

## TP 8 : Surcharge d'opérateurs

### 1 Définition de la classe Dictionnaire

Le but de ce TP est d'utiliser la surcharge d'opérateurs en C++ en implémentant un dictionnaire. La structure de dictionnaire utilisée sera simple, et la classe **Dico** sera composée des trois attributs suivants :

- **string\* mots** : un tableau de string qui stockera les mots du dictionnaire
- **string\* defs** : un tableau de string qui stockera les définitions du dictionnaire
- **int capacite** : la taille des deux précédents tableaux
- **int nbMots** : le nombre de mots présents dans le dictionnaire

La chaîne de caractères stockée à l'indice **i** du tableau **defs** sera la définition du mot stocké à l'indice **i** du tableau **mots**.

#### Testez votre code au fur et à mesure !

La classe comportera les méthodes suivantes :

- **Dico()** : le constructeur par défaut, il initialise les attributs (en réservant notamment la mémoire pour les tableaux), en fixant une capacité par défaut de votre choix, par exemple 10.
- **Dico(int capacite)** : le même constructeur que précédemment, mais avec la capacité de spécifiée cette fois-ci.
- **~Dico()** : le destructeur devra détruire les deux tableaux **mots** et **defs** (pour détruire un tableau dynamique en C++, il faut utiliser *delete[ ]*).
- **int getNbMots()** : qui renvoie le nombre de mots présents dans le dictionnaire.
- **int indexOf(string mot)** : qui renvoie l'indice du mot **mot** dans le tableau **mots** s'il est présent, et -1 sinon.

- **void add(string mot, string def)** : qui ajoute une nouvelle entrée dans le dictionnaire si le mot n'était pas présent, et qui remplace la précédente définition si le mot était déjà présent. Dans un premier temps, supposez que le nombre de mots n'atteint jamais la capacité maximale. Dans second temps, prenez cette éventualité en compte en créant de plus gros tableaux le cas échéant.
- **string get(string mot)** : retourne la définition associée au mot **mot** si celui-ci est présent, et une chaîne vide si celui-ci n'est pas présent.
- **void addAll(Dico& d)** : qui ajoute au dictionnaire courant tous les mots et définitions du dictionnaire **d**.

## 2 Surcharge des opérateurs

On souhaite enfin surcharger certains opérateurs afin que le main décrit à la page suivante fonctionne. Il est fortement conseillé d'utiliser les fonctions que vous avez déjà écrites.

Indication : les surcharges d'opérateurs sont utilisées lignes 11, 12, 16, 24, 28, 35 et 39 du main.

## 3 Bonus

Creez une classe *DicoOrdonne* qui hérite de la classe *Dico*, et qui stockera les entrées par ordre alphabétique. Ainsi, lors la recherche d'une définition, on fera une recherche par dichotomie.

## 4 Main de test

```
1 int main(int argc, char* argv[]) {
2
3     Dico d1(3);
4     d1.add("mot1", "def1");
5     d1.add("mot2", "def2");
6     d1.add("mot3", "def3");
7     d1.add("mot4", "def4");
8     d1.add("mot2", "nouvelle_def2");
9
10    cout << "_____ " << endl;
11    cout << "Definition de mot2 : " << d1["mot2"] << endl;
12    cout << "Definition de mot2321 : " << d1["mot2321"] << endl;
13
14    cout << "_____ " << endl;
15    cout << "Affichage du dictionnaire d1 : " << endl;
16    cout << d1 << endl;
17
18    Dico d2(3);
19    d2.add("mot5", "def5");
20    d2.add("mot6", "def6");
21    d2.add("mot7", "def7");
22    d2.add("mot8", "def8");
23
24    d2 += d1;
25
26    cout << "_____ " << endl;
27    cout << "Affichage du dictionnaire d2 : " << endl;
28    cout << d2 << endl;
29
30    Dico d3(3);
31    d3.add("mot9", "def9");
32    d3.add("mot10", "def10");
33    d3.add("mot11", "def11");
34
35    Dico d4 = d3 + d2;
36
37    cout << "_____ " << endl;
38    cout << "Affichage du dictionnaire d4 : " << endl;
39    cout << d4 << endl;
40
41    return 0;
42 }
```

## 5 Résultat attendu du main précédent

```
1
2 Definition de mot2 : nouvelle_def2
3 Definition de mot2321 :
4
5 Affichage du dictionnaire d1 :
6 Taille du dico : 4
7 1 : mot1 : def1
8 2 : mot2 : nouvelle_def2
9 3 : mot3 : def3
10 4 : mot4 : def4
11
12
13 Affichage du dictionnaire d2 :
14 Taille du dico : 8
15 1 : mot5 : def5
16 2 : mot6 : def6
17 3 : mot7 : def7
18 4 : mot8 : def8
19 5 : mot1 : def1
20 6 : mot2 : nouvelle_def2
21 7 : mot3 : def3
22 8 : mot4 : def4
23
24
25 Affichage du dictionnaire d4 :
26 Taille du dico : 11
27 1 : mot9 : def9
28 2 : mot10 : def10
29 3 : mot11 : def11
30 4 : mot5 : def5
31 5 : mot6 : def6
32 6 : mot7 : def7
33 7 : mot8 : def8
34 8 : mot1 : def1
35 9 : mot2 : nouvelle_def2
36 10 : mot3 : def3
37 11 : mot4 : def4
```