```
/****
 *
 * Class AddCardButtons is a view containing OK, Clear, and Cancel buttons for
 * the add card dialog.
 *
 */

#ifndef addcardbuttonsIncluded
#define addcardbuttonsIncluded

#include "view.h"
#include <InterViews/button.h>

class Rolodex;
class AddCardDialog;
class RolodexMenuUI;
class OKAddButton;
class ClearAddButton;
class CancelAddButton;


class AddCardButtons : public View {

  public:

    AddCardButtons(Rolodex* r, AddCardDialog* acd, RolodexMenuUI* rmui);
    /*
     * Construct this with the given inputs.  The given Rolodex is the
     * companion model; the given add card dialog is the immediate view parent;
     * the given rolodex menu ui is the view grandparent.  One or more of these
     * model or view pointers is passed to the buttons contained in this.  See
     * the button constructor specs below for further discussion.
     */

    ~AddCardButtons();

    virtual void Compose();
    /*
     * Compose this by inserting the buttons into an hbox.
     */

  protected:

    OKAddButton* okab;          // OK button
    ClearAddButton* clrab;      // Clear button
    CancelAddButton* canab;     // Cancel button
};


class OKAddButton : public PushButton {

  public:
    OKAddButton(Rolodex* r, AddCardDialog* acd, RolodexMenuUI* rmui);
    /*
     * Construct this with the given rolodex model and parent dialog view and
     * grandparent rolodex view.  Access to the model is for calling its Add
     * function.  Access to the parent dialog is for gathering data to be sent
     * to Rolodex::Add.  Access to the grandparent view is for reporting input
     * error messages.  See the specs for Press below for further discussion.
     */

  protected:
    virtual void Press();
    /*
     * Respond to a press of this button by calling Rolodex::Add.  The card
     * data sent as the parameter to Add are gathered from the string editors
     * for the card fields.  These editors are contained in the parent dialog,
     * and are hence view siblings of this button.
     */

    Rolodex* r;                 // The companion model
    AddCardDialog* acd;         // The parent view
    RolodexMenuUI* rmui;        // The grandparent view
};

class ClearAddButton : public PushButton {

  public:
    ClearAddButton(AddCardDialog* acd);
    /*
     * Construct this with the given parent dialog view.  Access to the parent
     * dialog is for clearing any data entered into the dialog string editors.
     * Note that no model access is necessary in this button since pressing
     * this has no effect on the rolodex, only on the add card dialog view.
     */

  protected:
    virtual void Press();
    /*
     * Respond to a press of this button by calling AddCardDialog::Clear.
     * This in turn wlll clear all data out of the
     */

    AddCardDialog* acd;         // The parent view
};

class CancelAddButton : public PushButton {

  public:
    CancelAddButton(RolodexMenuUI* rmui);
    /*
     * Construct this with the given grandparent rolodex view.  Access to the
     * grandparent view is for the purposes of effecting the cancelation, which
     * in terms of the view means to replace the add 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();
    /*
```

```
     * Respond to a press of this button by calling
     * RolodexMenuUI::ShowNoDialog.
     */

    RolodexMenuUI* rmui;        // The grandparent view
};

#endif
```