```
/****
 *
 * Class AddCardDialog is a partial rolodex view containing string editors for
 * entering the values of the fields in a rolodex card.  Above the string
 * editors is a prompting message directed to the user (i.e., "Enter
 * Information ...:").  Below the string editors are OK, Clear, and Cancel
 * buttons.
 *
 * That AddCardDialog is a "partial" rolodex view means that the companion view
 * for the model is a rolodex, however the view does not show all of the
 * rolodex contents.  What precisely AddCardDialog shows is a single card that
 * will be used as the input parameter to Rolodex::Add.  There is no model data
 * member to which AddCardDialog corresponds.
 *
 * This is an example of a general MVP principle that views may correspond to
 * model data independent of where the model data are allocated.  Frequently a
 * view corresponds to a model datum that is allocated as a data member in some
 * larger containing class.  Howerver, it is perfectly acceptable to have a
 * view correspond to a model datum that is allocated as a function parameter.
 *
 * One might consider the model for AddCardDialog view to be a single card,
 * rather than the larger rolodex.  However, in addition to the view of a card,
 * AddCardDialog has an OK button that is a view of the Rolodex::Add operation.
 * Hence, the dialog is larger than just a card itself and hence the term
 * "partial" rolodex view is used.  Such terminology is not really that
 * important.  What *is* important is that each view has a companion model that
 * represents some or all of the data and functions depicted in the view.  It's
 * OK if the model has more data than are depicted in the view.
 *
 */

#ifndef addcarddialogIncluded
#define addcarddialogIncluded

#include "add-card-buttons.h"
#include "str.h"
#include "view.h"
#include <InterViews/box.h>
#include <InterViews/streditor.h>

class RolodexMenuUI;            // Avoid circular include


class AddCardDialog : public View {

  public:

    AddCardDialog(Rolodex* r, RolodexMenuUI* rmui);
    /*
     * Construct the dialog to enter the fields of a card and add it to the
     * rolodex.  The given Rolodex is the companion model.  The given
     * RolodexMenuUI is the parent view.
     */

    virtual void Compose();
    /*
     * Compose this by inserting a labeled string editor for each of the five
     * card fields.
     */

    String* GetName();
    String* GetId();
    String* GetAge();
    String* GetSex();
    String* GetAddr();
    /*
     * Return the text value of the appropriate string editor -- Name, Id, Age,
     * Sex, or Address.
     */

    void Clear();
    /*
     * Clear the string editors of all that may have been entered data.
     */


  protected:

    void ConstructEditors();
    /*
     * Consruct and properly size the field string editors.
     */

    void ComposeLabels();
    /*
     * Insert each field label into the labels vbox.
     */

    void ComposeEditors();
    /*
     * Insert each field string editor in stiing editors vbox.
     */

    void ComposeOuter();
    /*
     * Insert the label and string editor boxes into an outer hbox with some
     * stretchable glue.
     */


    StringEditor* sename;       // String editor for name field
    StringEditor* seid;         // String editor for id field
    StringEditor* seage;        // String editor for age field
    StringEditor* sesex;        // String editor for sex field
    StringEditor* seaddr;       // String editor for address field
    ButtonState* bs;            // Dummy (i.e., unused) button state for all
                                // string editors

    VBox* vblabels;             // VBox for holding editor labels
    VBox* vbseds;               // VBox for holding string editors
    HBox* hbouter;              // HBox for holding the above two vboxes

```

```
    AddCardButtons* acb;        // HBox containing add card buttons OK, Clear,
                                // and Cancel
};

#endif
```