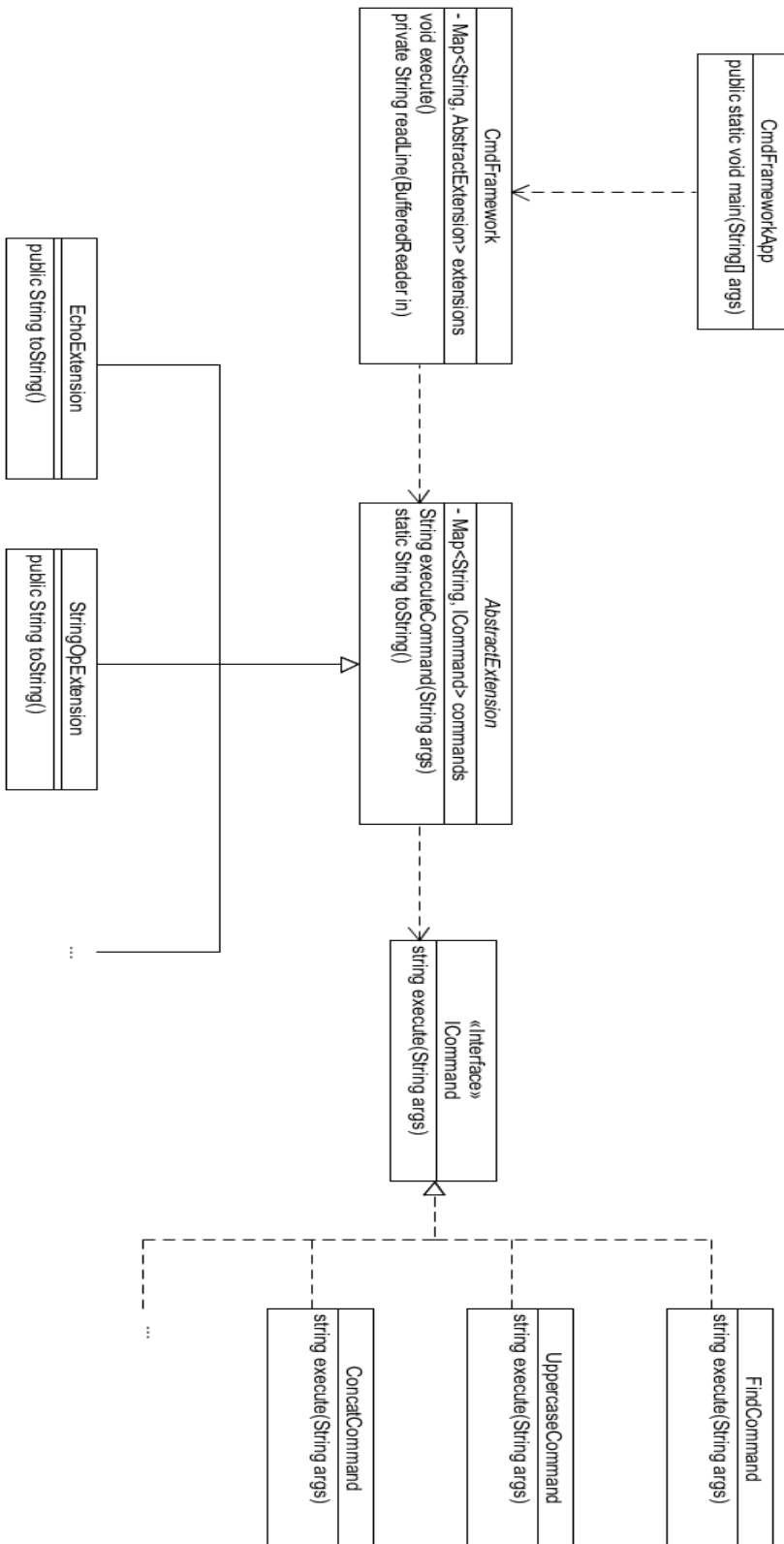


1.



The CmdFrameworkApp creates a CmdFramework object which is then run. The CmdFramework object has a map of AbstractExtensions and these extensions are values for String keys. The concrete extensions EchoExtension and StringOpExtension each have their own maps with String keys and ICommand values. Each extension parses the given String the same way in the executeCommand method so it made more sense to abstract the extensions rather than interface them to avoid code duplication. The only thing that changes is the exception being thrown so I made a toString() method for each extension. Once an extension is retrieved from the map, it receives the command string. If the extension supports the command, a corresponding ICommand object will be retrieved from its map. Each ICommand object will then execute it's execute method on the given arguments. Since each command varies a lot from the next command, it made sense to interface the commands.