```
/****
 *
 * Implementation of edit-menu.h.
 *
 * The implementation is partially stubbed in that it supplies only dummy
 * parameter values to model functions.
 *
 */

#include "edit-menu.h"


EditMenu::EditMenu(Name* n, Edit* e) : View(NULL, e) {
    /*
     * For convenience, save a copy of the model as a downcast value.
     */
    this->e = e;

    /*
     * Make the window of this an IV pulldown menu.
     */
    w = pm = new PulldownMenu(n->ConstConvert());
}

EditMenu::~EditMenu() {
}

void EditMenu::Compose() {
    /*
     * Insert the six menu items, passing each the File model for subsequent
     * access in menu item Do functions.
     */
    pm->Include(new UndoMenuItem(e));
    pm->Include(new RedoMenuItem(e));
    pm->Include(new RepeatMenuItem(e));
    pm->Include(new CutMenuItem(e));
    pm->Include(new CopyMenuItem(e));
    pm->Include(new PasteMenuItem(e));
    pm->Include(new EditDeleteMenuItem(e));
    pm->Include(new SelectAllMenuItem(e));
}

UndoMenuItem::UndoMenuItem(Edit* e) : MenuItem("Undo") {
    this->e = e;
}

RedoMenuItem::RedoMenuItem(Edit* e) : MenuItem("Redo") {
    this->e = e;
}

RepeatMenuItem::RepeatMenuItem(Edit* e) : MenuItem("Repeat") {
    this->e = e;
}

CutMenuItem::CutMenuItem(Edit* e) : MenuItem("Cut") {
    this->e = e;
}

CopyMenuItem::CopyMenuItem(Edit* e) : MenuItem("Copy") {
    this->e = e;
}

PasteMenuItem::PasteMenuItem(Edit* e) : MenuItem("Paste") {
    this->e = e;
}

EditDeleteMenuItem::EditDeleteMenuItem(Edit* e) : MenuItem("Delete") {
    this->e = e;
}

SelectAllMenuItem::SelectAllMenuItem(Edit* e) : MenuItem("SelectAll") {
    this->e = e;
}

void UndoMenuItem::Do() {
    e->Undo();
}

void RedoMenuItem::Do() {
    e->Redo();
}

void RepeatMenuItem::Do() {
    e->Repeat(0);
}

void CutMenuItem::Do() {
    e->Cut();
}

void CopyMenuItem::Do() {
    e->Copy();
}

void PasteMenuItem::Do() {
    e->Paste();
}

void EditDeleteMenuItem::Do() {
    e->Delete();
}

void SelectAllMenuItem::Do() {
    e->SelectAll();
}
```