

TP 4 C++

Soit le programme suivant :

```
void main() {
    CBasePile* ptPile ;
    CPileFifo fifo ; // pile First In First Out
    CPileLifo lifo ; // pile Last In First Out

    ptPile = &fifo; // ptPile pointe vers la pile FIFO
    *ptPile < 0 < 1; // on empile 0 puis 1
    cout << "pile=" << *ptPile ; // à l'écran : pile=0 1

    int i;
    *ptPile > i ; // dépile dans i
    cout << "pile=" << *ptPile << "i=" << i ; // à l'écran : pile=1 i=0
    ptPile = &lifo ; // ptPile pointe vers la pile LIFO
    *ptPile < 0 < 1 ; // on empile 0 puis 1
    cout << "pile=" << *ptPile ; // à l'écran : pile=0 1
    *ptPile > i ; // dépile dans i
    cout << "pile=" << *ptPile << " i=" << i ; // à l'écran : pile=0 i=1
}
```

Le but de cet exercice est de regrouper dans une classe de base, `CBasePile`, les opérations communes aux piles LIFO (Last In First Out) et FIFO (First In First Out), puis de spécialiser cette classe par dérivation en une classe permettant de gérer une pile LIFO, et une classe permettant de gérer une pile FIFO. Pour cela, il faut répondre aux questions suivantes :

- Quelle structure de donnée peut être commune à une pile LIFO et FIFO ? En déduire les données membres de la classe de base.
- D'après le programme ci-dessus quels sont les opérateurs communs aux piles LIFO et FIFO ? En déduire les fonctions membres de la classe de base.
- La classe de base contient les données membres communes à une pile LIFO et FIFO, mais c'est dans les classes dérivées que doivent être spécialisées les fonctions membres. Ecrire alors les classes dérivées `CPileLifo` et `CPileFifo` pour les piles LIFO et FIFO respectivement.
- Comment assurer la ligature dynamique des fonctions pour faire fonctionner le programme ci-dessus (appel des fonctions des classes dérivées via un pointeur sur une classe de base) ?