

```
// exam2006.C

#include <iostream>
#include <string.h>
using namespace std;

class Temps
{
public :
    Temps ();
    Temps (int h, int m);

    int getHeu() const      { return heu; }
    int getMin() const      { return min; }
    void setHeu (int h)     { heu = h; }
    void setMin (int m)     { min = m; }

    friend ostream& operator<< (ostream& flout, const Temps& t);

    bool operator<= (const Temps& t2) const;

    Temps operator+ (const Temps& t2) const;
    Temps operator- (const Temps& t2) const;

private :
    int heu, min;
};

Temps::Temps ()
{
    heu = 0;
    min = 0;
}

Temps::Temps (int h, int m)
{
    heu = h;
    min = m;
}

ostream& operator<< (ostream& flout, const Temps& t)
{
    flout << t.heu << "h";
    if (t.min < 10)
        flout << "0";
    flout << t.min;

    return flout;
}

bool Temps::operator<= (const Temps& t2) const
{
    return (heu < t2.heu) || (heu == t2.heu && min <= t2.min);
}

Temps Temps::operator+ (const Temps& t2) const
{
    Temps result;

    result.heu = heu + t2.heu + (min + t2.min)/60;
    result.min = (min + t2.min)%60;

    return result;
}

Temps Temps::operator- (const Temps& t2) const
{
    Temps result;

    if (t2.min <= min)
    {
        result.heu = heu - t2.heu;
    }
}
```

```

        result.min = min - t2.min;
    }
    else
    {
        result.heu = heu - t2.heu - 1;
        result.min = (60 - t2.min) + min;
    }

    return result;
}

class IntervalleTemps
{
public :
    IntervalleTemps (int hdeb, int mdeb, int hfin, int mfin);

    Temps getDebut() const { return debut; }
    Temps getFin() const { return fin; }
    void setDebut (int h, int m);
    void setFin (int h, int m);

    void afficher () const;

    bool contient (const Temps& t) const;

private :
    Temps debut, fin;
};

// dans la liste d'initialisation, appel des constructeurs des objets membre
// debut et fin
IntervalleTemps::IntervalleTemps (int hdeb, int mdeb, int hfin, int mfin)
    : debut(hdeb,mdeb), fin(hfin,mfin)
{
}

void IntervalleTemps::setDebut (int h, int m)
{
    debut.setHeu (h);
    debut.setMin (m);
}

void IntervalleTemps::setFin (int h, int m)
{
    fin.setHeu (h);
    fin.setMin (m);
}

void IntervalleTemps::afficher () const
{
    cout << debut << "-" << fin << endl;
}

bool IntervalleTemps::contient (const Temps& t) const
{
    return debut <= t && t <= fin;
}

class TachePlanifiee : public IntervalleTemps
{
public :
    TachePlanifiee (const char* ti, int hdeb, int mdeb, int hfin, int mfin);

    void afficher () const;

    bool plageCommune (const TachePlanifiee& tache2) const;

    void changerHoraire (int hdeb, int mdeb);
    void mettreAvant (const TachePlanifiee& tache2);
    void mettreAprès (const TachePlanifiee& tache2);

private :

```

```

    char titre[50];
};

// dans la liste d'initialisation, appel du constructeur de la classe mère
TachePlanifiee::TachePlanifiee (const char* ti, int hdeb, int mdeb, int hfin,
                                int mfin)
    : IntervalleTemps(hdeb,mdeb,hfin,mfin)
{
    strcpy (titre, ti);
}

void TachePlanifiee::afficher () const
{
    IntervalleTemps::afficher();
    cout << "titre : " << titre << endl;
}

bool TachePlanifiee::plageCommune (const TachePlanifiee& tache2) const
{
    // la tâche a une plage commune avec la 2ème tâche si elle contient le
    // debut ou la fin de la 2ème tâche
    return contient(tache2.getDebut()) || contient(tache2.getFin());
}

void TachePlanifiee::changerHoraire (int hdeb, int mdeb)
{
    Temps duree = getFin() - getDebut();

    setDebut (hdeb, mdeb);

    Temps nouvelleFin = getDebut() + duree;
    setFin (nouvelleFin.getHeu(), nouvelleFin.getMin());
}

void TachePlanifiee::mettreAvant (const TachePlanifiee& tache2)
{
    Temps duree = getFin() - getDebut();
    Temps nouveauDebut = tache2.getDebut() - duree;

    changerHoraire (nouveauDebut.getHeu(), nouveauDebut.getMin());
}

void TachePlanifiee::mettreApres (const TachePlanifiee& tache2)
{
    Temps nouveauDebut = tache2.getFin();

    changerHoraire (nouveauDebut.getHeu(), nouveauDebut.getMin());
}

int main ()
{
    // test classe Temps
    /*
    int h1,m1,h2,m2;
    cout << "saisir 2 Temps : ";
    cin >> h1 >> m1 >> h2 >> m2;
    Temps tp1(h1,m1), tp2(h2,m2);
    if (tp1 <= tp2) cout << tp1 << " <= " << tp2 << endl;
    if (tp2 <= tp1) cout << tp2 << " <= " << tp1 << endl;
    cout << tp1 + tp2 << endl;
    cout << tp1 - tp2 << endl;
    */

    // test classe IntervalleTemps
    /*
    IntervalleTemps it (8,30,12,0);
    it.setDebut (9,15);
    it.setFin (12,20);
    it.afficher();
    Temps t3(9,0), t4(10,0);
    if (!it.contient(t3) && it.contient(t4))
        cout << "contient OK" << endl;
    */
}

```

```
*/  
  
// test classe TachePlanifiee  
TachePlanifiee ta1 ("cours",14,0,17,0), ta2("examen",16,0,18,0),  
    ta3("projet",10,0,12,0);  
ta1.afficher();  
if (ta1.plageCommune(ta2) && !ta1.plageCommune(ta3))  
    cout << "plageCommune OK" << endl;  
ta1.changerHoraire (13,0);  
ta2.mettreAvant (ta1);  
ta3.mettreApres (ta1);  
ta1.afficher();  
ta2.afficher();  
ta3.afficher();  
  
return 0;  
}
```