Interface de la classe job : job.h

```
#ifndef _CJob_H_
                     ⊥ ×
                             #define _CJob_H_
Fileattente classes
  Ė....⁼<mark>"</mark> iob
                             #include <iostream>
                             #include <string>
        getNumOfJob()
                             // les travaux sont doublement chaînée.
                             // La liste des travaux dans le spool est une double liste chaînée
        getPrior()
        job(int iNum = 0, strind
                             using namespace std;
        ~job()
        setFollow(job &a)
                             class job
                                  //friend class spool:
        setPrevious(job &a)
                                  private :
        ⊕ bPrior
                                      int iNum;
                                                       // numéro du job
                                      string strLib; // libellé du job
       😭 Follow
                                                       // Priorité du job
                                      bool bPrior:
       🔐 iNum
                                      job *Previous; // job précédent
       Previous
                                      job *Follow: // job suivant
       ∰o strLib
  Ė....•□ spool
                                  public :
        getDataOnSpool()
                                      job(int iNum = 0, string strLib = "", bool bPrior = false);
        getNumberOfSpool()
                                                                 ; // permet d'afficher toutes les informations sur un job
                                      void getDataOnJob(
                                      void setPrevious(job &a);
        ~spool()
                                      void setFollow(job &a);
       -@p dNumber
       dNumberOfPriorJob
       - 😭 dNumberOfSpool
                                      int getNumOfJob();
       -∰a First
                                      bool getPrior();
       🚳 iNum
                                      ~iob():
       🔐 Last
     🛅 Globals
                             };
                             #endif _CJob_H_
```

Interface de la classe spool : spool.h

```
#ifndef CSpool H
                           #define CSpool H
🖃 📴 fileattente classes
    doi 💤 .
                            #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.
       aetNumOfJob()
                           // 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
       getPrior()
                           // L'affichage des informations du spool provoque l'affichage de la liste des travaux du spool dans l'ordre
       iob(int iNum = 0, strind)
                            // croissant et décroissant. De plus apparaîssent les informations propres à chaque travaux.

◇ ~iob∩
                            // Afin de simplifier le listage des jobs l'appel des affichages successifs sera RECURSIF.
       setFollow(job &a).
                           class spool
       setPrevious(job &a).
                                static double dNumberOfSpool; // nombre de spool créés, nombre d'occurences de la classe spool
      ⊕ bPrior
                                private :
      😭 Follow
                                    int iNum:
                                                                 // Numéro du spool
      🔐 iNum
                                    double dNumber:
                                                                 // nombre de job dans la liste
      Previous
                                    double dNumberOfPriorJob; // nombre de job prioritaire
      📸 strLib
                                    iob *First:
                                                                 // Premier job de la liste
    -ta spool
                                    iob *Last;
                                                                 // Dernier job de la liste
                                public :
       getDataOnSpool()
       getNumberOfSpool()
                                    static void getNumberOfSpool(); // fonction membre privée statique
                                    void getDataOnSpool();
                                                                 // affiche les informations du spool ainsi que la liste des travaux
       ~spool()
                                                                  //ajoute un job dans le spool
      📸 dNumber
      dNumberOfPriorJob

⊕ dNumberOfSpool

                                    ~spool():
      📸 First
      🔐 iNum
       🔐 Last
                            #endif _CSpool_H_
     Globals
```

Le programme appelant : theApp.cpp

```
#include "job.h"
                              #include "spool.h"
■ Workspace 'fileattente': 1 project(:
in fileattente files
                             // Les lignes qui commencent par >< font référence à une suppression d'objet de la MC
                             // Les lignes qui commencent par (i) affichent des informations sur un objet
  - Source Files
                             // Les lignes qui commencent par *** affiche des informations sur les occurences d'une classe
        🔝 job.cpp
        🔝 spool.cpp
                              void main()
        theApp.cpp
                                  cout<<"<TEST 1> \n";
  🖃 📵 Header Files
                                  //spool::getNumberOfSpool():
        🗐 job.h
        spool.h
                                  // liste des jobs qui sont intégrés dans le spool : votre jeu d'essai
     Resource Files
                                  job A(1, "AP.xls", true);
                                  job B(2, "impression admin.txt", false);
                                  job C(3, "test test.cpp", true);
job D(4, "note de synthèse.doc", false);
job E(5, "help.sos", true);
                                  //job F(6, "TP2.doc", false);
                                  //job G(7, "math.jpg", true);
                                  //job H(8, "readme.txt", false);
                                  //job I(9, "durdur.htm", true);
                                  //job J(10, "beautiful.bmp", false);
                                  // le spool
                                  spool S(1):
                                  spool::getNumberOfSpool();
                                  // ajout des jobs dans le spool
                                  S.addJobInSpool(A);
                                  S.addJobInSpool(B);
                                  S.addJobInSpool(C);
                                  S.addJobInSpool(D);
                                  S.addJobInSpool(E);
                                  //S.addJobInSpool(F);
                                  //S.addJobInSpool(G);
                                  //S.addJobInSpool(H);
                                  //S.addJobInSpool(I);
                                  //S.addJobInSpool(J);
                                  // Affichage des données propres au spool + liste des jobs en attente
                                  // La liste des jobs triéé croissante puis décroissante sur la priorité et l'arrivée dans la file
                                  S.getDataOnSpool();
```