Authors:
Umangkumar Maheshbhai Patel - 40228475
Raveena Choudhary - 40232370

Assignment 3- APP

Observer pattern is used as a base application to write aspectj definition named as "Instrumentator.aj"

"Instrumentator.aj" covers method used in base application such as:

1. To add an observer: addObserver(Observer observer)
2. To notify observers: notifyObservers()
3. To update an observer: update()
4. To remove an observer: removeObserver(Observer observer)
5. To get headLine from Blogger: getHeadline()
6. To get current state from Observer: getState()

Also, a file "callgraph.dot" has been created through program(Please refer Logger.java) - to generate UML(Sequence Diagram) using a tool dotUML(https://dotuml.com/playground.html#).
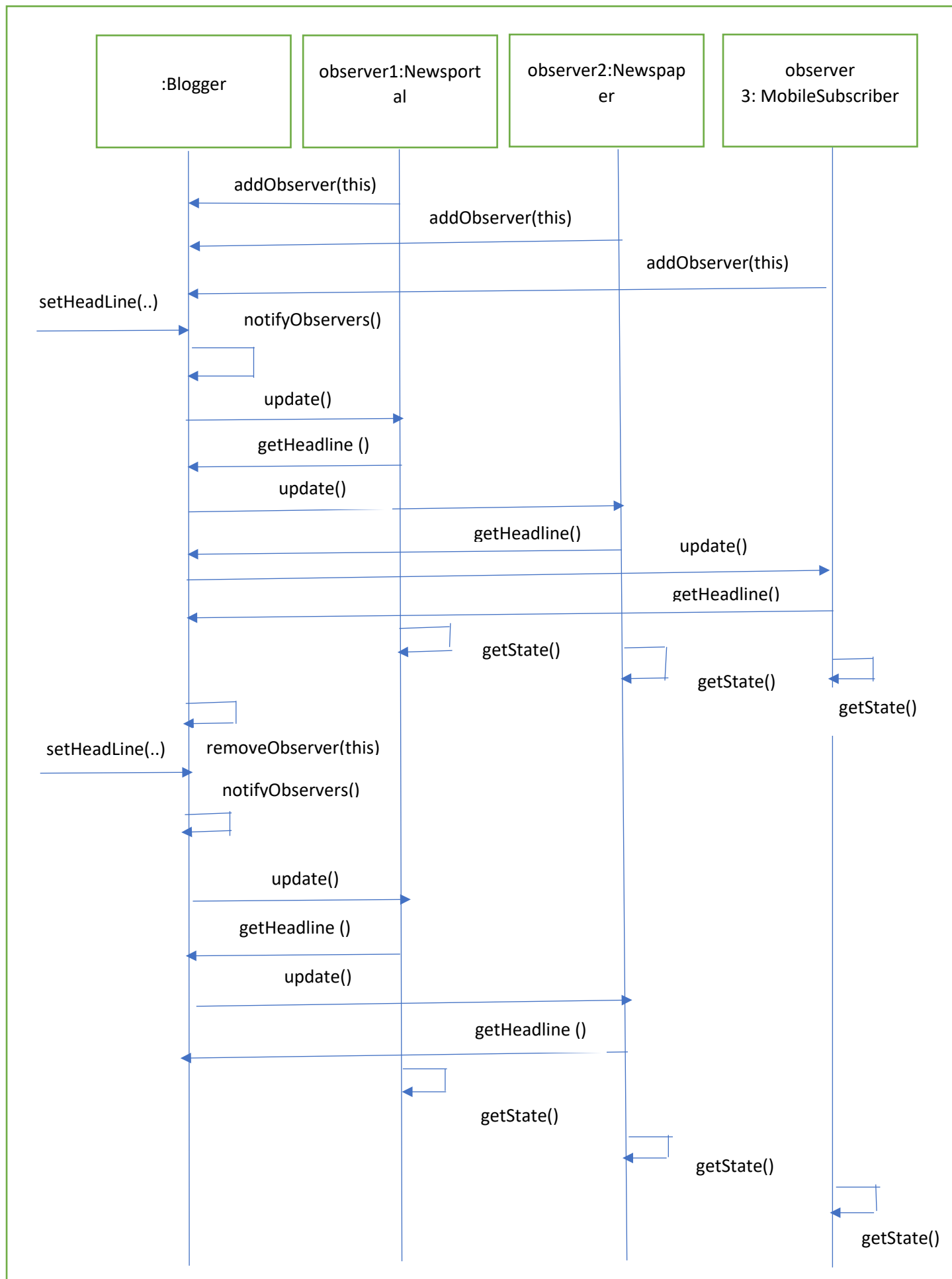
DOT graphical language is as below:

```
/*
 * Description: This is an automatically generated .DOT file
 * representing a call graph.
 * Author: Umangkumar Maheshbhai Patel - 40228475, Raveena Choudhary - 40232370
 * Date: Sun Oct 30 15:05:35 EDT 2022
 *
 */
SequenceDiagram [frame=true framecolor=steelblue label="Observer Design
PatternSequence Diagram"]{
 lifeline "Blogger:blogger" as Blogger
 lifeline "Newsportal:observer 1" as Newsportal
 Newsportal --> Blogger "1. addObserver(..)"
 lifeline "Newspaper:observer 2" as Newspaper
 Newspaper --> Blogger "2. addObserver(..)"
 lifeline "MobileSubscriber:observer 3" as MobileSubscriber
 MobileSubscriber --> Blogger "3. addObserver(..)"
 -f-> Blogger "4. setHeadline(..)"
 Blogger --> Blogger "5. notifyObservers()"
 Blogger --> Newsportal "6. update()"
 Newsportal --> Blogger "7. getHeadline()"
 Blogger --> Newspaper "8. update()"
 Newspaper --> Blogger "9. getHeadline()"
 Blogger --> MobileSubscriber "10. update()"
 MobileSubscriber --> Blogger "11. getHeadline()"
 Newsportal --> Newsportal "12. getState()"
 Newspaper --> Newspaper "13. getState()"
```

```
MobileSubscriber --> MobileSubscriber "14. getState()"
Blogger --> Blogger "15. removeObserver(..)"
-f-> Blogger "16. setHeadline(..)"
Blogger --> Blogger "17. notifyObservers()"
Blogger --> Newsportal "18. update()"
Newsportal --> Blogger "19. getHeadline()"
Blogger --> Newspaper "20. update()"
Newspaper --> Blogger "21. getHeadline()"
Newsportal --> Newsportal "22. getState()"
Newspaper --> Newspaper "23. getState()"
MobileSubscriber --> MobileSubscriber "24. getState()"
}
```

Expected Visualization:

UML Sequence Diagram: Observer Pattern

	:Blogger	observer1:Newsportal	observer2:Newspaper	observer3: MobileSubscriber

addObserver(this)
addObserver(this)
addObserver(this)
setHeadLine(..)
notifyObservers()
update()
getHeadline ()
update()
getHeadline()
update()
getHeadline()
getState()
getState()
getState()
setHeadLine(..)
removeObserver(this)
notifyObservers()
update()
getHeadline ()
update()
getHeadline ()
getState()
getState()
getState()

Visualization using dotUML tool:

Observer Design Pattern Sequence Diagram using AspectJ

| | Blogger:blogger | Newsportal:observer 1 | Newspaper:observer 2 | MobileSubscriber:observer 3 |

1. addObserver(..)
2. addObserver(..)
3. addObserver(..)
4. setHeadline(..)
5. notifyObservers()
6. update()
7. getHeadline()
8. update()
9. getHeadline()
10. update()
11. getHeadline()
12. getState()
13. getState()
14. getState()
15. removeObserver(..)
16. setHeadline(..)
17. notifyObservers()
18. update()
19. getHeadline()
20. update()
21. getHeadline()
22. getState()
23. getState()
24. getState()

Powered by DotUML