

Programmation orientée objet

LIV 2024-2025

TP - Patron de Classe

Site du cours : <https://defelice.up8.site/poo.html>

Les exercices marqués de (©) sont optionnels.

Un fichier écrit en langage C++ se termine conventionnellement par .cpp.

Une commande de compilation est `g++ fichier_source1.cpp fichier_source2.cpp fichier_source3.cpp`.

Voici des options de cette commande.

- `-o nom_sortie` pour donner un nom au fichier de sortie (par défaut `a.out`).
- `-Wall` pour demander au compilateur d'afficher plus d'avertissements (warnings)
- `-Wextra` pour demander au compilateur d'afficher plus d'avertissements (warnings)
- `-std=c++20` pour compiler selon la norme c++20
- `-g -fsanitize=address` pour que l'exécutable puisse detecter certaines erreurs de mémoire (adresse inaccessible, fuite mémoire) (l'inconvenient est que l'exécutable est plus lent, -g pour avoir plus de précision).
- `-g` pour ajouter des informations de débogage, utile avec debug (exécutable plus lent).

Exemple : `g++ -Wall -Wextra fichier1.cpp -o monprogramme`

Dans tous les exercices de ce TP (et des tp suivants) il faut, si possible, penser à déclarer les méthodes constantes. Pensez également à éviter toute erreur de mémoire (fuite de mémoire, dépassement d'indice,...).

Exercice 1. *Somme*

Créer, dans un fichier `sommeGenerique.hpp`, le patron de fonction `template <typename T> T pSomme(int n, T* tab)` qui renvoie la somme des éléments d'un tableau non vide (on suppose que `n>0`) d'objets sommables de type `T` (on suppose que l'argument de `T` possède une surcharge de `T operator+(...)`).

Exemple :

```
#include "sommeGenerique.hpp"
class MaClasse {
private :
    int r;
public :
    MaClasse(int pr): r{pr}{ }
    MaClasse operator + (const MaClasse & p){
        return MaClasse{this->r+p.r};
    }
    int get(void){return r;}
};

int main(){
    int T[5]={1,2,3,4,5};
    MaClasse G[3]={1,5,7};
    pSomme(5,T); // renvoie 1+2+3+4+5
    MaClasse s=pSomme(3,G); // renvoie un objet MaClasse{1+5+7}
    s.get(); // renvoie 13
    return 0;
}
```

Exercice 2. *Tri*

Écrire un patron de fonction `triGeneral` qui tri un tableau d'objets comparables (implantant `operator<`) de type `T`, `T` étant un type paramétré. (Vous pouvez utiliser l'algorithme de tri de votre choix)

Exercice 3. *Couples sommables et comparables*

Créer un patron de classe `Couple` vérifiant les spécificités suivantes :

Si `H` et `T` sont deux noms de types. `Couple couple1<T,H>(a,b)` fabrique un couple de deux objets (de types différents `T,H`). On suppose que les types `T` et `H` implantent (outre le constructeur de copie) ...

- les opérateurs `T operator+(T,T) operator+(H,H)`.
- les opérateurs `int operator<(T,T) int operator<(H,H)`.

Dans la classe (ou dans le patron de classe) implanter :

1. Les méthodes `getFirst` et `getSecond` renvoient respectivement la première et respectivement la seconde composante d'un couple. Simple illustration : `getFirst : (a,b) → a`. `getSecond : (a,b) → b`.
2. L'opérateur `operator+` renvoie un couple en faisant la somme de chacune des deux composantes : $(a,b) + (c,d) = (a+c, b+d)$
3. L'opérateur `operator<` qui compare deux couples : on dira que $(a,b) < (c,d)$ soit si $a < c$ soit si $(a = c \text{ et } b < d)$. Sinon on dira que $(a,b) \geq (c,d)$.

Donner un exemple d'utilisation des patrons de fonction `pSomme` et `triGeneral` sur la classe couple.

Exercice 4. Polynôme

Un polynôme sur deux variables X et Y est une somme de monômes. Exemple : $3X^2Y^2 - 2X^3Y^2 + 4X^7 + 6$. On veut représenter un polynôme par une `map` (de la bibliothèque standard). Les clés de la map seront les monômes présents dans la somme. Les valeurs de la map seront les coefficients de chaque monôme. Par exemple pour le polynôme précédents la map ressemblera à :

- $X^2Y^2 \rightarrow 3$,
- $X^3Y^2 \rightarrow -2$,
- $X^7 \rightarrow 4$,
- $1 \rightarrow 6$.

1. Construire une classe Monôme.
2. Construire une classe Polynôme qui utilisera un objet `map`. Les clés seront les monômes, les valeurs seront les coefficients (de type `double`).
3. Implanter le produit de monômes.
4. Implanter la somme de polynômes.
5. @ Implanter le produit de polynômes.

Exercice 5. Fonction Min

Écrire le patron de la fonction `template <typename T> T min(T const& a, T const& b)` qui renvoie une copie de la valeur minimum de deux objets comparables.

Exercice 6. Exponentiation rapide

Créer un patron de fonction qui prend un objet a de type paramétré T , et un entier n de type `int` et renvoie a^n . On suppose que $n > 0$. Utiliser la méthode de l'exponentiation rapide (à partir d'un opérateur binaire `T operator*`).

Pour rappel, on peut dire que l'exponentiation rapide s'inspire des formules suivantes : $a^n = \left(a^{\frac{n}{2}}\right)^2$ si n est pair. $a^n = a \times \left(a^{\frac{n-1}{2}}\right)^2$ si n est impair.