目录：

1 项目名称................................................................................2

2 项目简介................................................................................2

3 开发环境................................................................................2

4 相关设计模式简介................................................................2

5 操作截图 ..............................................................................3

6 项目思想简介.........................................................................7

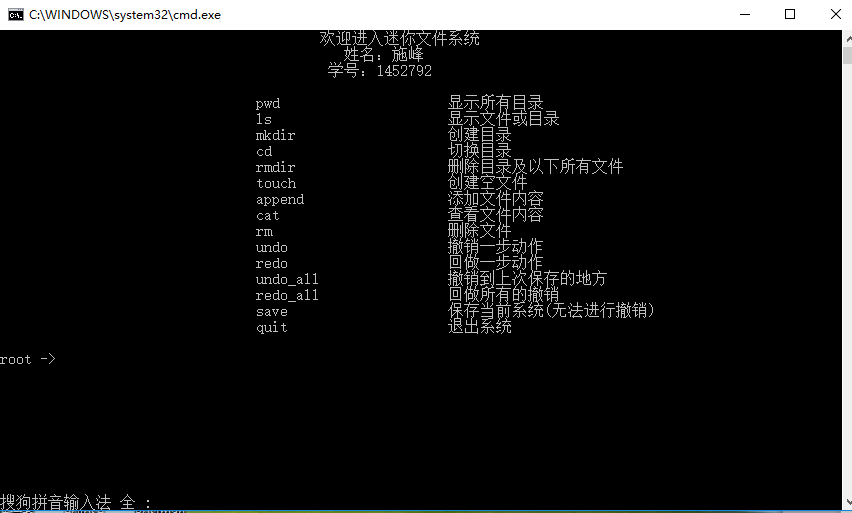
1. 项目名称：迷你文件管理系统
2. 项目简介：

本项目为选题二，在C++下实现三种设计模式以及设计一个小应用来使用上这这些模式。以下是项目进展:

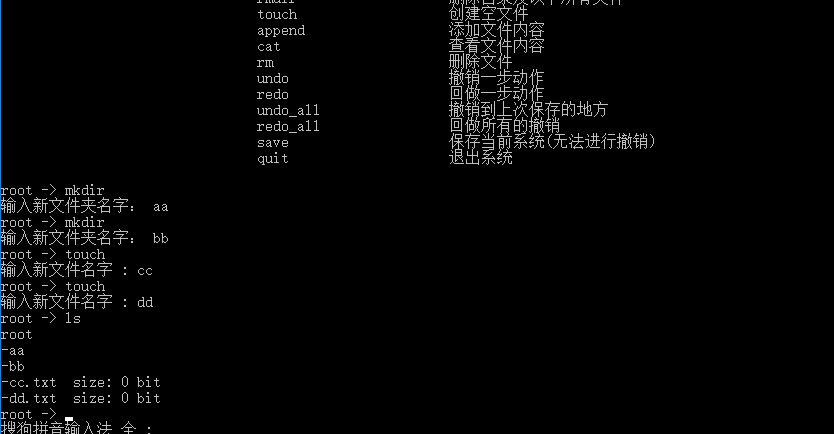
1. 项目立项初期，实现了侯捷老师讲义上的三种设计模式的接口，并进行了简单的测试
2. 项目实现阶段，在win32控制台应用程序下，实现了一个模拟的文件管理系统，该文件管理系统具有基本的文件系统功能以及undo/redo(包括撤销多步与重做多步) 等使用命令模式实现的功能。
3. 开发环境

Visial stdio 2015下的win32控制台程序

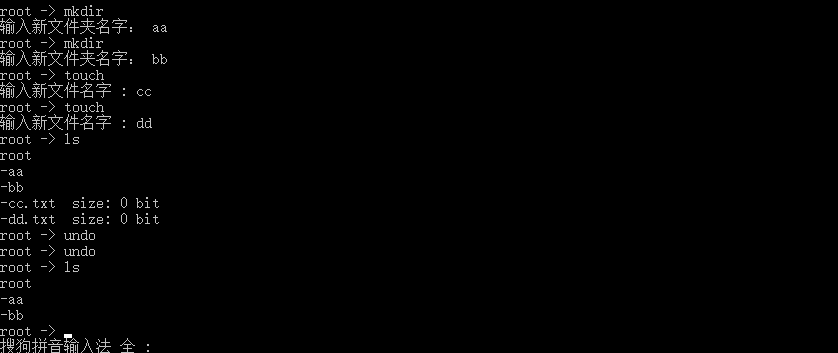
1. 相关设计模式简介
   1. 观察者模式：文件系统“拥有”一个遥控器，这个“遥控器”会向一个叫“support”的类注册自己。当文件进行添加或删除等功能时，support会将文件的操作包装成一个“事件”，同时通知这个遥控器，让它接收这个操作，放到自己内部的文件操作队列中。
   2. 组合模式：在文件系统中，文件夹与文件都派生于一个虚基类，在进行文件的操作时，无论是“单个”的文件与“整体”的文件夹都可以看成是同一类型的对象进行操作与删除这样使得代码易于维护与编写。
   3. 命令模式：用来实现文件系统的undo/redo功能。将文件的操作通过观察者模式，发送给“遥控器”类，该类会将所有的操作保存到自己内部的队列中。当接收到undo/redo命令时，就去队列中取出相应的元素，进行操作。当undo之后又插入新的动作时，会将之后的所有操作抹去，符合实际中的使用。
2. 操作截图
   1. 进入文件系统，选择自己想要的操作：



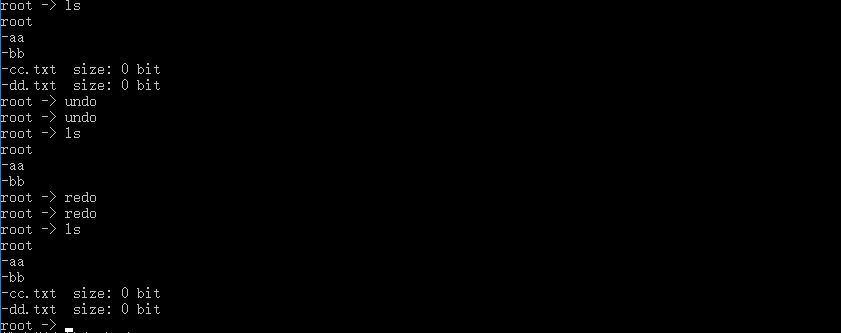
* 1. 插入2个文件夹，2个文件



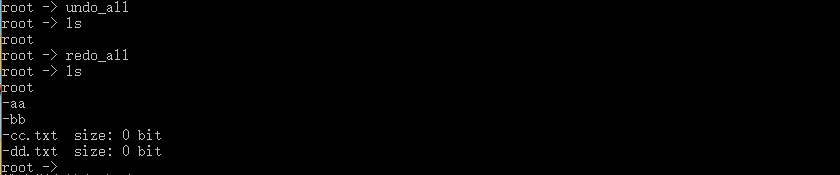
* 1. 执行两次undo,只剩下两个文件夹：



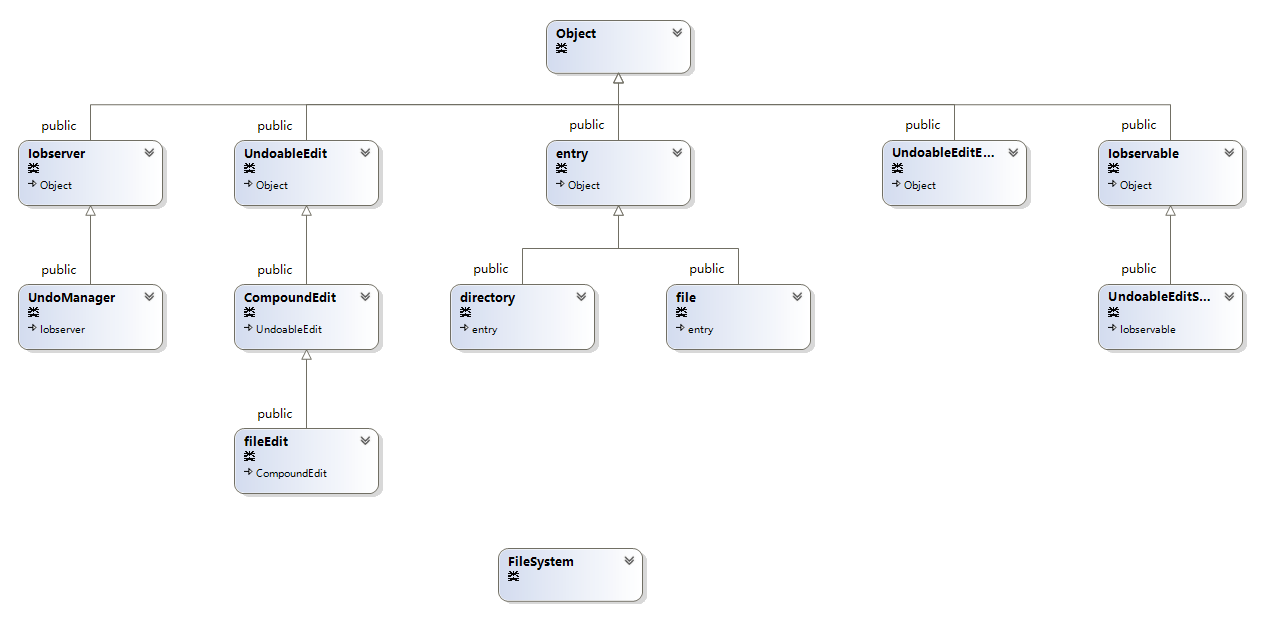
* 1. 执行两次redo,恢复原来状态：



* 1. 尝试多步撤销，输入undo\_all,与redo\_all

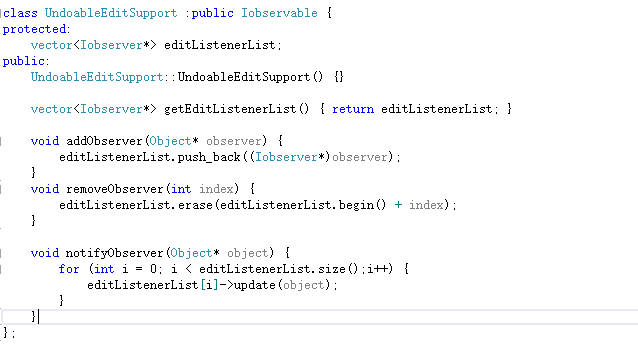


1. 项目思想简介
   1. 项目类图：



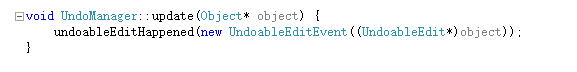
思想：让所有的类继承于一个object类，以此来模拟Java下的单根继承体系，这样在上层数据参数就可以统一用object类型的参数，不必理会下层的具体类型

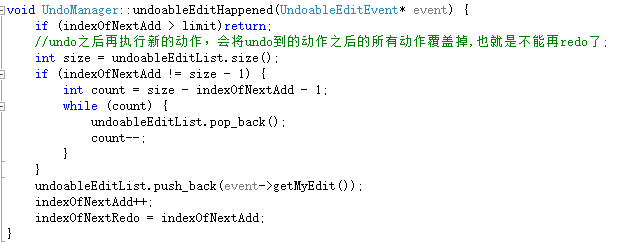
* 1. 观察者模式实现：



让support这个类继承于Iobservable，实现了添加观察者，通知观察者等功能，其中notifyObserver的参数，具体实现的时候，是一个文件操作的对象，用于命令模式中使用

* 1. 命令模式的实现：





当有新操作发生时，经过support通知会将操作的对象传给update(),update()会将这个对象包装为一个事件对象传给undoableEditHappen()这个函数，该函数会将操作插入manager类的队列中。其中indexofnext这个变量来控制下一步可以进行undo的操作的位置；

同时，每次插入会检测之前undo了多少个动作，将这些操作清空，所以每次undo之后进行新操作便不能redo回去。