



1 s.attach(p)




```
s.attach(p);
```

2 s.attach(o)




```
s.attach(o);
```

3 s.setState("New ipad.....")



```
s.setState("The New iPad is out today");
```

3.1 p.update()

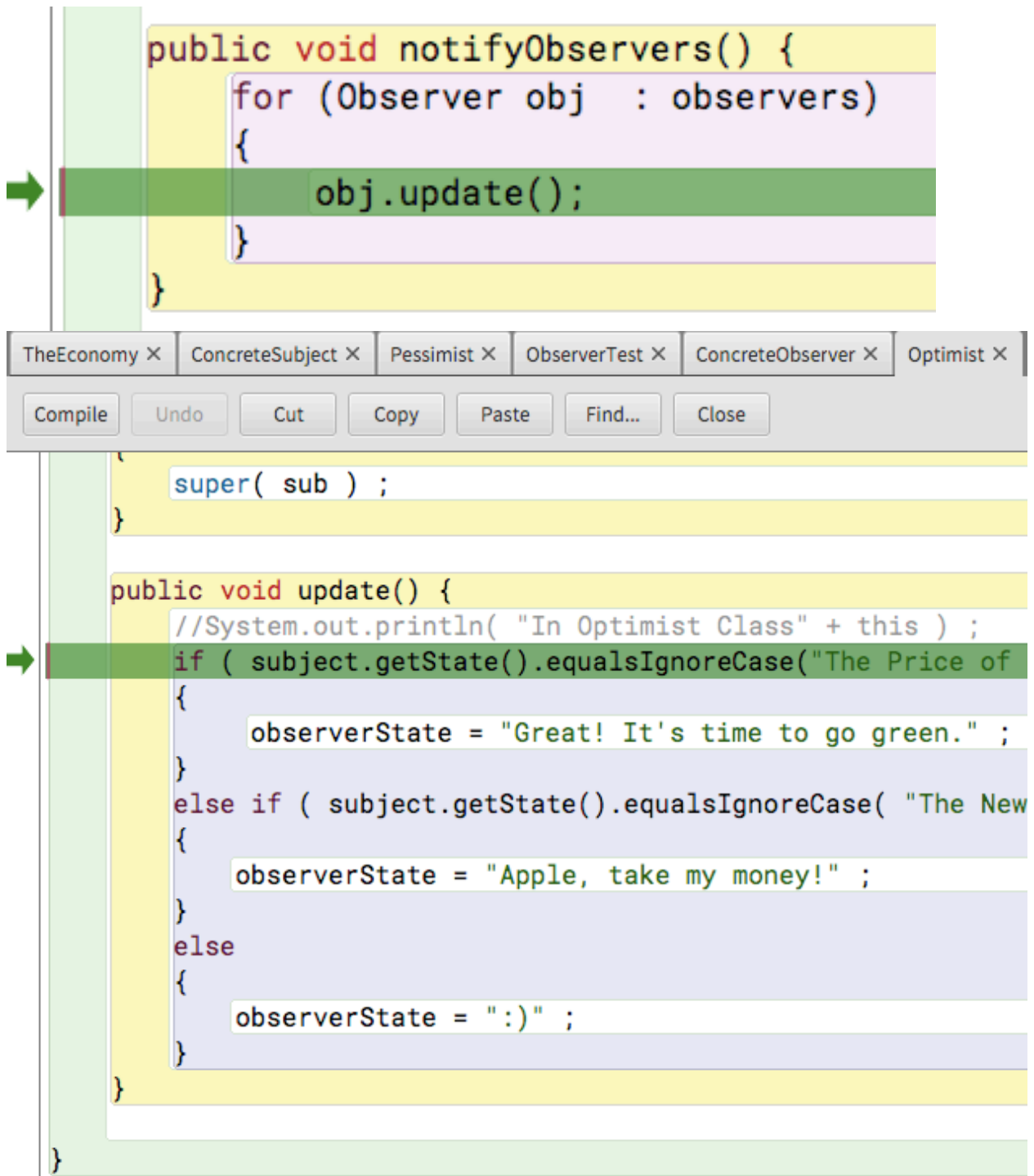


```
public void notifyObservers() {  
    for (Observer obj : observers)  
    {  
        obj.update();  
    }  
}
```

TheEconomy ×	ConcreteSubject ×	Pessimist ×	ObserverTest ×	ConcreteObserver ×	Optimist ×	
Compile	Undo	Cut	Copy	Paste	Find...	Close

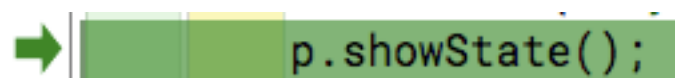
```
public Pessimist( ConcreteSubject sub )  
{  
    super( sub ) ;  
}  
  
public void update() {  
    if ( subject.getState().equalsIgnoreCase("The Price of g  
{  
        observerState = "This is the beginning of the end of  
    }  
    else if ( subject.getState().equalsIgnoreCase( "The New  
{  
        observerState = "Not another iPad!" ;  
    }  
}
```

3.2 o.update()




```
public void notifyObservers() {  
    for (Observer obj : observers)  
    {  
        obj.update();  
    }  
}  
  
super( sub ) ;  
  
public void update() {  
    //System.out.println( "In Optimist Class" + this ) ;  
    if ( subject.getState().equalsIgnoreCase("The Price of  
    {  
        observerState = "Great! It's time to go green." ;  
    }  
    else if ( subject.getState().equalsIgnoreCase( "The New  
    {  
        observerState = "Apple, take my money!" ;  
    }  
    else  
    {  
        observerState = " :) " ;  
    }  
}
```

4 p.showState()




```
p.showState();
```

5 o.showState()




```
o.showState();
```

6 s.setState("Hey, Its Friday!")

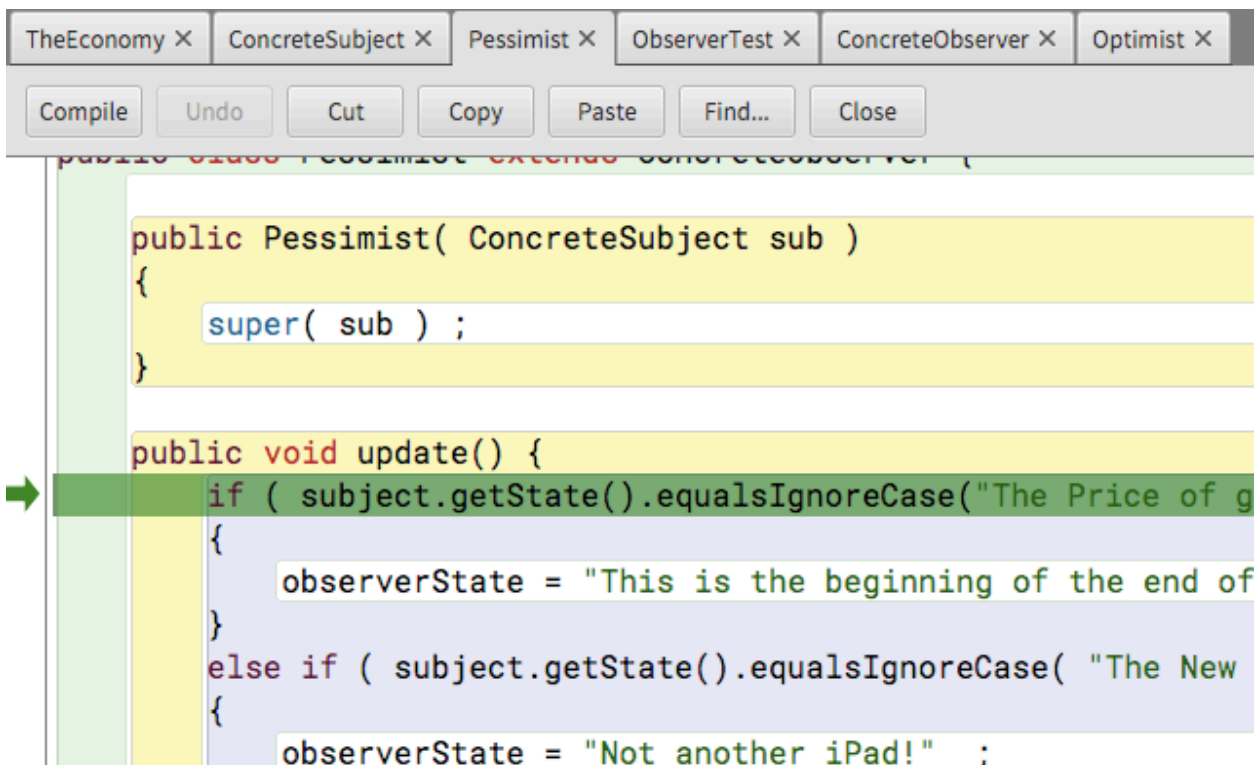


```
s.setState("Hey, Its Friday!");
```

6.1 p.update()

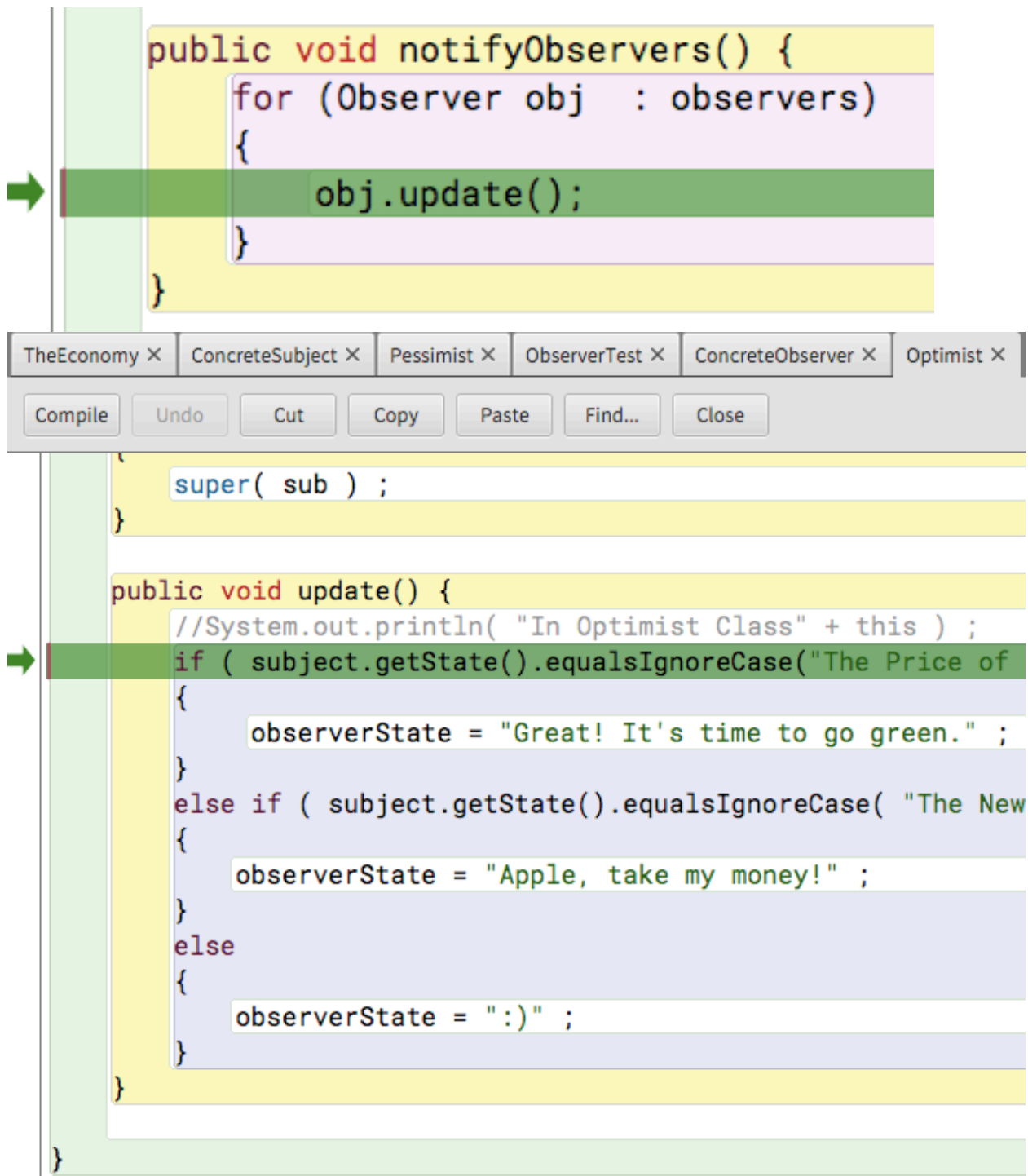


```
public void notifyObservers() {  
    for (Observer obj : observers)  
    {  
        obj.update();  
    }  
}
```



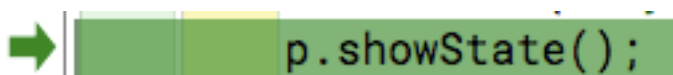
```
TheEconomy × ConcreteSubject × Pessimist × ObserverTest × ConcreteObserver × Optimist ×  
Compile Undo Cut Copy Paste Find... Close  
public class Pessimist extends ConcreteObserver {  
    public Pessimist( ConcreteSubject sub )  
    {  
        super( sub ) ;  
    }  
    public void update() {  
        if ( subject.getState().equalsIgnoreCase("The Price of g  
        {  
            observerState = "This is the beginning of the end of  
        }  
        else if ( subject.getState().equalsIgnoreCase( "The New  
        {  
            observerState = "Not another iPad!" ;  
        }  
    }  
}
```

6.2 o.update()




```
public void notifyObservers() {  
    for (Observer obj : observers)  
    {  
        obj.update();  
    }  
}  
  
super( sub ) ;  
  
public void update() {  
    //System.out.println( "In Optimist Class" + this ) ;  
    if ( subject.getState().equalsIgnoreCase("The Price of  
    {  
        observerState = "Great! It's time to go green." ;  
    }  
    else if ( subject.getState().equalsIgnoreCase( "The New  
    {  
        observerState = "Apple, take my money!" ;  
    }  
    else  
    {  
        observerState = " :) " ;  
    }  
}
```

7 p.showState()



```
p.showState();
```

8 o.showState()



```
o.showState();
```