Assignment1: Design of a Basic Command Shell

1. Title page

Date:10/19/2017

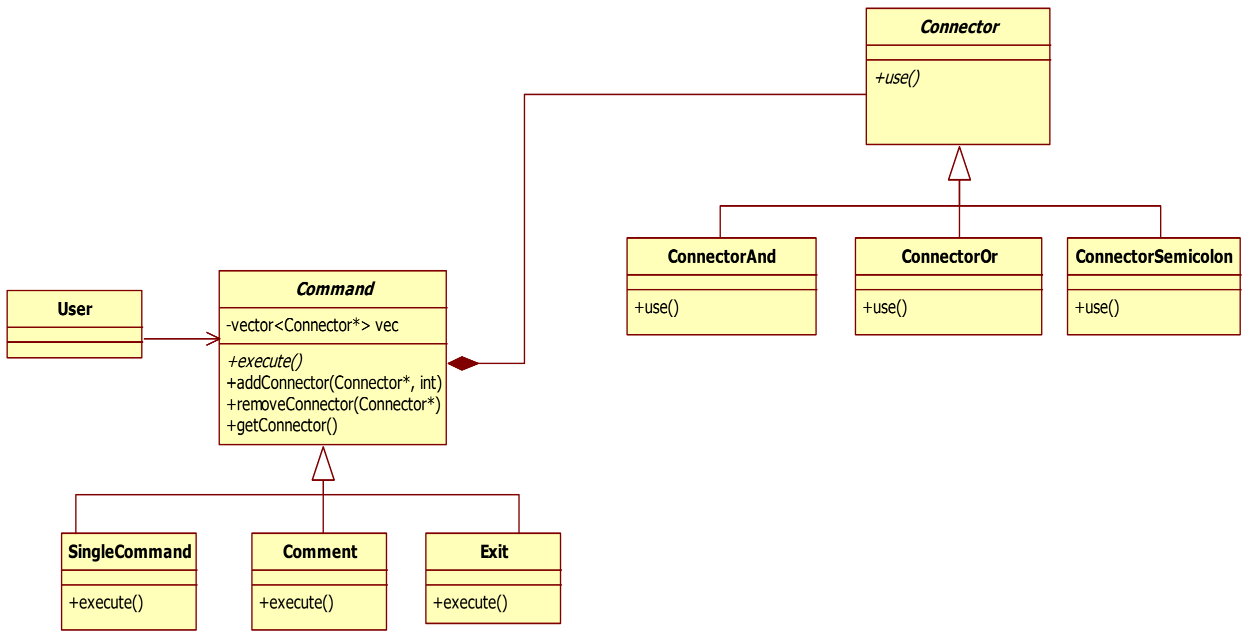
Quarter: 2017Fall

Authors: Fan Wu & Yunru Ge

2. Introduction

This project is about writing a command shell which allow the user to call commands like ls, echo, etc. We use composite pattern and strategy pattern to design the project.

3. Diagram



4. Classes/classes group

We have 9 classes in total:

1. Command
   Command is a base class of the Command class group and it has three child class. This class group is for all the commands user input and use this class to distinguish them. Command holds a collection of instances of that class Connector that the arrow is pointed to. In the abstract class Command, it has totally four operations: execute() is the vitual function and must be used in three child classes, it is used for those different command to be

used in a different way. addConnector() is used to add certain connector into the vector<Connector*>, and removeConnector() is used to those connectors which are in the comment or after the exit command. Getconnector() is used when a singleCommand is used.

Following are the three child classes of Command:

a. SingleCommand: it contains all the basic commands, for instance,the ls, echo, mkdir…. In this child class we will deal with all the single commands with the fork(), execvp() and wait().
b. Comment: this child class is used only the command is the #, which means that commands after this comment will be ignored until the comment is finished..
c. Exit: this child class is used when user input the exit word and our shell will be stop and ignore all the commands after the exit word.

2. Connector

   Connector is a base class of all the connectors user input, it contains a virtual function called use(), and it has three child classes which are used for the three different connector: and, or, semicolon.

   Following are the three child classes of Connector:

   a. ConnectorAnd: used to the && connector, use() must have the function that the next command is executed only if the first one succeeds.
   b. ConnectorOr: used to the || connector, use() must have the function that the next command is executed only if the first one fails.
   c. ConnectorSemicolon:use the ; connector, use() must have the function that the next command is always executed.

3. User

   User class is for collecting the user input and to test all the class and commands.

5. Coding strategy

As for the coding strategy, Fan is responsible for ‘command’ part(has 3 child classes), and Yunru is responsible for ‘connector’ part(also 3 child classes) and the other classes code. We both would work together for the rest part, including test all the code.

6. Roadblocks

When we are designing the project, we don’t set ‘removeconnector’ in the command class. So we may have problem when we need to edit the command. Also, if we don’t use strategy pattern, it would be difficult to maintain in the future.