

**Exercice 2 :** Une bibliothécaire vous demande de lui développer une application C++ lui permettant de gérer la situation suivante : Un livre est écrit par un ou plusieurs auteurs. Dans une bibliothèque, il est possible de classer différents types de livres.

Il s'agit d'implémenter l'ensemble des classes et d'ajouter **toutes les méthodes nécessaires à leur bon fonctionnement**. Il est à noter que les attributs des classes à définir doivent être non publiques. Définissez, en C++, une **interface (déclaration)** et une **implémentation (définition)** de ces classes.

- 1- Un auteur est caractérisé par un numéro (int) et un nom (string). La classe auteur offre principalement :
  - Un constructeur initialisant les données membres de la classe,
  - les fonctions permettant d'accéder et de retourner à la donnée membre « numero »,
  - Une méthode Affiche () permettant d'afficher les caractéristiques d'un auteur.

#### Exercice 2

\*\*\*\*\*Question 1\*\*\*\*\*

```
class auteur
{
    int numero;
    string nom;
public:
    auteur(int =999, string="");
    void setNumero(int n){ numero=n;}
    int getNumero(){ return numero;}
    void afficher(string = "");
    ~auteur(void);
};

auteur::auteur(int n, string no):numero(n), nom(no)
{
}
void auteur::afficher(string msg)
{
    cout<<msg<<endl;
    cout<<"\n numero: "<<numero<<" nom: "<<nom<<endl;
}
auteur::~auteur(void)
{
}
```

\*\*\*\*\*Question 2\*\*\*\*\*

- 2 – Un livre est caractérisée par un numéro ISBN, un titre, un prix, un nombre d'auteurs et les auteurs. Un livre peut être rédigé par un ou plusieurs auteurs: **utiliser la classe vector**.

- Définir les constructeurs nécessaires, - Une méthode addAuteur permettant d'ajouter un auteur,
- et une méthode Affiche () permettant d'afficher les caractéristiques d'un livre et ses auteurs.

```

class livre
{
protected:
    int ISBN;
    string titre;
    float prix;
    int nbAuteurs;
    vector<auteur*>tab;
public:
    livre(int =99, string="POO", float=20.0, int =1);
    virtual void afficher(string = ""); // virtual pour Question 3
    void addAuteur(auteur, int =0);
    livre(const livre&); // constructeur de copie
    livre& operator=(const livre&); // surcharge affectation
    virtual ~livre(void); // virtual pour Question 3
};
livre::livre(int i, string t, float p, int nb)
{

    ISBN=i;
    titre=t;
    prix=p;
    nbAuteurs=nb;

    auteur *q;
    for(int j=0; j<nb; j++)
    {
        q=new auteur;
        q->saisir();
        tab.push_back(q);
    }

}

livre::~~livre(void)
{
    for(int i=0; i<tab.size(); i++)
        delete tab[i];
    tab.clear();
}

void livre::afficher(string msg)
{
    cout<<"\n ISBN: "<<ISBN<<" titre: "<<titre<<" prix: "
        <<prix<<" nbre auteurs: "<<nbAuteurs<<endl;
    for(int i=0; i<tab.size(); i++)
        tab[i]->afficher();
    // appel de afficher de la classe auteur
}

```

```

void livre::addAuteur(auteur a, int ind)
{
    auteur *q=new auteur(a);
    tab.insert( tab.begin()+ind, q);
}

livre::livre(const livre& w)
{
    ISBN=w.ISBN;
    titre=w.titre;
    prix=w.prix;
    nbAuteurs=w.nbAuteurs;
    for(int i=0; i<w.tab.size(); i++)
    {
        auteur*q=new auteur(*w.tab[i]);
        tab.push_back(q);
    }
}

livre& livre::operator=(const livre& w)
{
    if (this!=&w)
    {
        for(int i=0; i<tab.size(); i++)
            delete tab[i];
        tab.clear();

        ISBN=w.ISBN;
        titre=w.titre;
        prix=w.prix;
        nbAuteurs=w.nbAuteurs;
        for(int i=0; i<w.tab.size(); i++)
        {
            auteur*q=new auteur(*w.tab[i]);
            tab.push_back(q);
        }
    }
    return *this;
}

```

\*\*\*\*\*Question 3\*\*\*\*\*

- 3- La bibliothécaire souhaite classer les livres en deux catégories : Livre d'enfant et Livre d'adulte
- Un livre d'adulte est caractérisé par une catégorie (int) (1:roman, 2:scientifique, 3:humour). Une fonction affiche qui permet d'afficher toutes les caractéristiques du livre d'adulte.
  - La bibliothécaire ne souhaite pas afficher les livres d'enfant par contre elle vous demande d'ajouter un traitement pour qui permet de compter le nombre de livre d'enfant créé.
  - Définir les constructeurs nécessaires pour chaque classe.

```
class livreEnfant: virtual public livre // virtual pour Question 5
{
    static int nbLivreEnfant;
public:
    livreEnfant(int =99, string="P00", float=20.0, int =1);
    void afficher(string = "");
    static void afficherNbLivreEnfant(string = "");
    ~livreEnfant(void);
};

int livreEnfant::nbLivreEnfant=0;
livreEnfant::livreEnfant(int i, string t, float p, int nb): livre(i,t,p,nb)
{
    nbLivreEnfant++;
}
livreEnfant::~~livreEnfant(void)
{
}

void livreEnfant::afficher(string msg)
{ // corps vide car cette méthode est annulée
}

void livreEnfant::afficherNbLivreEnfant(string msg)
{
    cout<<msg<<endl;
    cout<<"\n le nombre de livres pour enfants: "<<nbLivreEnfant<<endl;
}

class livreAdulte: virtual public livre // virtual pour Question 5
{
protected:
    int categorie;
public:
    livreAdulte(int =99, string="P00", float=20.0, int =1, int =1);
    ~livreAdulte(void);
    void afficher(string = "");
};

livreAdulte::livreAdulte(int i, string t, float p, int nb, int c):livre(i,t,p,nb)
{
    categorie=c;
}

livreAdulte::~~livreAdulte(void)
{
}
void livreAdulte::afficher(string msg)
{
    cout<<msg<<endl;
    livre::afficher(); // appel de afficher de la classe livre
    cout<<"le num de categorie est "<<categorie<<endl;
}
```

## \*\*\*\*\*Question 4\*\*\*\*\*

4- Définir une classe Bibliothèque caractérisé par : un ensemble de livre(utiliser vector) - les constructeurs nécessaires – une fonction qui permet d'ajouter n'importe quelle type de livre et une fonction qui permet de supprimer un livre.

```
class bibliotheque
{
    vector<livre*>tab;
public:
    bibliotheque(void);
    ~bibliotheque(void);
    void afficher(string = "");
    bibliotheque(const bibliotheque&);
    bibliotheque& operator= (const bibliotheque&);
    void ajouter(livreEnfant, int =0);
    void ajouter(livreAdulte, int =0);
    void supprimer(int =0);
};

#include "bibliotheque.h"

bibliotheque::bibliotheque(void)
{
    char rep;int choix;
    livre*q;
    do
    {
        cout<<"\n taper 1: livreEnfant et 2 livreAdulte "<<endl;
        cin>>choix;
        if(choix==1) q=new livreEnfant;
        else q=new livreAdulte;
        q->saisir();
        tab.push_back(q);
        cout<<"\n voulez vous rajouter ?"<<endl;
        cin>>rep;
    }
    while(rep=='o' || rep=='O');
}

bibliotheque::~bibliotheque(void)
{
    for(int i=0; i<tab.size(); i++)
        delete tab[i];
    tab.clear();
}
```

```
void bibliotheque::afficher(string msg)
{
    cout<<msg<<endl;
    for(int i=0; i<tab.size(); i++)
        tab[i]->afficher();
}

bibliotheque::bibliotheque (const bibliotheque& w)
{
    livre*q;
    for(int i=0; i<w.tab.size(); i++)
    {
        if(typeid(*w.tab[i])== typeid(livreEnfant))
            q=new livreEnfant(static_cast<livreEnfant&> (*w.tab[i]));

        else if(typeid(*w.tab[i])== typeid(livreAdulte))
            q=new livreAdulte(static_cast<livreAdulte&> (*w.tab[i]));

        tab.push_back(q);
    }
}

bibliotheque& bibliotheque::operator=(const bibliotheque& w)
{
    for(int i=0; i<tab.size(); i++)
        delete tab[i];
    tab.clear();

    livre*q;
    for(int i=0; i<w.tab.size(); i++)
    {
        if(typeid(*w.tab[i])== typeid(livreEnfant))
            q=new livreEnfant(static_cast<livreEnfant&> (*w.tab[i]));

        else if(typeid(*w.tab[i])== typeid(livreAdulte))
            q=new livreAdulte(static_cast<livreAdulte&> (*w.tab[i]));

        tab.push_back(q);
    }
}

void bibliotheque::ajouter(livreEnfant l, int ind)
{
    livre*q=new livreEnfant(l);
    tab.insert(tab.begin()+ind, q);
}

void bibliotheque::ajouter(livreAdulte l, int ind)
{
    livre*q=new livreAdulte(l);
    tab.insert(tab.begin()+ind, q);
}

void bibliotheque::supprimer(int ind)
{
    tab.erase(tab.begin()+ind);
}
```

## \*\*\*\*\*Question 5\*\*\*\*\*

5- Avec l'évolution des technologies la bibliothécaire a fait appelle à une société qui lui à numérisé tout les livres. Elle vous demande donc de définir une classe ebook qui dérive de livre d'enfant et livre d'adulte. Cette classe est en plus caractérisée par une taille(Mo). Définir toutes les méthodes nécessaires pour cette classe et effectuez les modifications nécessaires dans les classes précédentes.

```
class ebook:public livreEnfant, public livreAdulte
{
    int taille;
public:
    ebook(int =99, string="POO", float=20.0, int =1, int =1, int =10);
    ~ebook(void);
    void afficher(string = "");
};
```

```
#include "ebook.h"
```

```
ebook::ebook(int i, string t, float p, int nb, int c, int
tail):livreEnfant(i,t,p,nb),
    livreAdulte(i,t,p,nb,c), livre(i,t,p,nb), taille(tail)
{
}
ebook::~ebook(void)
{
}
void ebook::afficher(string msg)
{
    livreAdulte::afficher();
    cout<<"\n la taille est "<<taille<<endl;
}
```