```
/****
 *
 * Class FindCardDialog contains a string editor for holding the name of the
 * card(s) to be found.  Above the string editor is the find dialog message.
 * Below the string editors are OK and Cancel buttons.
 *
 */

#ifndef findcarddialogIncluded
#define findcarddialogIncluded

#include "rolodex.h"
#include "view.h"
#include <InterViews/box.h>
#include <InterViews/button.h>
#include <InterViews/streditor.h>

class CancelFindButton;
class OKFindButton;
class RolodexMenuUI;            // Avoid circular include


class FindCardDialog : public View {
  public:
    FindCardDialog(Rolodex* r, RolodexMenuUI* rmui);
    /*
     * Construct the dialog to enter the name of cards to be found in the
     * rolodex.  The given Rolodex is the companion model.  The given
     * RolodexMenuUI is the parent view.
     */

    ~FindCardDialog();

    virtual void Compose();
    /*
     * Compose this by inserting a labeled string editor for entering the card
     * id.
     */

    String* GetName();
    /*
     * Return the text value of the name string editor.
     */

  protected:

    void ComposeButtons();
     /*
      * Compose the OK and Cancel dialog buttons.
      */


    void ShowResults();
    /*
     * Display the results of the find command in browsable sequence of
     * screens.
     */

    CardList* cl;               // List of most recent Find results

    StringEditor* sen;          // String editor for name field
    ButtonState* bs;            // Dummy (i.e., unused) button state for all
                                // string editors

    VBox* vb;                   // VBox for holding the label, str ed, and
                                // button box.
    HBox* hb;                   // Hbox for holding the buttons.

    OKFindButton* okfb;         // OK button
    CancelFindButton* cfb;      // Cancel button

};

class OKFindButton : public PushButton {

  public:
    OKFindButton(Rolodex* r, FindCardDialog* fcd);
    /*
     * Construct this with the given rolodex model and parent dialog view.
     * Access to the model is for calling its Find function.  Access to the
     * parent dialog is for gathering data to be sent to Rolodex::Find.  See
     * the specs for Press below for further discussion.
     */

  protected:
    virtual void Press();
    /*
     * Respond to a press of this button by calling Rolodex::Find.  The id
     * of the card sent as the parameter to Find is gathered from the string
     * editor for the card id field.  This editor is contained in the parent
     * dialog, and is hence a view sibling of this button.
     */

    Rolodex* r;                         // The companion model
    FindCardDialog* fcd;                // The parent view
};

class CancelFindButton : public PushButton {

  public:
    CancelFindButton(RolodexMenuUI* rmui);
    /*
     * Construct this with the given grandparent rolodex view.  Access to the
     * parent view is for the purposes of effecting the cancelation, which in
     * terms of the view means to replace the delete card dilaog with an empty
     * data entry area.  Note that no model access is necessary in this button
     * since pressing this has no effect on the rolodex, only on the rolodex
     * view.
     */

  protected:
    virtual void Press();
```

```
113      /*
114       * Respond to a press of this button by calling
115       * RolodexMenuUI::ShowNoDialog.
116       */
117
118      RolodexMenuUI* rmui;                // The grandparent view
119  };
120  #endif;
```