```
/****
 *
 * Class RolodesMenu is an operational view of the Rolodex model class in the
 * menu-based GUI.  The RolodesMenu window is an pulldown menu.  Classes
 * AddMenuItem, DeleteMenuItem, ChangeMenuItem, and FindMenuItem specialize
 * InterViews MenuItem to define the items on the RolodesMenu window.
 *
 * That RolodesMenu is an "operational" view of the rolodex model means that it is
 * a view of the model operations but not of its data.  Other views of rolodex
 * show some or all of its data.  See ./add-card-dialog.h for further
 * discussion of the correspondence between the rolodex model and its views.
 *
 */

#ifndef rolodexmenuIncluded
#define rolodexmenuIncluded

#include "rolodex.h"
#include "rolodex-menu-ui.h"
#include "view.h"
#include <InterViews/menu.h>

class RolodexMenuUI;


class RolodexMenu : public View {
  public:
    RolodexMenu(Name* n, Rolodex* r, RolodexMenuUI* rmui);
    /*
     * Construct this with the given name as the pulldown label.  The given
     * Rolodex is the companion model.  The given RolodexUI is the parent view.
     * Note that in the current non-modal implementation, the model is not
     * necessary here, since the input dialogs will invoke the model operations
     * directly.  The model is passed in as a matter of "good form" in case
     * implementation details change such that one or more menu operations
     * invoke model functions directly.
     */

    ~RolodexMenu();
    /*
     * Deep destruct this by deleting all of its items.
     */

    virtual void Compose();
    /*
     * Compose this by inserting each of its four menu items -- Add, Delete,
     * Change, and Find -- into the pulldown menu window.
     */

  protected:
    Rolodex* r;           // Companion model (downcast version of this->m)
    RolodexMenuUI* rmui;  // The parent view
    PulldownMenu* pm;     // For convenience, downcast version of this->w
};

class AddMenuItem : public MenuItem {
  public:
    AddMenuItem(Rolodex* r, RolodexMenuUI* rmui);
    /*
     * Construct this with the given rolodex model.  The given rolodex menu ui
     * is the parent view which provides access to the add card dialog.
     */

  protected:
    virtual void Do();
    /*
     * Respond to the selection of this menu item by calling the Add model
     * function.  Note that the call to Rolodex::Add will be indirect through
     * the OK button in the AddCardDialog, q.v.
     */

    Rolodex* r;                 // The compnion model
    RolodexMenuUI* rmui;        // The parent view
};

class DeleteMenuItem : public MenuItem {
  public:
    DeleteMenuItem(Rolodex* r, RolodexMenuUI* rmui);
    /*
     * Construct this with the given Rolodex model.
     */

  protected:
    virtual void Do();
    /*
     * Respond to the selection of this menu item by calling the Delete model
     * function.
     */

    Rolodex* r;                 // The compnion model
    RolodexMenuUI* rmui;        // The parent view
};

class ChangeMenuItem : public MenuItem {
  public:
    ChangeMenuItem(Rolodex* r, RolodexMenuUI* rmui);
    /*
     * Construct this with the given Rolodex model.
     */

  protected:
    virtual void Do();
    /*
     * Respond to the selection of this menu item by calling the Change model
     * function.
     */

    Rolodex* r;                 // The compnion model
    RolodexMenuUI* rmui;        // The parent view
};

class FindCardMenuItem : public MenuItem {
```

```
113   public:
114     FindCardMenuItem(Rolodex* r, RolodexMenuUI* rmui);
115     /*
116      * Construct this with the given Rolodex model.
117      */
118
119   protected:
120     virtual void Do();
121     /*
122      * Respond to the selection of this menu item by calling the Find model
123      * function.
124      */
125
126     Rolodex* r;                 // The compnion model
127     RolodexMenuUI* rmui;        // The parent view
128 };
129
130 #endif
```