Machine generated alternative text:
ObserverTest - observer 
Class Edit Tools 
ObserverTest X 
Compile 
undo 
Options 
Copy 
Paste 
Findm 
Close 
package observer; 
ublic class Observer Test 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
new o server. 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
bublic static void main( String [ ] args ) 
ObserverTes 
ObserverTest 

*Screen clipping taken: 6/28/2018 6:37 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
StringO args = 
<0bject reference> 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:39 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ObserverTest.<init> 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
Step Into 
Continue 
Term i 
Screen clip 
2016 

*Screen clipping taken: 6/28/2018 6:40 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit 
ObserverTest X 
Compile 
undo 
package 
public 
Tools 
Options 
Copy 
Paste 
Findm 
Close 
observer ; 
void runTest() 
o server. 
e conomy s 
observer . Pessimist p 
new o server. 
e conomy 
new observer .Pessimist(s) ; 
observer . Optimist o = new observer .0ptimist(s) ; 
s. attach(p) ; 
s. attach(o) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
serv 
Cannot set breakpoint: no code in this line 
ser r est 

*Screen clipping taken: 6/28/2018 6:40 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
StringO args = 
<0bject reference> 
Halt 
Step Into 
Continue 
Observer Test - observer 
Terminate 

*Screen clipping taken: 6/28/2018 6:41 PM*

Machine generated alternative text:
Obser.'erTest - 
Class 
Edit 
ObserverTest X 
Compile 
undo 
Tools 
Options 
Copy 
Paste 
Findm 
Close 
package observer ; 
ublic class Observer Test 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( "Hey, Its Friday! 'v) ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
server es 
new 
server es 
Cannot set breakpoint: no code in this line 

*Screen clipping taken: 6/28/2018 6:41 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer. ObserverTest.main 
Static variables 
Instance variables 
Local variables 
StringO args = <0bject reference> 
Observer Test t = <0bject reference> 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:42 PM*

Machine generated alternative text:
Obser.'erTest - 
Class 
Edit 
ObserverTest X 
Compile 
undo 
Tools 
Options 
Copy 
Findm 
Close 
package observer; 
ublic class Observer Test 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new o server. 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( " Hey, Its Friday! ; 
p. showState() ; 
OWS 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 6:42 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer. Observer Test.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:42 PM*

Machine generated alternative text:
Obser.'erTest - 
Class 
Edit 
ObserverTest X 
Compile 
undo 
Tools 
Options 
Copy 
Paste 
Findm 
Close 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( "Hey, Its Friday! 'v) ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
server es 
new 
server es 

*Screen clipping taken: 6/28/2018 6:42 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = null 
private ArrayList<Observer> observers = 
Local variables 
null 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:42 PM*

Machine generated alternative text:
TheEco no 
Class Edit Tools Options 
ObserverTest X TheEconomy X 
Compile 
undo 
package observer ; 
Copy 
Paste 
Findm 
Close 
ublic class TheEconomy extends ConcreteSubject { 
public TheEconomy() 
super .setState( "The Price of gas is at $5.00/ga1 " ) , 
The Price of gas is at $5.8ß/ga1 
The New iPad is out today 

*Screen clipping taken: 6/28/2018 6:43 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.<init> 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = null 
private ArrayList<Observer> observers = 
Local variables 
null 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:43 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Findm 
Close 
private String subjectState; 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
Åub1ic void notify0bservers() { 
or (Observer obj 
observers) 
new 

*Screen clipping taken: 6/28/2018 6:43 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = null 
private ArrayList<Observer> observers = 
Local variables 
<0bject reference> 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:43 PM*

Machine generated alternative text:
TheEconomy - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
package observer; 
Copy 
Paste 
Findm 
Close 
ublic class TheEconomy extends ConcreteSubject { 
ublic TheEconomy() 
The Price of gas is at $5.8ß/ga1 
The New iPad is out today 

*Screen clipping taken: 6/28/2018 6:43 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = null 
private ArrayList<Observer> observers = 
<0bject reference> 
Local variables 
String status = 
Step Into 
Continue 
"The Price of gas is at $5.00/gal" 
Terminate 

*Screen clipping taken: 6/28/2018 6:44 PM*

Machine generated alternative text:
Com: reteSubject - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Findm 
Close 
ublic class ConcreteSubject implements Subject { 
private String subjectState; 
private observers 
ublic String getState() { 
return subjectState 
public void setState(String status) { 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ÅOb1ic void detach (Observer obj) { 
observers . remove(obj ) 
ÅOb1ic void notify0bservers() { 
or (Observer obj 
observers) 
new 

*Screen clipping taken: 6/28/2018 6:44 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
"The Price of gas is at $5.00/gal" 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:44 PM*

Machine generated alternative text:
Com: reteSubject - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Findm 
Close 
ublic class ConcreteSubject implements Subject { 
private String subjectState; 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
public void notify0bservers() { 
or (Observer obj 
observers) 
new 

*Screen clipping taken: 6/28/2018 6:44 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
*alt 
Local variables 
Step Into 
Continue 
ConcreteSubiect - observer 
Terminate 

*Screen clipping taken: 6/28/2018 6:44 PM*

Machine generated alternative text:
Com: reteSubject - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Findm 
Close 
ublic class ConcreteSubject implements Subject { 
private String subjectState; 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
public void notify0bservers() { 
new 

*Screen clipping taken: 6/28/2018 6:44 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:45 PM*

Machine generated alternative text:
Com: reteSubject - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Findm 
Close 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:45 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
"The Price of gas is at $5.00/gal" 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:45 PM*

Machine generated alternative text:
Com: reteSubject - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Findm 
Close 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:45 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.TheEconomy.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Halt 
Local variables 
Step Into 
Continue 
ConcreteSubject - observer 
Terminate 

*Screen clipping taken: 6/28/2018 6:45 PM*

Machine generated alternative text:
TheEconomy - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
package observer; 
Copy 
Paste 
Findm 
Close 
ublic class TheEconomy extends ConcreteSubject { 
ublic TheEconomy() 
super . setState( "The Price of gas is at $5.00/ga1"), 
The Price of gas is at $5.8ß/ga1 
The New iPad is out today 

*Screen clipping taken: 6/28/2018 6:46 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
Halt 
Step Into 
Continue 
TheEconomv - observer 
Terminate 

*Screen clipping taken: 6/28/2018 6:46 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Findm 
Close 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today") ; 
p. showState() ; 
o. showState() ; 
s.setState( "Hey, Its Friday! 'v) ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
server es 
new 
server es 

*Screen clipping taken: 6/28/2018 6:46 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer. Observer Test.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s 
Halt 
<0bject reference> 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:47 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Findm 
Close 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( " Hey, Its Friday! ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
server es 
ew 
server es 
Cannot set breakpoint: no code in this line 

*Screen clipping taken: 6/28/2018 6:47 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer. Pessimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = null 
Local variables 
ConcreteSubject sub 
Halt 
<0bject reference> 
Step Into 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
Close 
ublic class Pessimist extends Concrete0bserver { 
ublic Pessimist( ConcreteSubject sub ) 
bublic void update() { 
f ( subject.getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ) 
observerState 
"This is the beginning of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.<init> 
observer.Pessimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = null 
Local variables 
ConcreteSubject theSubject 
Halt 
<0bject reference> 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Conc reteO bserver 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Compile 
undo 
package observer ; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
ublic Concrete0bserver( ConcreteSubject theSubject ) 
this. subject 
theSubject 
ublic void update() { 
// do nothing 
public void showState() 
System . out . println( 
" Observer : 
+ this.getC1ass() .getName() + 
+ observerState 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.<init> 
observer.Pessimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = null 
Local variables 
ConcreteSubject theSubject 
Halt 
<0bject reference> 
Step Into 
Continue 
ConcreteObserver - observer 
Terminate 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
bublic Concrete0bserver( ConcreteSubject theSubject ) 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out . println(" Observer : 
+ this.getC1ass() .getName() + 
+ observerState 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.<init> 
observer.Pessimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
ConcreteSubject theSubject 
Halt 
<0bject reference> 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Compile 
undo 
package observer ; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
ublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out . println( 
" Observer : 
+ this.getC1ass() .getName() + 
+ observerState 

*Screen clipping taken: 6/28/2018 6:48 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
ConcreteSubject sub 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:49 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
Concreteoöserver X 
Close 
ublic class Pessimist extends Concrete0bserver { 
bublic Pessimist( ConcreteSubject sub ) 
super( sub ) 
ublic void update() { 
f ( subject .getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ) 
observerState 
"This is the beginning of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:49 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer. Observer Test. run Test 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = 
Halt 
<0bject reference> 
Step 
Step Into 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
o 
public void runTest() 
o serve . 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
Concreteobserver X 
Close 
serve . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
new o 
public void runTest() 
o server. 
e conomy s 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
Concreteobserver X 
Close 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
ser r es 
Cannot set breakpoint: no code in this line 
ser r est 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = null 
Local variables 
ConcreteSubject sub 
Halt 
<0bject reference> 
Step Into 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public Optimist( ConcreteSubject sub ) 
ublic void update() { 
if ( subject.getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ") 
observerState 
"Great! It's time to go green. 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Apple, take my money! " 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.<init> 
observer.Optimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = null 
Local variables 
ConcreteSubject theSubject 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
public Concrete0bserver( ConcreteSubject theSubject ) 
this. subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out. println( "Observer : 
+ this.getC1ass() .getName() + 
+ observerState 

*Screen clipping taken: 6/28/2018 6:50 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.<init> 
observer.Optimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = null 
Local variables 
ConcreteSubject theSubject 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:51 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
public Concrete0bserver( 
ublic void update() 
// do nothing 
ublic void showState() 
System . out . println( 
ConcreteSubject theSubject ) 
" Observer : 
+ this.getC1ass() .getName() + 
+ observerState ) ; 

*Screen clipping taken: 6/28/2018 6:51 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.<init> 
observer.Optimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
ConcreteSubject theSubject 
<0bject reference> 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:51 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
bublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubj ect 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out. println( "Observer : 
+ this.getC1ass() .getName() + 
+ observerState 

*Screen clipping taken: 6/28/2018 6:51 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.<init> 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
ConcreteSubject sub 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public Optimist( ConcreteSubject sub ) 
super( sub ) 
ublic void update() { 
if ( subject.getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ") 
observerState 
"Great! It's time to go green. 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Apple, take my money! " 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer. Observer Test.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Step Into 
Cont' 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
server es 
new 
server es 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( "Hey, Its Friday! 'v) ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
server es 
new 
server es 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.attach 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Observer obj 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) 
ublic void detach (Observer obj) 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:52 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.attach 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/g; 
private ArrayList<Observer> observers = <0bject reference 
Local variables 
Observer obj 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers. add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
Continue 
Term i na 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteoöserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( "Hey, Its Friday! 'v) ; 
p. showState() ; 
ublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.attach 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Observer obj 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
public void attach (Observer obj) 
ÅOb1ic void detach (Observer obj) 
observers . remove(obj ) 
public void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.attach 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Observer obj 
Halt 
<0bject reference> 
Step Into 
Continue 
Term i na 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers. add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:53 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The Price of gas is at $5.00/gal" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
"The New iPad is out today" 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
public void setState(String status) { 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
Halt 
"The New iPad is out today" 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:54 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ÅOb1ic void detach (Observer obj) { 
observers . remove(obj ) 
public void notify0bservers() { 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Observer obj 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
Åub1ic void notify0bservers() { 
or (Observer obj 
observers) 
new 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
81ueJ: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
Concreteobserver X 
Close 
ublic class Pessimist extends Concrete0bserver { 
public Pessimist( ConcreteSubject sub ) 
super( sub ) 
ublic void update() { 
observerState 
"This is the beginning 
Optimist X 
of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.getState 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 
ConcreteSubiect - observer 
Terminate 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
public String getState() { 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
Åub1ic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:55 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
Concreteobserver X 
Close 
ublic class Pessimist extends Concrete0bserver { 
ublic Pessimist( ConcreteSubject sub ) 
super( sub 
public void update() { 
observerState 
"This is the beginning 
Optimist X 
of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
ublic class Pessimist extends Concrete0bserver { 
ublic Pessimist( ConcreteSubject sub ) 
super( sub 
bOb1ic void update() { 
f ( subject.getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 " ) 
observerState 
observerState 
Ise 
observerState 
"This is the beginning of the end of the world! " 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.getState 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
public String getState() { 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
ublic class Pessimist extends Concrete0bserver { 
ublic Pessimist( ConcreteSubject sub ) 
super( sub 
bCib1ic void update() { 
f ( subject.getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 " ) 
observerState 
observerState 
Ise 
observerState 
"This is the beginning of the end of the world! " 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 

*Screen clipping taken: 6/28/2018 6:56 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
ublic class Pessimist extends Concrete0bserver { 
bublic Pessimist( ConcreteSubject sub ) 
super( sub 
public void update() { 
f ( subject .getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ) 
observerState 
"This is the beginning of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
Ise 
observerState 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = "Not another iPad!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
ublic class Pessimist extends Concrete0bserver { 
bublic Pessimist( ConcreteSubject sub ) 
super( sub 
public void update() { 
f ( subject .getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ) 
observerState 
"This is the beginning of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
Åub1ic void notify0bservers() { 
or (Observer obj 
observers) 
obj . update(); 
new 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ÅOb1ic void detach (Observer obj) { 
observers . remove(obj ) 
public void notify0bservers() { 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Observer obj 
*alt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:57 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteoöserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
new 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public Optimist( ConcreteSubject sub ) 
super( sub 
ublic void update() 
observerState 
"Great! It's time to go green. 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Apple, take my money! " 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.getState 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
public String getState() { 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
81ueJ: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Optimist - observer 
Continue 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public Optimist( ConcreteSubject sub ) 
super( sub 
ublic void update() 
observerState 
"Great! It's time to go green. 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Apple, take my money! " 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:58 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
bublic Optimist( ConcreteSubject sub ) 
super( sub 
ÅOb1ic void update() { 
if ( subject .getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ") 
observerState 
observerState 
Ise 
observerState 
"Great! It's time to go green. 
" Apple, 
take my money! " 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.getState 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 
ConcreteSubject - observer 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
public String getState() { 
setState(String status) { 
ublic void 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = null 
protected ConcreteSubject subject = <0bject reference> 
Local variables 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public Optimist( ConcreteSubject sub ) 
super( sub 
ublic void update() { 
if ( subject.getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ") 
observerState 
observerState 
Ise 
observerState 
"Great! It's time to go green. 
" Apple, 
take my money! " 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Optimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 'Apple, take my money!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
Optimist 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public Optimist( ConcreteSubject sub ) 
super( sub 
ublic void update() { 
if ( subject .getState() .equa1sIgnoreCase( "The Price of gas is at $5.00/ga1 ") 
observerState 
"Great! It's time to go green. 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Apple, take my money! " 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
*alt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 6:59 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ÅOb1ic void detach (Observer obj) { 
observers . remove(obj ) 
public void notify0bservers() { 
or (Observer obj 
observers) 
obj . update(); 
new 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 
ConcreteSubject - observer 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
Halt 
"The New iPad is out today" 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:00 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today") ; 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
bublic static void main( String [ ] args ) 
Cannot set breakpoint: no code in this line 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = "Not another iPad!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer ; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
ublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = "Not another iPad!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
ublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out . println( 
"Observer : 
+ this. getC1ass() .getName() + 
+ observerState ) , 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer. Observer Test.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
OWS 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 7:01 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 'Apple, take my money!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 

*Screen clipping taken: 6/28/2018 7:02 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
public Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 

*Screen clipping taken: 6/28/2018 7:02 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 'Apple, take my money!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 7:02 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer ; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
ublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
bOb1ic void showState() 
System . out . println( 
"Observer : 
+ this. getC1ass() .getName() + 
+ observerState ) ; 

*Screen clipping taken: 6/28/2018 7:02 PM*

Machine generated alternative text:
81ueJ: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
CO 

*Screen clipping taken: 6/28/2018 7:02 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteoöserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( "Hey, Its Friday! 'I); 
p. showState() ; 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 7:02 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "The New iPad is out today" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
*alt 
"Hey, Its Friday!" 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:03 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
public void setState(String status) { 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:03 PM*

Machine generated alternative text:
Blue]: Debu 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "Hey, Its Friday!" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
String status = 
Halt 
"Hey, Its Friday!" 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:03 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove(obj ) 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:03 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "Hey, Its Friday!" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Halt 
Step 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:03 PM*

Machine generated alternative text:
ConcreteS u b 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
public void notify0bservers() { 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:03 PM*

Machine generated alternative text:
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "Hey, Its Friday!" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 
Observer obj 
Halt 
<0bject reference> 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteoöserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
ublic String getState() { 
return subjectState 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
ublic void detach (Observer obj) { 
observers . remove( obj 
ublic void notify0bservers() { 
or (Observer obj 
observers) 
d 
new 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = "Not another iPad!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
Pessimist - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
Concreteobserver X 
Close 
ublic class Pessimist extends Concrete0bserver { 
public Pessimist( ConcreteSubject sub ) 
super( sub 
public void update() { 
observerState 
"This is the beginning 
Optimist X 
of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
Blue]: Debugger 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteSubject.getState 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
private String subjectState = "Hey, Its Friday!" 
private ArrayList<Observer> observers = <0bject reference> 
Local variables 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
ConcreteSub 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
private observers 
public String getState() { 
ublic void setState(String status) { 
subjectState 
status 
notify0bservers() ; 
ublic void attach (Observer obj) { 
observers . add (obj) 
Åub1ic void detach (Observer obj) { 
observers . remove(obj ) 
Åub1ic void notify0bservers() { 
or (Observer obj 
observers) 
obj .update(); 
new 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = "Not another iPad!" 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
*alt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Compile 
undo 
Copy 
Paste 
Pessimist X 
Findm 
Concreteobserver X 
Close 
ublic class Pessimist extends Concrete0bserver { 
bublic Pessimist( ConcreteSubject sub ) 
super( sub 
public void update() { 
observerState 
"This is the beginning 
Optimist X 
of the end of the world! " 
Ise if ( subject.getState() .equa1sIgnoreCase( "The New iPad is out today" 
observerState 
Ise 
observerState 
" Not another iPad! " 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.Pessimist.update 
observer.ConcreteSubject.notifyObservers 
observer.ConcreteSubject.setState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerStat 
protected ConcreteSubject sut 
Local variables 

*Screen clipping taken: 6/28/2018 7:04 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
bublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
ConcreteObserver - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
public Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out . println( 
"Observer : 
+ this. getC1ass() .getName() + 
+ observerState ) , 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (at breakpoint) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 
Continuf 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteoöserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s.setState( " Hey, Its Friday! 'v) ; 
p. showState() ; 
bublic static void main( String [ ] args ) 
Cannot set breakpoint: no code in this line 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:05 PM*

Machine generated alternative text:
ConcreteOb 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
public Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ConcreteObserver.showState 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
protected String observerState = 
protected ConcreteSubject subject = <0bject reference> 
Local variables 
Halt 
Step Into 
Continue 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
package observer ; 
Copy 
Paste 
ublic class Concrete0bserver 
implements Observer { 
protected String observerState; 
protected ConcreteSubject subject; 
ublic Concrete0bserver( ConcreteSubject theSubject ) 
this . subject 
theSubject 
ublic void update() { 
// do nothing 
ublic void showState() 
System . out . println( 
" Observer: 
+ this .getC1ass() .getName() + 
+ observerState ) , 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ObserverTest.runTest 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
TheEconomy s = <0bject reference> 
Pessimist p = <0bject reference> 
Optimist o = <0bject reference> 
Halt 
Step Into 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
o. showState() ; 
bublic static void main( String [ ] args ) 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
main (stopped) 
Call Sequence 
observer.ObserverTest.main 
Static variables 
Instance variables 
Local variables 
StringO args = <0bject reference> 
Observer Test t = <0bject reference> 
Halt 
Step Into 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
o server. 
e conomy s 
observer . Pessimist p 
observer . Optimist o 
s. attach(p) ; 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
ublic static void main( String [ ] args ) 
serv es 
t. runTest() 
Cannot set breakpoint: no code in this line 
ser r est 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
Blue]: Debug 
Options 
Threads 
Call Sequence 
Thread is running. 
Threads must be stopped to view details. 
Static variables 
Instance variables 
Local variables 
Halt 
Step 
Step Into 
Continue 
Terminate 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
ObserverTest - 
Class Edit Tools Options 
ObserverTest X TheEconomy X ConcreteSubject X 
Pessimist X 
Findm 
new o 
Concreteobserver X 
Close 
Optimist X 
Compile 
undo 
Copy 
Paste 
public void runTest() 
o server. 
e conomy 
observer . Pessimist p 
observer . Optimist o 
s.attach(p); 
s. attach(o) ; 
server . 
e conomy 
new observer .Pessimist(s) ; 
new observer . Optimist(s) ; 
s.setState( "The New iPad is out today"); 
p. showState() ; 
o. showState() ; 
s. setState( " Hey, Its Friday! ; 
p. showState() ; 
bublic static void main( String [ ] args ) 
Cannot set breakpoint: no code in this line 

*Screen clipping taken: 6/28/2018 7:06 PM*

Machine generated alternative text:
Blue]: Terminal Window - observer 
Options 
Observer: 
observer . Pessimist 
Observer : 
observer . Optimist 
Observer : 
observer . Pessimist 
Observer: 
observer . Optimist 
Observer: 
observer . Pessimist 
Observer : 
observer . Optimist 
Observer : 
observer . Pessimist 
Observer : 
observer . Optimist 
Not another iPad! 
Apple, take my money! 
Not another iPad! 
Apple, take my money! 

*Screen clipping taken: 6/28/2018 7:07 PM*

*Screen clipping taken: 6/28/2018 6:17 PM*