Algorithmique et Programmation C++

TD 4

1. Structures, unions, énumérations

```
/* TD4 - Exercice 1 - STRUCTURES UNIONS ENUMERATIONS *\
\* Par Gael Even et Jeremie Osmont                   */


#include <iostream.h>

// Definition des structures de stockage
enum Mois {janvier,fevrier,mars,avril,mai,juin,juillet,
           aout,septembre,octobre,novembre,decembre};

enum Sexe {masculin,feminin};

enum Sport {football,escalade,course};

struct Date
{
        int j;
        Mois m;
        int a;
};

struct Etudiant
{
         char nom[20];
         char prenom[20];
         Date naissance;
        Sexe se;
        Sport s;
        union
        {
                char coep[3][10];
                char lieu[20];
                int dist;
        };
};



Etudiant l[100];


// Affichage des donnees
void affiche(int i)
{
        cout << "\n\nFICHE ETUDIANT #"<<i+1<<" : ";
        if (l[i].se == masculin) cout << "M. "; else cout <<"Mlle ";
        cout << l[i].nom << " "<< l[i].prenom;
        cout << " (" << l[i].naissance.j << "/"<<
static_cast<int>(l[i].naissance.m)+1
             << "/" << l[i].naissance.a << ")";
        cout << "\nSport pratique : ";
        int j;
        switch(l[i].s)
        {
          case football :
                cout << "FOOT\nCoequipiers : ";
                for (j=0; j<3; j++)
                        cout << l[i].coep[j] << ";";
                break;
          case escalade :
                  cout << "ESCALADE\nLieu : " << l[i].lieu;
                break;
          case course :
                  cout << "COURSE\nDistance : " << l[i].dist << " m";
        }
}

// Enregistrement des donnees :
int main()
{
        cout << "\n\nSERVICE DE SCOLARITE : Entree de la liste des
etudiants :\n";
        cout << "Combien d'etudiants dans la promo ? ";
        int n;
        cin >> n;
        for(int i=0;i<n;i++)
        {
                cout << "----------------------------------------";
                cout<<"\nEtudiant "<<i+1<<endl;
                cout<<"Entrez votre nom : ";
                cin>>l[i].nom;
                cout<<"Entrez votre prenom : ";
                cin>>l[i].prenom;

                cout<<"Entrez votre sexe (M/F) : ";
                char ent=' ';
                while(ent!='M' && ent!='F') cin>>ent;
                if(ent=='M')l[i].se=masculin;
                else l[i].se=feminin;

                cout<<"Entrez votre annee de naissance : ";
                cin>>l[i].naissance.a;
                cout<<"Entrez votre mois de naissance : ";
                int val=0;
                while(val <= 0 || val >= 13) cin>> val;
                l[i].naissance.m=static_cast<Mois>(val-1);
                cout<<"Entrez votre jour de naissance : ";
                cin>>l[i].naissance.j;

                cout << "Entrez votre sport parmi (F)ootball,
(E)scalade, ou (C)ourse : ";
                char ent2=' ';
                while(ent2 !='E' && ent2!='F' && ent2!='C')
cin>>ent2;
                int j;
                switch(ent2)
                {
                  case 'E':
                        l[i].s=escalade;
                        cout << "Lieu d'escalade : ";
                        cin >> l[i].lieu;
                        break;
                  case 'F':
                          l[i].s=football;
                        for(j=0; j<3; j++)
                        {
                                cout << "Equipier #"<<j+1<< " : ";
                                cin >> l[i].coep[j];
                        }
                        break;
                  case 'C':
                          l[i].s=course;
                        cout << "Distance parcourue (en m) : ";
                        cin >> l[i].dist;
                }
        }
        for (int k=0; k<=30; k++) cout << "\n";
        cout << "AFFICHAGE DE LA LISTE DES ETUDIANTS DE LA PROMO,
SOIT "<< n << " ETUDIANTS";
        cout << "\n----------------------------------------";
        for (i=0; i<n; i++) affiche(i);
        cout << "\n\n";
}
```

2. Classes, constructeurs, fonctions membres

```
/* TD4 - Exercice 2 - CLASSE NIVEAU 1   *\
\* Par Jeremie Osmont                   */


#include <iostream.h>

// Definition des trois classes CPoint, CSegment et CCercle :
class CPoint
{
        // Variables membres :
        float x;
        float y;

    public:

        // Constructeurs :
        CPoint() {x=y=0;}
        CPoint(float a,float b=0) {x=a;y=b;}

        // Fonctions permettant de lire et d'ecrire x et y :
        float getx() {return x;}
        float gety() {return y;}
        void writex(float a) {x=a;}
        void writey(float b) {y=b;}

        // Affichage du point :
        void imp() const {cout << '(' << x << ',' << y << ')';}

        // Surdefinition de l'operateur += :
        void operator+=(CPoint p) {x+=p.x; y+=p.y;}
};


class CSegment
{
        // Variables membres :
        CPoint org;
        CPoint fin;

    public:

        // Constructeur :
        CSegment(CPoint a, CPoint b) {org=a; fin=b;}

        // Affichage des proprietes du segment :
        void dessine() const
        {
                cout << "Segment de "; org.imp();
                cout << " a "; fin.imp();
        }

        // Deplacement du segment par ajout d'un point :
        void deplace(CPoint p) {org+=p; fin+=p;}
};

class CCercle
{
        // Variables membres :
        CPoint org;
        float r;

    public:

        // Constructeur :
        CCercle(CPoint o, float f=0) {org=o; r=f;}

        // Affichage des proprietes du cercle :
        void dessine() const
        {
                cout << "Cercle de centre "; org.imp();
                cout << " et de rayon " << r;
        }

        // Deplacement du cercle par ajout d'un point :
        void deplace(CPoint p) {org+=p;}
};


// Programme de test de la classe :
void setpoint(CPoint & p)
{
        float x,y;
        cout << "x="; cin >> x;
        cout << "y="; cin >> y;
        p.writex(x);
        p.writey(y);
}


int main()
{
        cout << "\nDESSINEZ AVEC C++ !";
        int cx=-1;
        cout << "\n---------------------------------------"
             << "\n   0 - Quitter"
             << "\n   1 - Dessiner un point"
             << "\n   2 - Dessiner un segment"
             << "\n   3 - Dessiner un cercle"
             << "\n---------------------------------------";

        while(cx < 0 || cx > 3)
        {
                cout << "\nEntrez votre choix : ";
                cin >> cx;
        }
        cout << "---------------------------------------\n";

        switch(cx)
        {
        case 0:
                cout << "\nAu revoir.";
                break;
        case 1:
                {cout << "Entrez les coordonnees du point :\n";
                CPoint p;
                setpoint(p);
                p.imp();
                break;}
        case 2:
                {CPoint p1,p2,p3;
                cout << "Entrez les coordonnees de l'origine :\n";
                setpoint(p1);
                cout << "Entrez les coordonnees de l'autre point
:\n";
                setpoint(p2);
                CSegment s(p1,p2);
                cout << "\n>> ENREGISTREMENT... "; s.dessine();
                cout << "\n\nEntrez un troisieme point pour deplacer
ce segment :\n";
                setpoint(p3);
                s.deplace(p3);
                cout << "\n>> TRANSLATION...    "; s.dessine();
                break;}
        case 3:
                {CPoint p,p3;
                cout << "Entrez les coordonnees du centre :\n";
                setpoint(p);
                cout << "Entrez le rayon du cercle :\nr=";
                float r;
                cin >> r;
                CCercle c(p,r);
                cout << "\n>> ENREGISTREMENT... "; c.dessine();
                cout << "\n\nEntrez un autre point pour deplacer ce
cercle :\n";
                setpoint(p3);
                c.deplace(p3);
                cout << "\n>> TRANSLATION...    "; c.dessine();
                break;}
        }
        cout << "\n\n";
}
```

3. Classes, constructeurs, tableaux dynamiques, surdéfinition

```
/* TD4 - Exercice 3 - CLASSE NIVEAU 2   *\
\* Par Gael Even & Jeremie Osmont       */


#include <iostream.h>
#include <iomanip.h>
#include <stdlib.h>

// Definition de la classe Entiers :
class Entiers
{
        int * tab;
        int taille;
        int nombre;
        public :
                Entiers(int n=0)
                {
                        nombre=0;
                        taille=n;
                        tab=new int[n];
                }

                Entiers(const Entiers &e)
                {
                        taille=e.taille;
                        tab=new int[taille];
                        nombre=e.nombre;
                        for(int i=0;i<nombre;i++)
                                tab[i]=e.tab[i];
                }

                ~Entiers() {delete [] tab;}

                int cardinal() {return nombre;}

                bool appartient(int n)
                {
                        int i=0;
                        while(tab[i]<n && i<nombre-1)
                                i++;
                        if(tab[i]==n)
                                return true;
                        else
                                return false;
                }

                void inserer(int n)
                {
                        if(nombre<taille)
                        {
                                int i=0;
                                while(tab[i]<n&&i<nombre-1)
                                        i++;
                                if(tab[i]>n)
                                {
                                        for(int j=nombre;j>i;j--)
                                                tab[j]=tab[j-1];
                                        tab[i]=n;
                                        nombre++;
                                }
                                else if(tab[i]<n)
                                {
                                        tab[nombre]=n;
                                        nombre++;
                                }
                        }
                }

                Entiers &operator=(const Entiers &e)
                {
                        if(this!=&e)
                        {
                                delete tab;
                                nombre=e.nombre;
                                taille=e.taille;
                                tab=new int[e.taille];
                                for(int i=0;i<nombre;i++)
                                        tab[i]=e.tab[i];
                        }
                        return *this;
                }

                void operator<(int n) {inserer(n);}

                bool operator[](int n) {return appartient(n);}

                void operator-(int n)
                {
                        int i=0;
                        while(tab[i]<n && i<nombre-1)
                                i++;
                        if(tab[i]==n)
                        {
                                for(int j=i+1;j<=nombre-1;j++)
                                        tab[j-1]=tab[j];
                                nombre--;
                        }
                        else
                                cout<<"absent";
                }

                Entiers operator+(const Entiers &e)
                {
                        Entiers res;
                        res.taille = taille + e.taille;
                        res.tab = new int[res.taille];
                        res.nombre = nombre;
                        for(int i=0;i<nombre;i++)
                                res.tab[i]=tab[i];
                        for(int j=0;j<e.nombre;j++)
                                res.inserer(e.tab[j]);
                        return res;
                }

                void affiche()
                {
                        for(int i=0;i<nombre;i++)
                                cout << setw(4) << tab[i];
                        cout << "\n";
                }
};


// Generateur de nombres pseudo-aleatoires
long NbrAleat(long max)
{
        long a=rand();
        while (a>10000)
                a = a - 10000;
        return static_cast<long>(max*a/10000);
}



// Programme d'interfacage de la Classe
int main(){
        cout << "\nPROGRAMME DE TEST DE LA CLASSE ENTIERS";
        cout << "\n=====================================";
        cout << "\n\nEntrez deux tailles pour generer deux ensembles
Entiers :\n";
        int n1,n2,x;
        cout << "a="; cin >> n1;
        cout << "b="; cin >> n2;
        Entiers tab1(n1),tab2(n2);
        cout << "\nRemplissage du tableau #1 soit " << n1 << "
valeurs :\n";
        for(int i=0;i<n1;i++)
        {
                x = NbrAleat(n1);
                cout << setw(4) << x;
                tab1 < x;
        }
        cout << "\nRemplissage du tableau #2 soit " << n2 << "
valeurs :\n";
        for(int j=0;j<n2;j++)
        {
                x = NbrAleat(n2);
                cout << setw(4) << x;
                tab2 < NbrAleat(n2);
        }
        cout << "\n\nAFFICHAGE DES RESULTATS";
        cout << "\n====================================\n\n";
        cout << "Tableau #1 de " << tab1.cardinal() << " valeurs
enregistrees :\n";
        tab1.affiche();
        cout << "Tableau #2 de " << tab2.cardinal() << " valeurs
enregistrees :\n";
        tab2.affiche();
        Entiers tab3=tab1+tab2;
        cout << "Fusion de #1 et #2, soit " << tab3.cardinal() << "
valeurs stockees :\n";
        tab3.affiche();
        cout << "\n\n";
}
```