

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?

Première classe en C++

Le but de ce TP est d'aborder concrètement la notion de classe et d'objets en programmant une première classe Compte bancaire représentée par un solde.

1 Créez un fichier Compte.h

Le fichier Compte.h doit contenir la spécification de la classe Compte. Créez ce fichier au moyen d'un éditeur de textes comme nedit ou xemacs.

- On peut faire un dépôt sur le compte
- On peut faire un retrait sur le compte
- On peut afficher le solde
- On peut virer de l'argent vers un compte destinataire

```
class Compte {  
    //attributs  
  
public:  
    // Constructeur par défaut  
    // Constructeur avec paramètre  
    // depot d'une somme  
    // retrait d'une somme  
    // afficher le solde  
    // virer une somme vers un autre compte  
    //accesseurs  
};
```

2 Créez un fichier Compte.cpp

Le fichier Compte.cpp doit contenir le code des méthodes de la classe Compte. Attention, l'extension du fichier est en .cpp et non pas en .c pour le différencier d'un programme en langage C. Créez-le avec le code du constructeur par défaut comme suit :

```
#include « Compte.h »  
  
// code du constructeur par défaut  
Compte::Compte () {  
    this->solde = 0;  
}
```

3 Créez un fichier de test main.cpp

Le fichier main.cpp doit contenir le code du programme de test. La déclaration du Compte c1 provoque l'appel au constructeur par défaut qui vient d'être défini.

```
#include « Compte.h »

int main() {
// déclaration et création d'un compte
Compte c1 ;
// testez ici les autres fonctions de la classe Compte
}
```

4 Compilation, édition de liens et exécution

La compilation permet de créer un fichier objet `Compte.o`. Sous Unix, lancez la commande suivante : **g++ -c Compte.cpp**

Corrigez les éventuelles erreurs et relancez cette commande de compilation tant qu'il existe des erreurs. Vérifiez la création du fichier objet **Compte.o** dans votre répertoire.

De la même manière, compilez le fichier `main.cpp`. Vous devez obtenir le fichier objet **main.o**.

Quand tous les fichiers ont été compilés (on ne compile pas le fichier `Compte.h`), lancez l'édition de liens pour tous les fichiers : **g++ -o main *.o**

Vérifiez la création du fichier exécutable **main** dans votre répertoire. Vous pouvez alors lancer son exécution. Que se passe-t-il à l'exécution ? Pourquoi ?