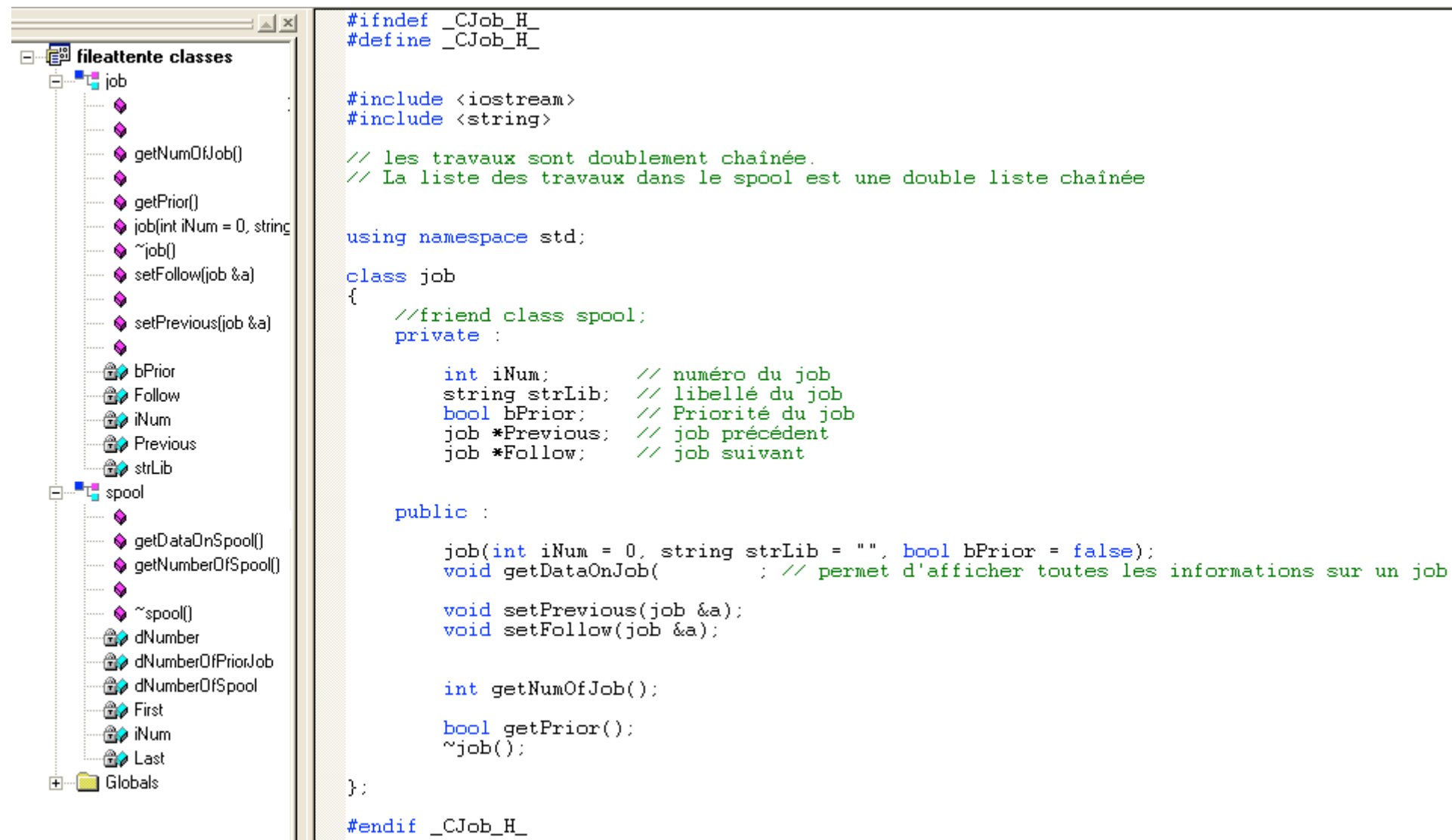


Interface de la classe job : job.h



```
#ifndef _CJob_H_
#define _CJob_H_

#include <iostream>
#include <string>

// les travaux sont doublement chaînée.
// La liste des travaux dans le spool est une double liste chaînée

using namespace std;

class job
{
    //friend class spool;
private :

    int iNum;           // numéro du job
    string strLib;      // libellé du job
    bool bPrior;        // Priorité du job
    job *Previous;      // job précédent
    job *Follow;        // job suivant

public :

    job(int iNum = 0, string strLib = "", bool bPrior = false);
    void getDataOnJob(      ; // permet d'afficher toutes les informations sur un job

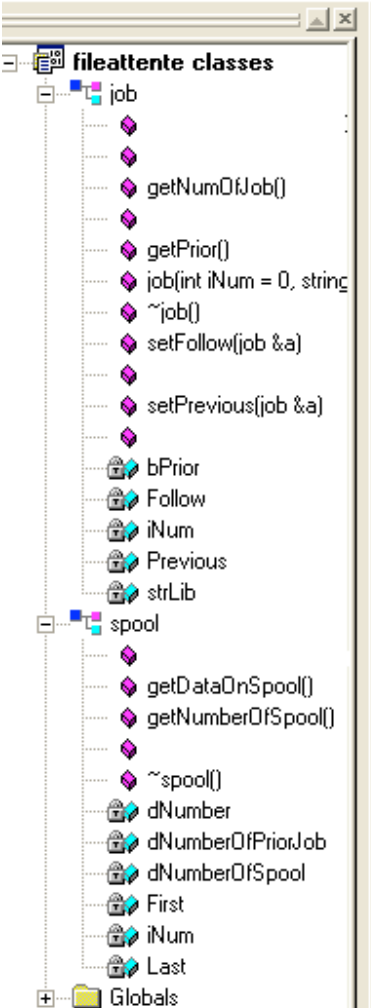
    void setPrevious(job &a);
    void setFollow(job &a);

    int getNumOfJob();

    bool getPrior();
    ~job();
};

#endif _CJob_H_
```

Interface de la classe spool : spool.h



The UML diagram shows two classes: **job** and **spool**. The **job** class has attributes `bPrior`, `Follow`, `iNum`, `Previous`, and `strLib`. Its methods include `getNumOfJob()`, `getPrior()`, `job(int iNum = 0, string ~job())`, `setFollow(job &a)`, and `setPrevious(job &a)`. The **spool** class has attributes `dNumber`, `dNumberOfPriorJob`, `dNumberOfSpool`, `First`, `iNum`, and `Last`. Its methods include `getDataOnSpool()`, `getNumberOfSpool()`, `~spool()`, and `~spool();`. A **Globals** folder is also shown at the bottom.

```

#ifndef _CSpool_H
#define _CSpool_H

#include "job.h"

// le spool fonctionne selon la méthode FIFO pour tous les travaux prioritaires ou non
// Un travail est prioritaire ou il ne l'est pas.
// L'ordre des travaux dans le spool fait apparaître les travaux prioritaires en premier.
// Si plusieurs jobs sont prioritaires ils se retrouvent en tête du spool avec comme ordre, leur ordre d'arrivée
// Le spool connaît en permanence le premier travail et le dernier de la liste
// L'affichage des informations du spool provoque l'affichage de la liste des travaux du spool dans l'ordre
// croissant et décroissant. De plus apparaissent les informations propres à chaque travaux.
// Afin de simplifier le listage des jobs l'appel des affichages successifs sera RECURSIF.

class spool
{
    static double dNumberOfSpool;    // nombre de spool créés, nombre d'occurences de la classe spool

private :
    int iNum;                        // Numéro du spool
    double dNumber;                  // nombre de job dans la liste
    double dNumberOfPriorJob;        // nombre de job prioritaire
    job *First;                      // Premier job de la liste
    job *Last;                       // Dernier job de la liste

public :

    static void getNumberOfSpool(); // fonction membre privée statique
    void getDataOnSpool();          // affiche les informations du spool ainsi que la liste des travaux
                                    // ajoute un job dans le spool

    ~spool();

};

#endif _CSpool_H

```

Le programme appelant : theApp.cpp

