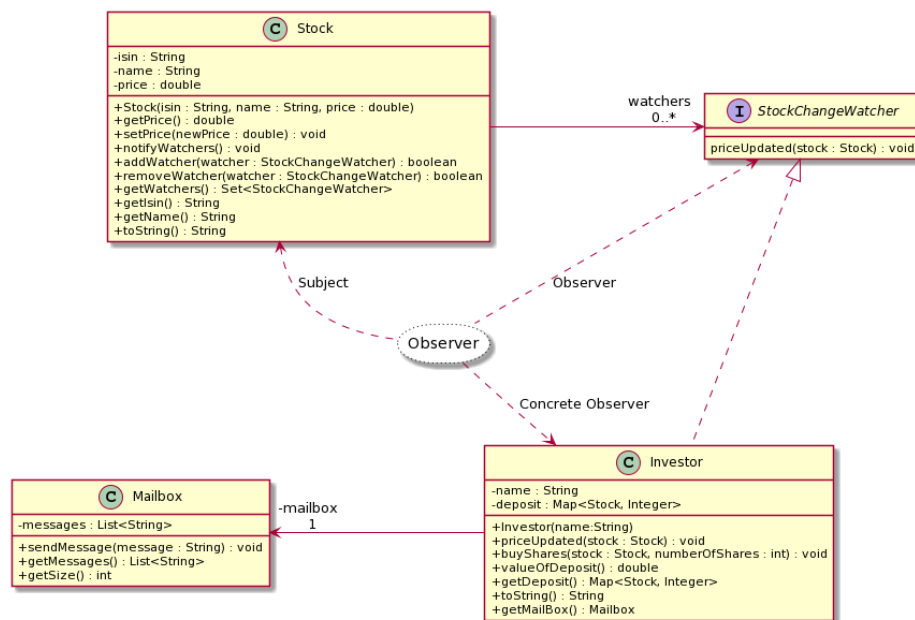


Softwaretechnologie, Übung 8

HENRY HAUSTEIN

Aufgabe 1

- (a) Observer
- (b) UML-Diagramm



Aufgabe 2

```
1  /**
2   * Aktien werden durch eine feste ISIN (International Securities
3   * Identification
4   * Number) und einen festen Namen beschrieben und koennen ihren
5   * Preis aendern.
6   * Durch einen Benachrichtigungsmechanismus ({@link
7   * StockChangeWatcher}) koennen
8   * interessierte Parteien ueber Preisverfaelle informiert werden.
9   */
10 public class Stock {
11     private final String isin;
```

```

9     private final String name;
10    private double price;
11    private double oldPrice;
12
13    private final Set<StockChangeWatcher> watchers = new HashSet<>()
        ;
14
15    public Stock(String isin, String name, double price) {
16        this.isin = isin;
17        this.name = name;
18        this.price = price;
19        this.oldPrice = price;
20    }
21
22    public double getPrice() {
23        return price;
24    }
25
26    public void setPrice(double newPrice) {
27        if (newPrice < 0.0) {
28            throw new IllegalArgumentException("newPrice darf nicht
                negativ sein");
29        } else {
30            this.oldPrice = price;
31            this.price = newPrice;
32            notifyWatchers();
33        }
34    }
35
36    private void notifyWatchers() {
37        for (StockChangeWatcher w : watchers) {
38            // Preisaenderung negativ
39            if (price < oldPrice) {
40                w.priceUpdated(this);
41            }
42        }
43    }
44
45    public boolean addWatcher(StockChangeWatcher watcher) {
46        return watchers.add(watcher);
47    }
48
49    public boolean removeWatcher(StockChangeWatcher watcher) {
50        return watchers.remove(watcher);
51    }
52
53    public Set<StockChangeWatcher> getWatchers() {
54        return Collections.unmodifiableSet(watchers);
55    }
56
57    public String getIsin() {

```

```

58     return isin;
59 }
60
61 public String getName() {
62     return name;
63 }
64
65 public String toString() {
66     return name + " (" + isin + ")";
67 }
68 }

```

Aufgabe 3

(a) Testfalltabelle

Testfall	Aufruf	Eingabe	Erwartetes Ergebnis
ON1	stock.setPrice	0.0	investor wird benachrichtigt
ON2	stock.setPrice	-1.0	IllegalArgumentException wird in stock geworfen, investor wird nicht benachrichtigt
ON3	stock.setPrice	500	investor wird nicht benachrichtigt
ON4	stock.setPrice	15.37	investor wird nicht benachrichtigt
ON5	stock.removeWatcher(investor); stock.setPrice	10	investor wird nicht benachrichtigt
ON6	stock.setPrice	10	investor wird benachrichtigt

(b) Implementierung

```

1  import static org.junit.jupiter.api.Assertions.*;
2  import org.junit.jupiter.api.*;
3
4  /**
5   * Testet die Interaktion von {@link Stock}- und {@link
6   *   Investor}-Objekten (Test
7   *   von Objektnetzen).
8   */
9  class ChangeNotificationTest {
10     private Investor investor;
11     private Mailbox mailbox;
12     private Stock stock;
13
14     @BeforeEach
15     void setUp() {
16         investor = new Investor("Erika Musterfrau");

```

```

17     mailbox = investor.getMailbox();
18     stock = new Stock("US36467W1099", "GME", 15.37);
19     stock.addWatcher(investor);
20 }
21
22 @Test
23 void Test1() {
24     stock.setPrice(0.0);
25     assertEquals(1, mailbox.getSize());
26     assertEquals("Neuer Wert von GME (US36467W1099): $0.0",
27         mailbox.getMessages().get(0));
28 }
29
30 @Test
31 void Test2() {
32     try {
33         stock.setPrice(-1.0);
34         fail("setPrice() sollte keine negativen Werte
35             akzeptieren");
36     } catch (IllegalArgumentException e) {
37         System.out.println("GG");
38     }
39     assertEquals(0, mailbox.getSize());
40 }
41
42 @Test
43 void Test3() {
44     stock.setPrice(500.0);
45     assertEquals(0, mailbox.getSize());
46 }
47
48 @Test
49 void Test4() {
50     stock.setPrice(15.37);
51     assertEquals(0, mailbox.getSize());
52 }
53
54 @Test
55 void Test5() {
56     stock.removeWatcher(investor);
57     stock.setPrice(10.0);
58     assertEquals(0, mailbox.getSize());
59 }
60
61 @Test
62 void Test6() {
63     stock.setPrice(10.0);
64     assertEquals(1, mailbox.getSize());
65 }

```

Ablauf Testfall ON6

