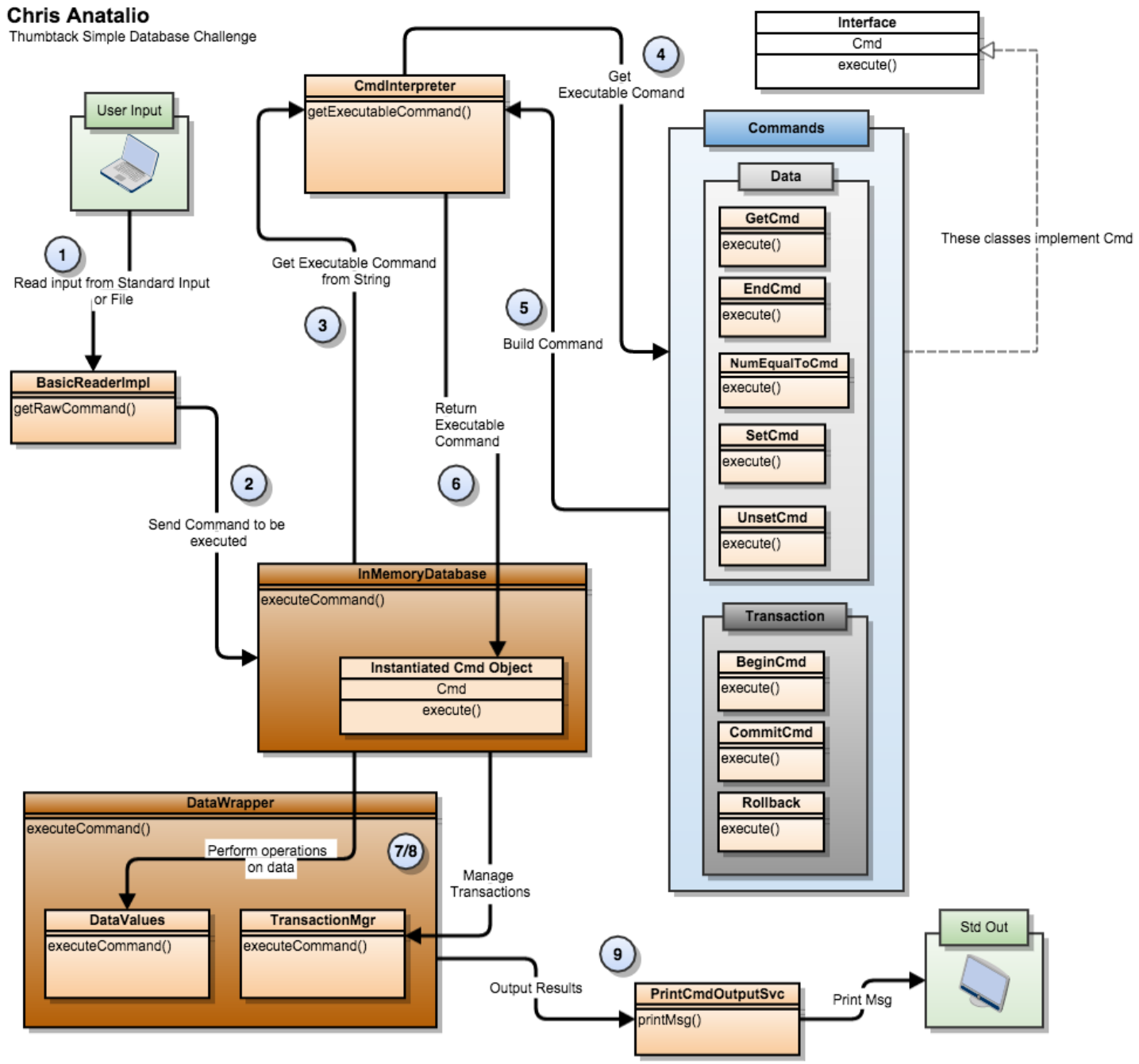

Thumbtack Simple Database Challenge

Created by: Chris Anatalio

January 20, 2016

THUMBTACK SIMPLE DATABASE CHALLENGE - CHRIS ANATALIO

Instructions to Run

- cd to location of simpledatabase.jar
- **Run in interactive mode**
  - java -jar simpledatabase.jar
- **Execute text file**
  - java -jar simpledatabase.jar /Users/canatalio/simpledatabase/src/com/thumbtack/database/commandlist1.txt
- Note use absolute path of file

SUMMARY

Implementation

A basic Reader Implementation uses a Scanner to read user input from Standard Input. This String command is sent to the In-Memory database to be execute. The String command is interpreted and and executable command object is returned. The executable command object is passed a wrapper containing the data and transaction manager and executes the commands. A print service then prints the output to standard output.

Code Coverage - Unit Tests

63% Overall code coverage
91% Command code coverage
100% Data model code coverage

Performance

Do BEGIN, GET, SET, UNSET, and NUMEQUALTO have an average-case runtime of O(log N) or better, where N is the total number of variables stored in the database?

THUMBTACK SIMPLE DATABASE CHALLENGE - CHRIS ANATALIO

Command	Operations	Complexity
BEGIN	Add operation on an ArrayList	O(1) amortized best case. O(n) worst case if array must be resized and copied
GET	Get data by key from HashMap Check transaction queue if needed	O(1) best case. O(n) worst case if currently managing a transaction.
SET	Set data in HashMap Check transaction queue if needed	O(1) best case. O(n) worst case if currently managing a transaction.
UNSET	Remove data from HashMap Check transaction queue if needed	O(1) best case. O(n) worst case if currently managing a transaction.
NUMEQUALTO	Check if data is in HashMap Check transaction queue if needed	O(1) best case. O(n) worst case if currently managing a transaction.

Do the vast majority of transactions will only update a small number of variables and is the solution efficient about how much memory each transaction uses.

Each transaction will generally on have to update a small amount of variables.

Transactions are stored using ArrayLists which are some of the most efficient storage structures in the JDK

Sources

Vorontsov, Mikhail. "Memory Consumption of Popular Java Data Types - Part 1 - Java Performance Tuning Guide." Java Performance Tuning Guide. Java-Performance, 09 July 2013. Web. 20 Jan. 2016.

"Best, Worst and Average Case." Wikipedia. Wikimedia Foundation, n.d. Web. 20 Jan. 2016.

"ArrayList (Java Platform SE 7)." ArrayList (Java Platform SE 7). N.p., n.d. Web. 20 Jan. 2016.

"When to Use LinkedList over ArrayList?" Stack Overflow. Stack Overflow, n.d. Web. 20 Jan. 2016.