

# Programmation orienté objet

## Partie 2

### Table des matières

---

Analyse du problème .....	page 1
Exercice 1 .....	page 2 - 3
Présentation du programme .....	page 2
Classe livre .....	page 2
Programme de test .....	page 3
Exercice 2 .....	page 4 - 5
Présentation du programme .....	page 4
Classe client et compte .....	page 4
Programme de test .....	page 5

---

### Analyse du problème :

L'objectif de ce TD, comme pour le précédent, est de réaliser un programme faisant appel à des classes, des objets, des fonctions et des constructeurs ainsi que les méthodes GET et SET afin d'afficher à l'utilisateur les spécificités d'un livre (pour l'exercice 1) et d'un compte bancaire (pour l'exercice 2).

## Exercice 1

L'objectif de ce programme est d'afficher des informations sur un livre, saisies préalablement par l'utilisateur.

On commence par créer la classe **livre** contenant les attributs suivants : le titre, l'auteur et le prix. Grâce à cette classe, on définira un constructeur permettant d'initialiser les attributs de la méthode par des valeurs saisies par l'utilisateur.

```
class livre {  
private:  
    float prix;  
    string auteur, titre;  
public:  
    void setPrix(float lePrix) {  
        prix=lePrix;  
    }  
    float getPrix() {  
        return prix;  
    }  
    void setAuteur(string lauteur) {  
        auteur=lauteur;  
    }  
    string getAuteur() {  
        return auteur;  
    }  
    void setTitre(string leTitre) {  
        titre=leTitre;  
    }  
    string getTitre() {  
        return titre;  
    }  
    void afficher() {  
        cout<<"Le titre est : "<<getTitre()<<endl;  
        cout<<"L'auteur est : "<<getAuteur()<<endl;  
        cout<<"Le prix est : "<<getPrix()<<endl;  
        cout<<"Titre : "<<getTitre()<<" , Auteur : "<<getAuteur()<<" , Prix : "<<getPrix()<<endl;  
    }  
};
```

The diagram illustrates the relationship between the methods of the `livre` class and its attributes. It shows three attribute boxes: `Prix`, `Auteur`, and `Titre`. Lines connect the `set` and `get` methods for each attribute to the attribute box itself. For example, `setPrix` and `getPrix` are connected to `Prix`, `setAuteur` and `getAuteur` are connected to `Auteur`, and `setTitre` and `getTitre` are connected to `Titre`.

La méthode **Afficher()** : permet d'afficher les informations du livre en cours.

On peut maintenant écrire un programme pour tester la classe **livre**.

```
int main() {  
    string saisieString;  
    float saisieFloat;  
    livre livre1;  
    cout<<"Saisir l'auteur : ";  
    getline(cin, saisieString);  
    livre1.setAuteur(saisieString);  
    cout<<"Saisir le titre : ";  
    getline(cin, saisieString);  
    livre1.setTitre(saisieString);  
    cout<<"Saisir le prix : ";  
    cin>>saisieFloat;  
    livre1.setPrix(saisieFloat);  
    livre1.afficher();  
  
    return 0;  
}
```

- ☐ On crée deux variables de saisie (pour l'utilisateur), une récupérant les chaînes de caractères (string) et une autre récupérant les valeurs de type flottantes.
- ☐ On définit un constructeur permettant d'initialiser les attributs de la méthode par des valeurs saisies par l'utilisateur.
- ☐ L'utilisateur saisit les informations sur le livre. La méthode `setAuteur` permet d'attribuer les valeurs à la classe livre les données saisies par l'utilisateur. La fonction `getline` permet à l'utilisateur de saisir des chaînes de caractères comprenant des espaces (pour le nom complet de l'auteur et le titre du livre).
- ☐ Affichage de l'ensemble des informations du livre grâce à la méthode `afficher()`.

Voici un exemple d'exécution du programme :

```
Saisir l'auteur : Hugo MARCEAU  
Saisir le titre : Guide des truites  
Saisir le prix : 10  
Le titre est : Guide des truites  
L'auteur est : Hugo MARCEAU  
Le prix est : 10  
Titre : Guide des truites, Auteur : Hugo MARCEAU, Prix : 10  
Program ended with exit code: 0
```

## Exercice 2

L'objectif de ce programme est d'afficher les détails d'un compte bancaire d'un client. Toutes les informations relatives au client et aux comptes bancaires seront saisies par l'utilisateur. Le programme permettra de également de créditer et de débiter le compte bancaire du client.

Le programme est composé de deux classes : **client** et **compte**.

```
class client {  
private:  
    string cin, nom, prenom, tel;  
public:
```

```
    void setCin(string leCin) {  
        cin=leCin;  
    }  
    string getCin() {  
        return cin;  
    }  
    void setNom(string leNom) {  
        nom=leNom;  
    }  
    string getNom() {  
        return nom;  
    }  
    void setPrenom(string lePrenom) {  
        prenom=lePrenom;  
    }  
    string getPrenom() {  
        return prenom;  
    }  
    void setTel(string leTel) {  
        tel=leTel;  
    }  
    string getTel() {  
        return tel;  
    }  
};
```

```
    void afficher() {  
        cout<<"CIN : "<<getCin()<<endl;  
        cout<<"NOM : "<<getNom()<<endl;  
        cout<<"Prénom : "<<getPrenom()<<endl;  
        cout<<"Tél : "<<getTel()<<endl<<endl;  
    }  
};
```

```
class compte {  
private:  
    int solde, code;  
public:
```

```
    void setSolde(int leSolde) {  
        solde+=leSolde;  
    }  
    void crediter(int a) {  
        solde+=a;  
    }  
    void debiter(int b) {  
        solde-=b;  
    }  
};
```

```
    int getSolde() {  
        return solde;  
    }  
    void setCode(int leCode) {  
        leCode++;  
        code=leCode;  
    }  
    int getCode() {  
        return code;  
    }  
};
```

```
    void afficher() {  
        cout<<"Numéro de compte : "<<getCode()<<endl;  
        cout<<"Solde de compte : "<<getSolde()<<endl;  
    }  
};
```

Méthodes permettant :

- D'attribuer les données saisies par l'utilisateur aux différentes variables de la classe (**set**),
- De récupérer les données des différentes variables de la classe (**get**).

Méthode **créditer()** et **débiter()**

Méthode d'affichage

La partie **main()** du programme va nous permettre d'entrer les informations nécessaires dans les classes, et de tester leur fonctionnement.

```

int main() {
    int saisieInt;
    string saisieString;

    compte comp1;
    client client1;

    comp1.setCode(0);
    comp1.setSolde(0);

    cout<<"Donner le CIN : ";
    cin>>saisieString;
    client1.setCin(saisieString);
    cout<<"Donner le nom : ";
    cin>>saisieString;
    client1.setNom(saisieString);
    cout<<"Donner le prenom : ";
    cin>>saisieString;
    client1.setPrenom(saisieString);
    cout<<"Donner le numéro de téléphone : ";
    cin>>saisieString;
    client1.setTel(saisieString);
    cout<<endl;

    cout<<"Détail du compte : "<<endl<<"*****"<<endl<<endl;
    comp1.afficher();
    cout<<"Propriétaire du compte : "<<endl;
    client1.afficher();

    cout<<"*****"<<endl<<endl;
    cout<<"Donner le montant à déposer : ";
    cin>>saisieInt;
    comp1.crediter(saisieInt);
    cout<<"Opération bien effectuée"<<endl<<endl;

    cout<<"*****"<<endl<<endl;
    comp1.afficher();
    cout<<"Propriétaire du compte : "<<endl;
    client1.afficher();

    cout<<"*****"<<endl<<endl;
    cout<<"Donner le montant à retirer : ";
    cin>>saisieInt;
    comp1.debiter(saisieInt);
    cout<<"Opération bien effectuée"<<endl<<endl;

    cout<<"*****"<<endl<<endl;
    comp1.afficher();
    cout<<"Propriétaire du compte : "<<endl;
    client1.afficher();

    return 0;
}

```

Création de deux constructeurs :

- Un pour les informations relatives au compte bancaire
- Un autre pour les informations relatives aux clients

Initialisation du code et du solde du compte à 0

L'utilisateur saisit les informations sur le client

On affiche les informations grâce aux méthodes **afficher()**

On crédite le compte du client

On débite le compte du client

Voici un exemple d'exécution du programme :

<pre> Donner le CIN : EE111222 Donner le nom : MARCEAU Donner le prenom : Hugo Donner le numéro de téléphone : 0601020304  Détail du compte : *****  Numéro de compte : 1 Solde de compte : 0 Propriétaire du compte : CIN : EE111222 NOM : MARCEAU Prénom : Hugo Tél : 0601020304 </pre>	<pre> ***** Donner le montant à déposer : 2000050000 Opération bien effectuée  *****  Numéro de compte : 1 Solde de compte : 50000 Propriétaire du compte : CIN : EE111222 NOM : MARCEAU Prénom : Hugo Tél : 0601020304 </pre>	<pre> ***** Donner le montant à retirer : 10000 Opération bien effectuée  *****  Numéro de compte : 1 Solde de compte : 40000 Propriétaire du compte : CIN : EE111222 NOM : MARCEAU Prénom : Hugo Tél : 0601020304 </pre>
---	--	---