Assignment 1: Design Document

November 1, 2017

Fall 2017

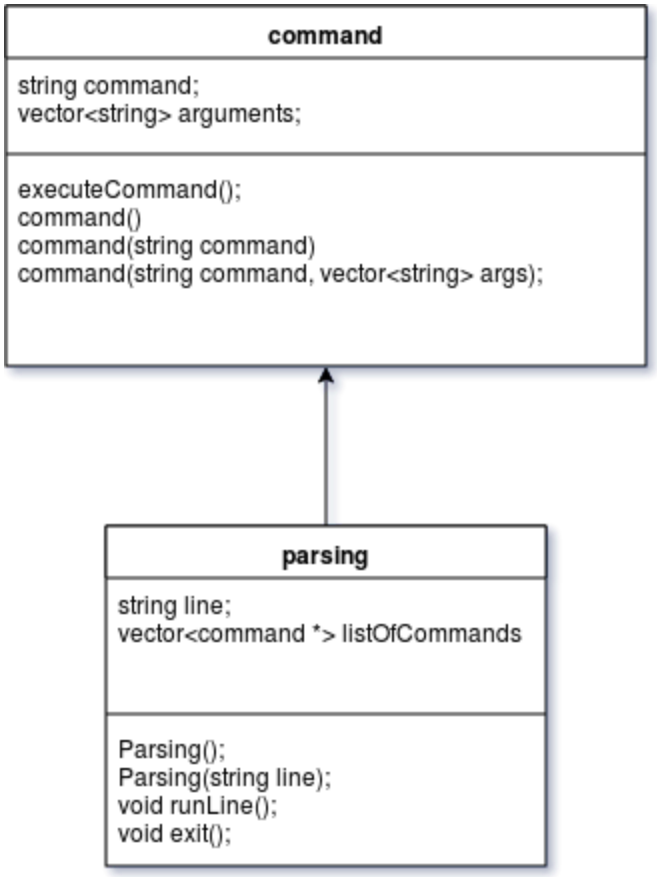
Daniel D'Orange

Jerry Tan

Introduction

With our command shell we will be using two classes to implement our design. We start by using a “parsing” class. The “parsing” class will be given a string that contains the line the user wrote. The parsing class has a “runLine()” function that iterates through the string. It breaks it down into commands, arguments, and connectors. Our Command class contains variables to store a command’s name, arguments, and its connector. It contains a function which runs the command, and determines whether or not it was successful.

Diagram



Class Groups

There will be two classes in our rshell assignment. Each class will require their own feature branch. One of the classes is the parsing class. This class will be passed in the user’s line of commands. The parsing class will begin iterating through the line

passed in. The parsing class will store the command's name, the command's arguments, and the connector following the command. This is where our other class "Command," comes into play. Parsing dynamically allocates a Command variable, and initializes it with the command name, the vector of arguments, and the connector. Commands are dynamically allocated as we do not know the number of commands the user will be using in a single line. Parsing will then attempt to execute the command, which is a function inside of Command. Command's execute function will attempt to run the command, and returns whether or not it was successful. Parsing will then determine if the following command should be executed based on the connectors used and whether or not the previous command was successful. The Parsing class will continue to iterate through the line until it reaches the end of the line.

Coding Strategy

In order to code our project and distribute the labor evenly, we will each construct one of the two classes and one of us will create the main file. Jerry will start implementing the command class, while Daniel will implement the parsing class and main file. There will be no direct inheritance from the parsing class to the command class. Each class will have its own feature branch, so there will be 3 branches total: a master branch with the main file and 2 feature branches. The master branch will be merged with the other branches once the implementation of each class is done and fully tested.

Roadblocks

For any team assignments, communication always plays a critical role towards the success of any collaborative assignments. To avoid any communication problems, we will communicate often via text or messenger. Also, we will inform each other if we plan to make additional commits and look over each other's code before committing, pushing, or merging our code. Other potential roadblocks for this assignment include mismatched naming schemes between classes when they are merged into master branch. If we do not coordinate the names of functions and classes ahead of time there could be issues. As such we will coordinate the names used in each class ahead of time. Another potential issue is using the C++ syscalls as we are unfamiliar with them at the time of this writing. This can be ameliorated by reading documentation and practicing with them prior to implementing the assignment.