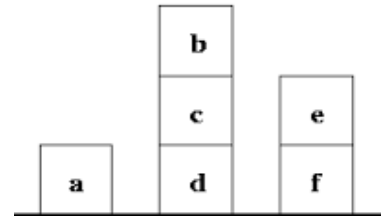


CSCI 2073 – Fall 2018 - Programming Assignment 3

Your task is to simulate a “blocks world” in which a robot arm performs tasks involving the manipulation of blocks under certain constraints. Your program will act as a robotic arm that responds to a limited set of commands that instructs the robot arm in how to manipulate blocks that lie on a flat table. Blocks are labeled with lowercase alphabetic characters and arranged in three piles, as shown in the figure.



You are to write the Robot class, which will have the following public methods (of course, you may need additional helper methods to aid in implementing the required methods):

`Robot(String blocks)`: a constructor that initializes the robot’s world. The string argument contains characters representing the blocks in each of the three piles, separated by commas. For each pile, the block at the bottom of the pile is listed first. For instance, the world in the figure above would be represented by the string `"a,dcb,fe"` as an argument to the constructor.

`boolean readActions(String filename)`: a method to read and process commands for the robot to perform. The valid commands for the robot arm are:

- move X over Y: where X and Y are block labels, puts block X onto the top of the pile containing block Y. Any blocks that are stacked on top of block X should be restored to their original pile in their original order.
- pile X over Y: where X and Y are block labels, puts the pile of blocks consisting of block X, and any blocks that are stacked above X, onto the top of the pile containing block Y. The blocks stacked above block X retain their order when moved.
- discard X: remove block X from its pile and restore other blocks in their original order.
- quit

Any command involving a nonexistent block, or a command in which X and Y are in the same pile of blocks, should be ignored and have no effect (the robot is smart enough to realize the folly of such an action). The method returns true if the file is opened and processed successfully.

`String toString()`: return a properly formatted string with the contents of the three piles, one per line. For each pile, the block at the top should be listed first. The format should match the sample below as closely as possible.

Your class should be stored in a file named `Robot.java` and should be tested using the `RobotTest.java` class and text files provided by the instructor. In addition, use the `StackInt.java` and `LinkedList.java` files provided. Any stack must be declared as follows:

```
StackInt<Type> myStack = new LinkedList<>();
```

When finished, submit `Robot.java` to Mimir for testing. Do not submit `RobotTest.java`, `StackInt.java`, `LinkedList.java`, nor any .class files to Mimir.

Sample input file contents:

```
move a over c
move f over b
pile b over e
discard a
quit
```

Sample toString value:

```
PILE 1:
PILE 2: c d
PILE 3: f b e
```