```
/****
 *
 * Class FileMenu is the companion view to the File model class.  The FileMenu
 * window is an InterViews PullDownMenu.
 *
 * Classes NewMenuItem, OpenMenuItem, CloseMenuItem, CloseAllMenuItem,
 * SaveMenuItem, SaveAsMenuItem, SaveAllMenuItem, PrintMenuItem, and
 * QuitMenuItem specialize InterViews MenuItem to define the items on the
 * FileMenu window.
 *
 */

#ifndef filemenuIncluded
#define filemenuIncluded

#include "file.h"
#include "view.h"
#include <InterViews/menu.h>


class FileMenu : public View {
  public:
    FileMenu(Name* n, File* f);
    /*
     * Construct this with the given name as the pulldown label.  The given
     * File is the companion model.
     */

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

    virtual void Compose();
    /*
     * Compose this by inserting each of its six menu items -- New, Open, Save,
     * SaveAs, Print, and Quit -- into the pulldown menu window.
     */

  protected:
    File* f;            // Companion model (downcast version of this->m)
    PulldownMenu* pm;   // For convenience, downcast version of this->w
};

class NewMenuItem : public MenuItem {
  public:
    NewMenuItem(File* f);
    /*
     * Construct this with the given File model.
     */

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

    File* f;            // The compnion model
};

class OpenMenuItem : public MenuItem {
  public:
    OpenMenuItem(File* f);
    /*
     * Construct this with the given File model.
     */

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

    File* f;            // The compnion model
};

class CloseMenuItem : public MenuItem {
  public:
    CloseMenuItem(File* f);
    /*
     * Construct this with the given File model.
     */

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

    File* f;            // The compnion model
};

class CloseAllMenuItem : public MenuItem {
  public:
    CloseAllMenuItem(File* f);
    /*
     * Construct this with the given File model.
     */

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

    File* f;            // The compnion model
};
```

```
113 class SaveMenuItem : public MenuItem {
114   public:
115     SaveMenuItem(File* f);
116     /*
117      * Construct this with the given File model.
118      */
119
120   protected:
121     virtual void Do();
122     /*
123      * Respond to the selection of this menu item by calling the Save model
124      * function.
125      */
126
127     File* f;            // The compnion model
128 };
129
130 class SaveAsMenuItem : public MenuItem {
131   public:
132     SaveAsMenuItem(File* f);
133     /*
134      * Construct this with the given File model.
135      */
136
137   protected:
138     virtual void Do();
139     /*
140      * Respond to the selection of this menu item by calling the SaveAs model
141      * function.
142      */
143
144     File* f;            // The compnion model
145 };
146
147 class SaveAllMenuItem : public MenuItem {
148   public:
149     SaveAllMenuItem(File* f);
150     /*
151      * Construct this with the given File model.
152      */
153
154   protected:
155     virtual void Do();
156     /*
157      * Respond to the selection of this menu item by calling the SaveAll model
158      * function.
159      */
160
161     File* f;            // The compnion model
162 };
163
164 class PrintMenuItem : public MenuItem {
165   public:
166     PrintMenuItem(File* f);
167     /*
168      * Construct this with the given File model.
169      */
170
171   protected:
172     virtual void Do();
173     /*
174      * Respond to the selection of this menu item by calling the Print model
175      * function.
176      */
177
178     File* f;            // The compnion model
179 };
180
181 class QuitMenuItem : public MenuItem {
182   public:
183     QuitMenuItem(File* f);
184     /*
185      * Construct this with the given File model.
186      */
187
188   protected:
189     virtual void Do();
190     /*
191      * Respond to the selection of this menu item by calling the Quit model
192      * function.
193      */
194
195     /**
196      * Data members.
197      */
198     File* f;            // The compnion model (downcast copy of this->m)
199     PulldownMenu* pm;   // The window (Downcast copy of this->w)
200 };
201
202 class SystemTestMenuItem : public MenuItem {
203   public:
204     SystemTestMenuItem(File* f);
205     /*
206      * Construct this with the given File model.
207      */
208
209   protected:
210     virtual void Do();
211     /*
212      * Respond to the selection of this menu item by calling the SystemTest
213      * model function.
214      */
215
216     /**
217      * Data members.
218      */
219     File* f;            // The compnion model (downcast copy of this->m)
220     PulldownMenu* pm;   // The window (Downcast copy of this->w)
221 };
222
223 #endif
```