

Responsable pédagogique	AF	AM	OP	PM	[CPP1] CNotes
Période	Sem1	Sem2	Sem3	Sem4	
Volume horaire	Cours/TD		TP		
			6		

Indicateur temporel (hors rédaction du compte-rendu) :

questions	1h	2h	3h	4h	5h	6h
1.. 5						
6, 7						
8, 9						
10 .. 13						
14						

Documents à rendre : Compte-rendu contenant à minima les sources (avec entête standard) et les résultats obtenus.

Modèle de base de la classe CNotes

Un objet de classe `CNotes` est une liste chaînée de maillons de classe `CElem` dont chacun contient une note sur 20. Un objet `CNotes` maintient des pointeurs de maillon sur le début et la fin de la chaîne, et le nombre de maillons créés.

1. Étudier le code fourni en annexe.
2. Créer le fichier d'interface `cnotes.h` avec le code donné en annexe 1.
3. Créer le fichier d'implémentation `cnotes.cpp` avec le code donné en annexe 2.
4. Implémenter la méthode de calcul de moyenne dont le prototype apparaît dans l'interface.
5. Tester la classe avec le programme de test `main.cpp` suivant :

```
#include "cnotes.h"

void affiche(const string& m, const CNotes& c )
{
    cout << m << " : " << c.nbNotes() << " note" ;
    if ( c.nbNotes() > 1 ) cout << 's' ;
    cout << ", moy. = " << c.Moyenne() << endl ;
}

int main(void)
{
    CNotes math ; // constructeur par défaut
    math.Ajoute(12) ; // méthode Ajoute()
    math.Ajoute(10) ;
    affiche("math ", math ) ;
}
```

Extension des moyens de saisie de notes...

6. Créer une surcharge de l'opérateur += afin de pouvoir ajouter une note à la chaîne.
7. Tester la classe avec le programme de test suivant :

```
CNotes physique(7) ;           // constructeur avec argument
physique += 13 ;               // opérateur += avec arg. double
physique += 10 ;
affiche("physique", physique ) ;
```

8. Surcharger l'opérateur [] de manière à proposer un accès direct en lecture et en écriture sur une note de la liste. Si l'indice demandé est incorrect, la méthode doit créer les maillons manquants avec des notes nulles...
9. Développer les autres méthodes satisfaisant le code de test suivant :

```
CNotes anglais ;
anglais[0] = 11 ;              // opérateur d'accès direct []
anglais[1] = 15 ;
anglais = anglais + 13 ;       // opérateur + avec arg. double
                                // op. = sur même objet CNotes

CNotes francais(12) ;
francais[2] = 14 ;            // devoir d'indice 1 non rendu !
affiche("anglais ", anglais ) ;
affiche("francais", francais ) ;
```

Manipulations de listes complètes...

10. Créer une surcharge de la méthode Ajoute() afin d'accepter en argument le contenu d'un autre objet de classe CNotes : void Ajoute(const CNotes& cn) ;

11. Implémenter un constructeur de copie (utiliser la méthode précédente) :
CNotes(const CNotes& cn) ;
12. Ajouter une surcharge de l'opérateur += acceptant un objet CNotes en opérande.
CNotes& operator +=(const CNotes& cn) ;
13. Valider les méthodes en ajoutant le code de test suivant :

```
CNotes langues = francais ;    // constructeur de copie
langues += anglais ;          // opérateur += avec arg. CNotes
affiche("langues ", langues ) ;
```

14. Satisfaire les écritures suivantes...

```
CNotes bilan ;
bilan = math + physique ;     // opérateur + avec arg. CNotes
                                // op. = avec autre objet CNotes

bilan += anglais ;            // opérateur += avec arg. CNotes
bilan += francais ;
affiche("bilan ", bilan ) ;
```

Annexe 1

```
#ifndef CNOTES_H
#define CNOTES_H

#include <iostream>

using namespace std ;

class CElem ;

typedef CElem* pElem ;

class CElem {
public :
    double note ;
    pElem suiv ;
} ;

class CNotes {
    pElem prem , dern ;
    int      nbElem ;

public :
    CNotes(double n = -1 )
        : prem(NULL) , dern(NULL) , nbElem(0) {
        if ( n != -1 )      Ajoute(n) ;
    }
    ~CNotes() ;

    int nbNotes() const { return nbElem ; }
    void Ajoute(double v ) ;
    double Moyenne() const ;
} ;

#endif
```

*affichage normalement produit
par le code de test...*

```
math      : 2 notes, moy. = 11
physique  : 3 notes, moy. = 10
anglais   : 3 notes, moy. = 13
français  : 3 notes, moy. = 8.66667
langues   : 6 notes, moy. = 10.8333
bilan     : 11 notes, moy. = 10.6364
```

Annexe 2

```
#include "cnotes.h"

CNotes::~CNotes()
{
    pElem p = prem ;
    while ( prem != NULL ) {
        p = p->suiv ; delete prem ; prem = p ;
    }
    prem = dern = NULL ;
    nbElem = 0 ;
}

void CNotes::Ajoute(double v )
{
    pElem p = new CElem ;
    p->note = v ; p->suiv = NULL ;
    if ( prem == NULL )      prem = dern = p ;
    else {
        dern->suiv = p ; dern = dern->suiv ;
    }
    nbElem++ ;
}
```

