|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Word pairs | -check for proper format  -parse each word  -check for word ladder solution  -prepare word ladder solution for output  -clear previous solution to prepare for next word pair if any | -display error message for improper input format or improper word  -display word ladder solution or no solution |
| Dictionary | -process file based off assignment 4 guidelines  -store dictionary into hashset | None |

Assignment 4 IPO Diagram

Assignment 4 UML

Java Class

Assign4Driver

main(String[])

Java Class

NoSuchLadderException

Java Class

WordLadderSolver

**private** List<String> solutionList

**private** HashSet<String> dict

**private** HashSet<String> visited

computeLadder(String, String): List<String>

makeLadder(String, String, **int**): boolean

getDifferenceIndex(String, String): int

compareLetters(String, String): int

clear()

Java Interface

Assignment4Interface

computeLadder(String, String): List<String>

validateResult(String, String, List<String>): boolean

clear()

The algorithm needed for the driver logic (main method)

The main method first checks for the appropriate number of arguments. If this fails, the program exits after outputting an error message. Then it initializes a WordLadderSolver object with the appropriate dictionary input. Then it begins to read the word pair inputs. The main method checks for proper syntactical formatting when parsing the word pairs. The WordLadderSolver then takes the pair to find its word ladder if one exists. The main takes these results and outputs the ladder if there is one or an error message as appropriate. This process is repeated until the end of the input file is reached.