

## TP N° : 10

### Exercice 1:

Soit les deux classes suivantes:

```
class A
{ protected:
    int i;
public:
    A(int k)
    { i=k ;
      cout << " Constructeur A " << i << endl ;
    }
    ~A()
    { cout << " Destructeur A " << endl ;
    }
};

class B : public A
{ private:
    int j;
public:
    B(int k, int n = 1): A(k)
    { j=n ;
      cout << " Constructeur B " << j << endl ;
    }
    ~B()
    { cout << " Destructeur B " << endl ;
    }
};
```

- 1) Que génère le programme suivant?

```
int main()
{ A a (1) ;
  B b(2) ;
  return 0;
}
```

- 2) Que génère le programme suivant?

```
int main()
{ A * a, *b ;
  a = new A(10) ;
  b = new B(20,30) ;
  delete a ;
  delete b ;
  return 0;
}
```

- 3) Définir, pour les deux classes, la méthode *affiche()* qui affiche la valeur des membres privés ou protégés. Pour les deux classes, cette fonction doit avoir la même signature.  
4) Définir ensuite une fonction *afficheA(A\* t)* qui invoque la méthode *affiche()* de t.  
Que génère le programme suivant?

```
int main()
{ A * a, *b ;
  a = new A(10) ;
  b = new B(20,30) ;
  afficheA(a) ;
}
```



```
afficheA(b) ;
delete a ;
delete b ;
return 0;
}
```

5) Que faudrait-il faire pour que la méthode *affiche* de B imprime également la valeur de A?

## PARTIE 1 :

Une entreprise souhaite gérer les salaires de ces employés. Pour cela, elle crée une classe

**Employe** ayant les caractéristiques suivantes :

- Une donnée membre char\* nom
- Une donnée membre char\* prénom
- Une donnée membre int age
- Une donnée membre int anciennete (le nombre d'années de la personne dans l'entreprise)
- Une méthode double calculer\_base\_salaire() qui calcule la base du salaire de la personne.
- Une méthode void afficher() qui affiche la personne avec sa profession
- Une méthode double calculer\_salaire() qui calcule le salaire de la personne (base de salaire plus 10% par année d'ancienneté)
- Des constructeurs associés (par défaut et avec paramètres) et le constructeur par copie
- Un destructeur
- Un opérateur d'affectation

On souhaite pouvoir définir les classes dérivées de Employe suivantes :

- Une classe **Commercial** ayant une donnée membre double chiffre\_affaire et deux données statiques constantes double BASE et double PART. La base de salaire est donnée par la formule suivante :

**BASE+chiffre affaire\*PART**

- Une classe **Technicien** ayant une donnée membre int nb\_unité\_produite et trois données statiques constantes double BASE, double PART et double GAIN\_UNITE. La base de salaire est donnée par la formule suivante :

**BASE+nb unité produite\*PART\*GAIN UNITE**

- Une classe **Manutentionnaire** ayant une donnée membre int Nb\_heures et une donnée statique constante double BASE\_SALAIRE\_HORAIRE. La base de salaire est donnée par la formule suivante :

**BASE SALAIRE HORAIRE\*Nb heures.**

1. Quelles sont les fonctions qui doivent être déclarées en tant que fonctions virtuelles dans la classe Employé ?
2. Quelles sont les fonctions qui doivent être déclarées en tant que fonctions virtuelles pures dans la classe Employé ? La classe Employe est-elle une classe abstraite ?
3. Implémenter les différentes classes et tester-les.

## PARTIE 2:

On souhaite créer de nouvelles classes dérivées de Commercial :

- **Vendeur** dont le salaire est calculé comme un Commercial

– **Représentant** dont le salaire est calculé comme un Manutentionnaire

1. Que faut-il changer à la classe Commercial pour implémenter ces deux classes ?
2. Implémenter ces deux classes et tester-les.
3. Il-y-t-il de la duplication de code entre Manutentionnaire et Représentant ? Si oui, comment y remédier ?

### PARTIE 3:

On souhaite créer de nouvelles classes dérivées d'Employé correspondant aux employés avec prime de risque (un montant différent pour chaque employé qui est rajouté à la base de salaire) : **ManutentionnaireARisque** et **TechnicienARisque**.

1. Que faut-il changer aux classes Manutentionnaire et Technicien pour implémenter ces deux classes ?
2. Comment éviter la duplication de code pour la gestion des primes de risque dans ces deux classes ?
3. Implémenter ces deux classes et tester-les.

### PARTIE 4:

On souhaite créer une classe **Personnel** contenant :

- Un tableau d'Employés (sous forme d'un pointeur).
- Une méthode void embaucher(Employé\* newbie) qui ajoute un employé
- Une méthode void licencie(Employé\* e) qui supprime l'employé e
- Une méthode void licencie() qui supprime tous les employés
- Une méthode void afficher\_salaires() qui affiche le salaire de tous les employés
- Une méthode double salaire\_moyen() qui calcule le salaire moyen des employés
- Un constructeur
- Un destructeur

1. De quelles façons peut-on déterminer si un employé e est dans le vecteur extensible de Personnel ? Faut-il rajouter une méthode à la classe Employé ?
2. Implémenter cette classe et tester-la.