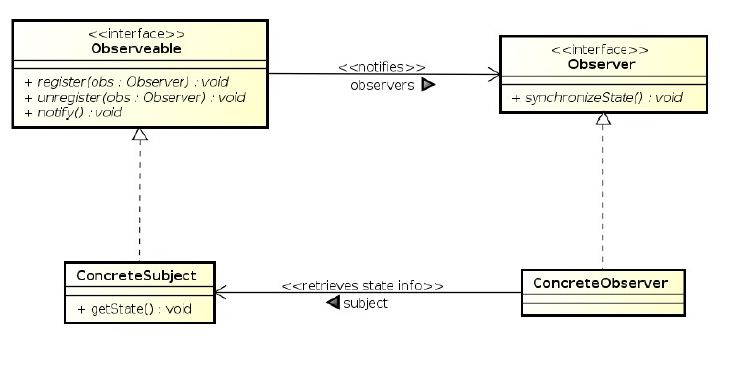
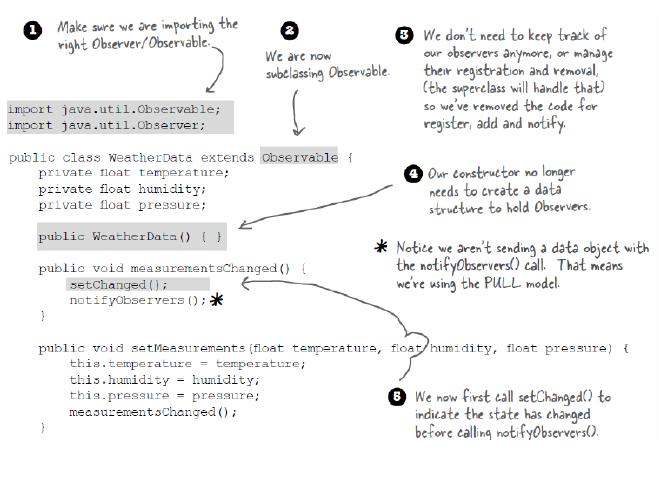
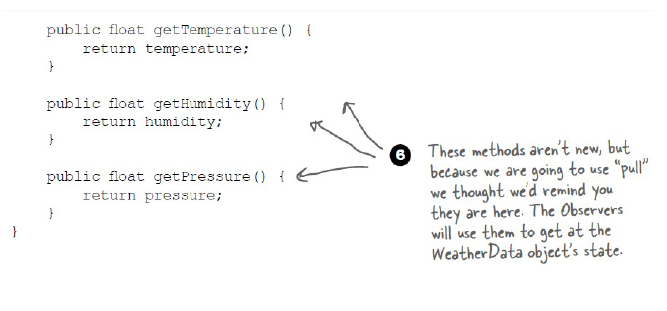
|  |
| --- |
| TEIL A |

|  |
| --- |
| Frage 1 |

**Observer Pattern**

Das Observer Pattern definiert eine „Eins-zu-viele-Abhängigkeit “zwischen Objekten in der Art, dass alle abhängigen Objekte benachrichtigt werden, wenn sich der Zustand des einen Objekts ändert.



**Strebe für lose gekoppelte Designs zwischen Objekten, die interagieren.**

Locker gebunden = Interaktion mit wenig Detailwissen

Lose gekoppelte Designs ermöglichen das Bauen von flexiblen OO Systemen, die die Änderung bewältigen können, weil sie die wechselseitige Abhängigkeit minimieren.

**Observer Pattern: Lockere Kopplung zwischen Subjekt und Beobachter:**

• Subjekt kennt von einem Beobachter nur die Beobachter-Schnittstelle

• Subjekt muss für neue Beobachter nicht verändert werden

• Subjekt und Beobachter sind unabhängig verwendbar

**Funktionsweise:**

• Beobachter-Klassen implementieren java.util.Observer

• Die Subjekt-Klasse erweitert java.util.Observable

• Nachrichten schicken: ♦ setChanged() aufrufen

♦ notifyObservers() oder notifyObservers(Object arg)

• Benachrichtigung erhalten:

♦ Update(Observable o, Object arg) implementieren

|  |
| --- |
| Frage 2 |

Man könnte verwenden:  
- BlockingQueue  
- ConcurrentMap  
- ConcurrentNavigationMap

Wenn Arraylists verwendet werden:

Zugriffsfehler können dabei auftreten, wenn mehrere Threads auf einen Datensatz zugreifen

wollen.

|  |
| --- |
| Frage 3 |

1) log auf synchronized setzten:

Public static synchronized void log() {}

2) Konstruktor von logger auf private setzten.

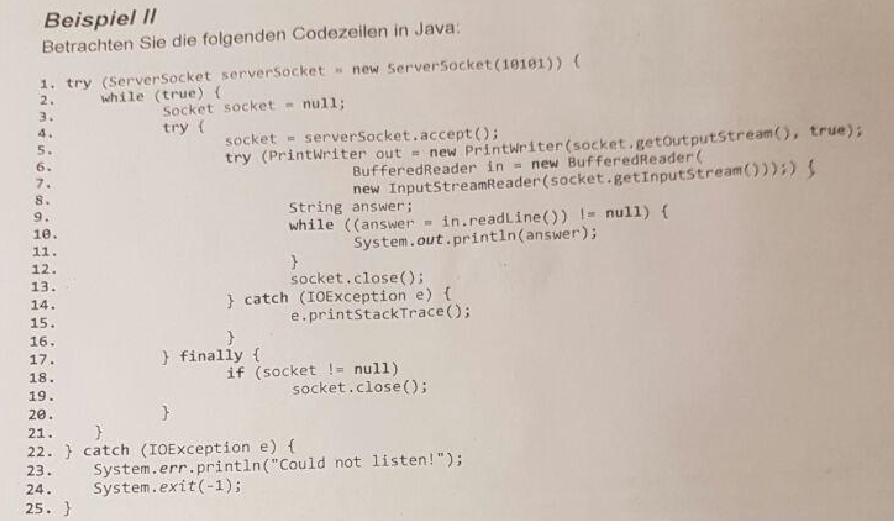
Eine instanz von logger in logger speichern (statisch)

Eine Statische Methode erstellen um Instanz zu holen: getInstanz(): logger

|  |
| --- |
| Frage 4 |

Nach welcher Zeile wäre der richtige Zeitpunkt, um einen neuen Thread zu erzeugen?

(Erweitert) Geben Sie den Java-Code an, den Sie hier einfügen würden, um diesen Thread zu erzeugen und zu starten!



Nach Zeile 5

Thread tnew = new Thread(MyRunImpl(socket));

Tnew.start();

|  |
| --- |
| Frage 5 |

