

Design Patterns

Context

The context is represented by Environment. This pattern is used to provide the Executor all necessary dependencies required to execute commands.

Dependency Injection

This can be seen in almost all Executables and the Executor. This pattern is used to provide commands and the Executor their dependencies.

Iterator

This can be seen in Directory on line 43. This pattern is used to allow iterating over a Directory without exposing the internal TreeSet used to store its children.

Chain of Responsibility

This is used in JShell on line 64, Executor on line 89, UserInput, and Executable. Handling of user input is passed through JShell, then Executor, then UserInput, then the Executable. This pattern is used because each part of the chain processes and validates a part of the user input. At any point along the chain, the execution can stop due to invalid input.

Polymorphism

This can be seen in Directory on line 48, and RealEnvironment line 68. Polymorphism is used since there are multiple classes for general files and commands. Directory only cares that its children are general files rather than specifically files or directories. The command map in RealEnvironment only cares that it contains Executables since it contains different commands.

Changes

Instead of commands having access to the console and collaborating with the console directly, they now only have access to 2 output devices passed in during execution: 1 for standard output, and 1 for error output. The commands don't need to know where the output is going. This design is better because it allows the Executor that's responsible for executing commands the ability to pass in any output device that implements Output. This design can accommodate for new requirements regarding output redirection more easily. For example, if the customer wants a new feature that redirects command output to a text-to-speech engine, this can easily be done by creating a new text-to-speech output device that implements Output and passing this new output device into the command during execution.