Input:

```
[Widget_type] [Widget_name]//add widget

[look-and-feel_style]/*set [look-and-feel_style] as current style*/

Present /*extra command, show all widgets to standard output*/

...
```

Output:

```
/*

Widgets must be shown with following rules:

     Window should be shown before ScrollBar.

     ScrollBar should be shown before Button.

If there are the same type widgets, show with the sequential order from
input.
*/

[Style_widget_type] [Widget_name]

...
```

Comment:

```
[Widget_type] must be one of followings:

     Window

     ScrollBar
```

Button

[look-and-feel_style] must be one of followings:

Motif

PM

if current [look-and-feel_style] is Motif, [Style_widget_type] must be one of followings:

MotifWindow

MotifScrollBar

MotifButton

if current [look-and-feel_style] is PM, [Style_widget_type] must be one of followings:

PMWindow

PMScrollBar

PMButton

The default [look-and-feel_style] is Motif.

You should read input from file.

And show output to standard output.

You are asked to write a main function in Class Main.

We'll test your program through "java Main inputFile"

```
e.g java Main sampleInput
```

Upload:

Please push your source code to the master branch of your team's homework Gitlab repository.
The folder structure should be:

[dir] GUIApplication
=> [dir] Team7
=> Main.java
=> *.java (optional)
=> [name of test case].in
=> [name of test case].out

You won't receive any point if you didn't follow the directory structure or main class name or compressed format!

sampleInput:

```
ScrollBar scrollBar1
Window window1
Window window2
Button button1
Present
PM
Present
```

sampleOutput:

```
MotifWindow window1
MotifWindow window2
MotifScrollBar scrollBar1
MotifButton button1
PMWindow window1
PMWindow window2
PMScrollBar scrollBar1
PMButton button1
```