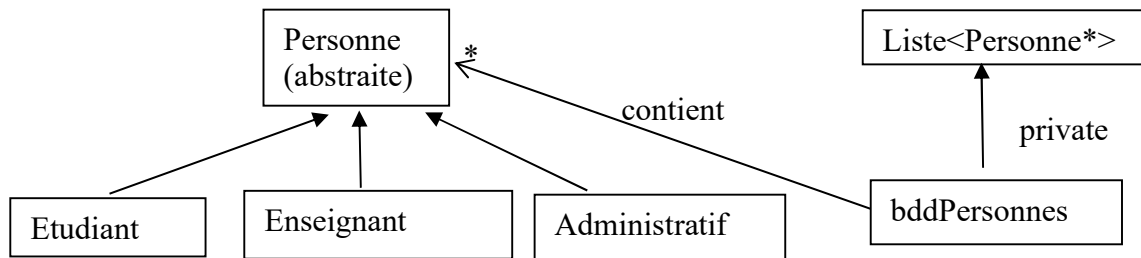


TP - BDD des personnes

Ce TP a pour but d'expérimenter l'héritage sur la classe abstraite `Personne` et de réutiliser les objets construits dans les TP précédents. La base de données (`bddPersonnes`) des personnes est une liste de personnes comme l'indique le schéma d'héritage :



La classe `Personne` est une classe abstraite :

```
class Personne {
    string nom ;
    ...
public :
    Personne(string nm, string pnm) : nom(nm), prenom(pnm) {} ;
    ...
    virtual string Nom() { return nom ; }
    virtual etablisement() { cout << "UTBM" ; }
    virtual qualite() = 0 ;
    virtual experience() = 0 ;
    virtual emploi_du_temps() = 0 ;
} ;
```

Elle contient des méthodes virtuelles pures dont le comportement doit être défini dans les classes dérivées, des méthodes virtuelles ayant un comportement par défaut, et éventuellement des méthodes simples qui ne seront jamais réécrites. Par exemple, la classe `Etudiant` peut ressembler à ceci :

```
class Etudiant : public Personne {
    ...
public :
    Etudiant(string nm, string pnm) : Personne(nm, pnm) {} ;
    ...
    virtual etablisement() {
        Personne::etablisement();
        cout << "Etudiant" ;
    }
    virtual qualite() {...} ;
    virtual experience() {...} ;
    virtual emploi_du_temps() {...} ;
} ;
```

Dans les classes dérivées les méthodes virtuelles sont complétées et/ou réutilisées, elles spécifient le comportement spécialisé de la classe en question. On n'oubliera pas d'utiliser la liste d'initialisation pour les constructeurs. La base de données complète sera utilisée dans un programme de test `main.cpp` avec des instructions du style :

```
main(){ bddPersonnes b; Personne* p = new Etudiant("toto", ...);
b.ajouter(p); ... ; b.statistiques(); }
```

où la méthode `b.statistiques()`, par exemple parcourt la base et renvoie les informations sur les personnes obtenues en utilisant l'interface abstraite.