```
/****
 *
 * Class AddCardDialog is a card view containing string editors for entering
 * each of the card field values.  Above the string editors is a message
 * directed to the user (i.e., "Enter card field values: ".  Below the string
 * editors are OK, Clear, and Cancel buttons.
 *
 * Note that AddCardDialog is a view of an input parameter to function
 * Rolodex::AddCard.  There is no model data member to which AddCardDialog
 * corresponds.  This is an example of a general MVP principle views correspond
 * to model data, independent of where the model data are allocated.
 * Typically, a view correspond 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.
 *
 */

#ifndef addcarddialogIncluded
#define addcarddialogIncluded

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


class AddCardDialog : public View {

  public:

    AddCardDialog();
    /*
     * Construct the dialog to enter the fields of a card and add it to the
     * rolodex.
     */

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


  protected:

    virtual void ComposeCardFieldEditors();
    /*
     * Worker bee called from Compose.  ComposeCardFieldEditors creates each of
     * the labeled string editors for the card fields.
     */

    virtual void ComposeOneFieldEditor(HBox* hb, char* label);
    /*
     * Worker beelet called from ComposeCardFieldEditors.
     * ComposeOneFieldEditor inserts the given label in the given HBox,
     * followed by some glue, followed by an empty string editor.
     */


    HBox* hbname;       // String editor for name field
    HBox* hbid;         // String editor for id field
    HBox* hbage;        // String editor for age field
    HBox* hbsex;        // String editor for sex field
    HBox* hbaddr;       // String editor for address field
    ButtonState* bs;    // Dummy (i.e., unused) button state for all str eds
};

#endif
```