe est défini par son nom, sa date de création , un admin et ses membres.

Les utilisateurs communiquent entre eux via des conversations instantanées ou pas !

Une conversation instantanée ne peut avoir lieu que si les utilisateurs concernés sont tous connectés.

Une conversation est composée de messages , une conversation vide n'a pas de sens .

Un historique des actions d'un utilisateur est composé d'un trace de toutes les modifications apportées par un utilisateur donné.

Proposer un diagramme de classe pour votre réseau social.

2.1 Implémentation

Implémenter votre diagramme en C++ .

Tester votre code avec un réseau social composé de 15 personnes et 3 groupes.

Ajouter les historiques de conversations des utilisateurs du 1er groupe.