

## SE 2 Klausur

Vorname: \_\_\_\_\_

Nachname: \_\_\_\_\_

Matr.-Nr.: \_\_\_\_\_

Dies ist mein letzter Prüfungsversuch

- Bitte schalten Sie alle elektronischen Geräte aus.
- Es sind keine Hilfsmittel zugelassen.
- Die Klausur dauert 60 Minuten und es gibt insgesamt 60 Punkte.
- Bitte geben Sie die Klausur komplett und geheftet wieder ab.
- Bitte schreiben Sie mit schwarzem oder blauem Stift. Nicht mit Bleistift.
- Schreiben Sie bei Platzmangel auf der Rückseite weiter und markieren Sie dies.
- Bitte schreiben Sie deutlich! Nur Lesbares kann bewertet werden.

---

Folgende Tabelle ist für die Korrektur vorgesehen: Bitte nicht beschriften!

Aufgabe	1	2	3	4	5	6	7	8	9	10	11	12	insgesamt
Punkte	4	3	4	5	3	5	7	8	9	5	3	4	60
erreicht	4	3	3,5	5	1	4	7	8	9	5	3	3	55,5

Note: \_\_\_\_\_

1,3



**Aufgabe 1**

4  
4 P.

Tragen Sie die Namen von acht git Befehlen in die Kästchen ein:

add ✓	commit ✓
push ✓	pull ✓
branch ✓	merge ✓
status ✓	fetch ✓

**Aufgabe 2**

3  
3 P.

Diskutieren Sie (in ganzen Sätzen) die Aussage:  
„Testen zeigt die Abwesenheit von Fehlern“.

Testen kann nicht die Abwesenheit von Fehlern zeigen, sondern nur die Anwesenheit von Fehlern, da immer Fehler vorhanden sein können, aber unentdeckt bleiben. ✓"

**Aufgabe 3**

3,5  
4 P.

Erklären Sie die Begriffe „Kohäsion“ und „Kopplung“ mit ganzen Sätzen.

**Kohäsion:**

Ist ein Maß für die innere Abhängigkeit einer Systemkomponente. ✓

**Kopplung:**

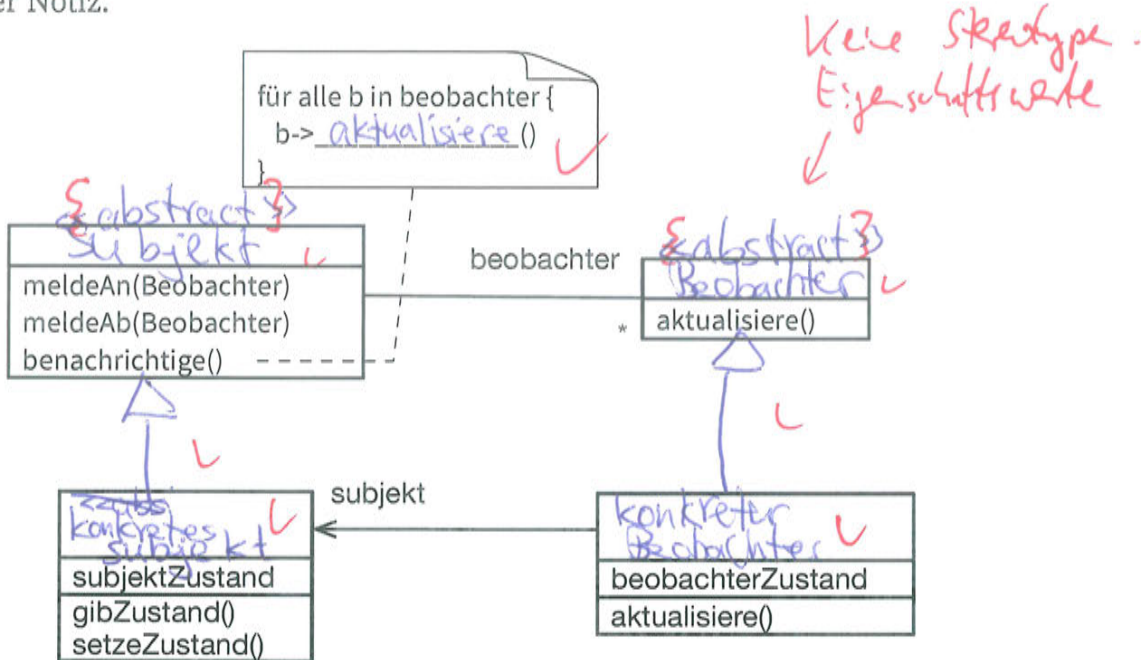
ist ein Maß für die Stärke der Abhängigkeit für mehreren Systemkomponenten. Zwischen +1 1/2



### Aufgabe 4

5  
5 P.

Hier ist das Entwurfsmuster „Beobachter“ abgebildet. Vervollständigen Sie das Diagramm hinsichtlich fehlender Klassennamen, Assoziationen und dem Inhalt der Notiz.



### Aufgabe 5

1  
3 P.

In Bezug auf Kodierungsrichtlinien: Was bedeuten folgende Abkürzungen? Schreiben Sie den vollständigen Namen oder erklären Sie die Bedeutung kurz:

DRY Don't repeat yourself ✓

POLA Please over loop anybody X

SRP Sea reaper X



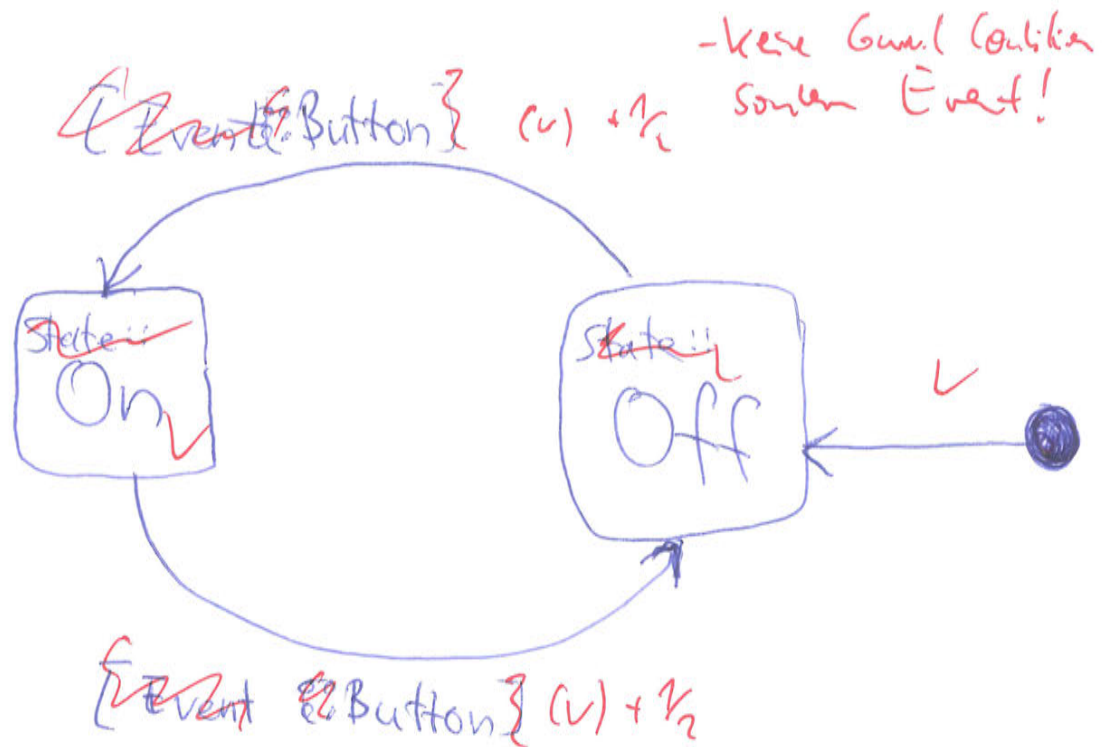
### Aufgabe 6

4  
5 P.

Zeichnen Sie ein Zustandsdiagramm für folgenden C++ Quelltext:

```

1  enum class State { On, Off };
2  enum class Event { Button };
3
4  class StateMachine {
5      State current_ = State::Off;
6  public:
7      State current_state() const { return current_; }
8      void trigger(Event e) {
9          switch (current_) {
10         case State::On:
11             if (e == Event::Button) current_ = State::Off;
12             break;
13         case State::Off:
14             if (e == Event::Button) current_ = State::On;
15             break;
16         }
17     }
18 };
    
```



Präfixe generell ok, besser ohne



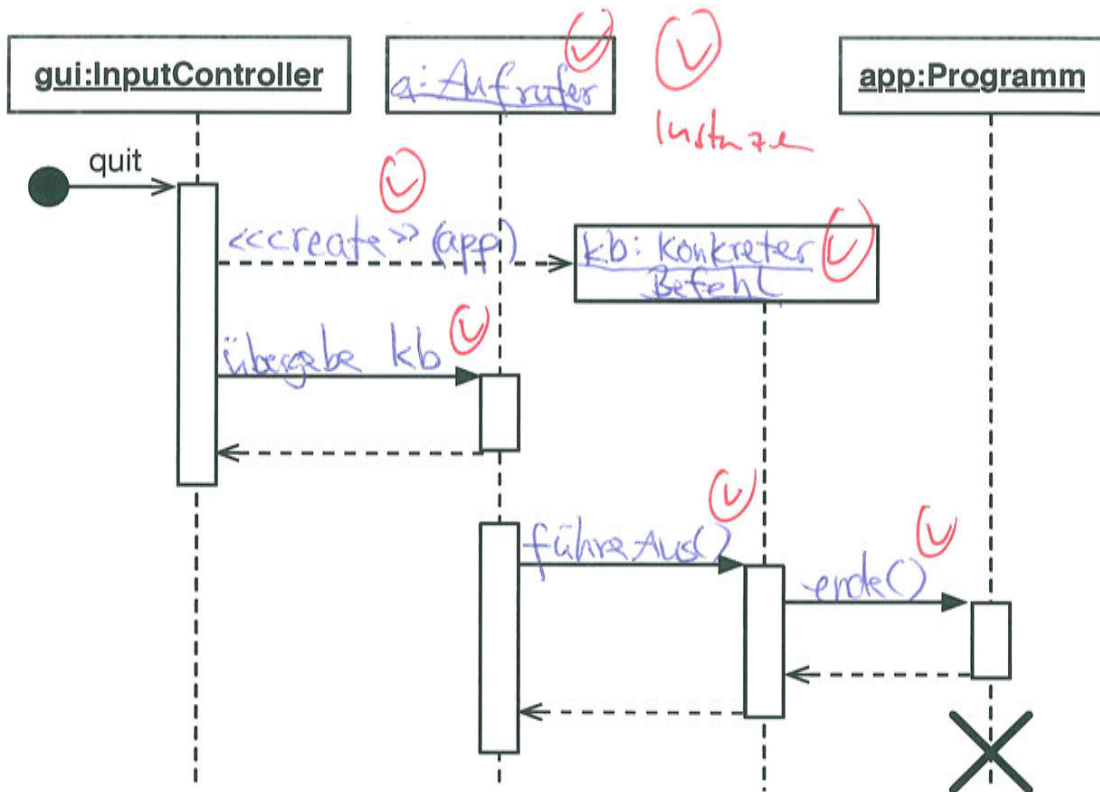
### Aufgabe 7

Gegeben ist folgende Situation:

7  
7 P.



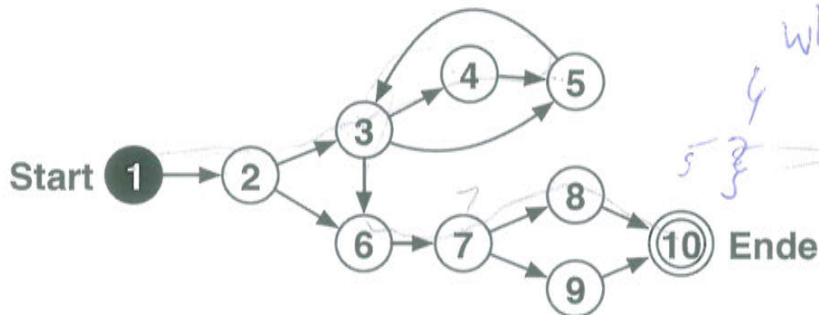
Entkoppeln Sie die Klassen **InputController** und **Programm** mittels des **Befehl** Entwurfsmusters. Vervollständigen Sie dazu in dem untenstehenden Sequenzdiagramm die Namen der Instanzen, sowie die Beschriftung der Aufrufe.





**Aufgabe 8**

8  
8 P.



while() {  
int i=0;  
int a=5;  
}

Entwerfen Sie für den oben abgebildeten Kontrollflussgraphen abstrakte Testfälle nach den folgenden vier Überdeckungskriterien.

**Abstrakte Testfälle für 100% Anweisungsüberdeckung:**

~~{1,2,3,4,5,3,6,7,8,10}~~, ✓  
{1,2,6,7,8,10}

**Abstrakte Testfälle für 100% Entscheidungsüberdeckung:**

{1,2,3,4,5,3,6,7,8,10}, {1,2,3,5,3,6,7,8,10},  
{1,2,6,7,9,10} ✓

**Abstrakte Testfälle für 100% Grenze-Inneres Überdeckung:**

~~{1,2,6,7,9,10}~~, {1,2,3,4,5,3,6,7,8,10}, ✓  
{1,2,3,4,5,3,4,5,3,6,7,8,10}, ~~{1,2,3,5,3,6,7,9,10}~~

**Abstrakte Testfälle für 100% Pfadüberdeckung:**

Unmöglich! ✓



**Aufgabe 9**

9  
9 P.

Sie sollen ein Programm zur Festlegung der Umweltfreundlichkeit von Kraftfahrzeugen auf korrekte Funktion mit Hilfe einer Äquivalenzklassenbildung testen. Die Funktion zur Festlegung der Umweltfreundlichkeit verwendet drei Parameter. Wählen Sie die richtige Anzahl an gültigen und ungültigen Äquivalenzklassen für jeden der folgenden Parameter:

1. Markiert ob das Fahrzeug ein Elektroauto oder ein Verbrenner ist →enum
2. Die Anzahl der Achsen im Bereich 2 bis 5
3. Eine Unterscheidung in PKW, LKW oder Bus

**Geben Sie für die drei Parameter die Äquivalenzklassenaufstellung an:**

1.  $g\ddot{A}K_1: x = \{ \text{Elektroauto} \}$   
 $g\ddot{A}K_2: x = \{ \text{Verbrenner} \}$   
 $u\ddot{A}K_1: x \notin \{ \text{Elektroauto, Verbrenner} \}$

2.  $g\ddot{A}K_3: \text{2} \leq a \leq 5 \rightarrow 2$   
 $u\ddot{A}K_2: a = \{1\}$   
 $u\ddot{A}K_3: a = \{6\}$

3.  $g\ddot{A}K_4: t = \{ \text{PKW} \}$   
 $g\ddot{A}K_5: t = \{ \text{LKW} \}$   
 $g\ddot{A}K_6: t = \{ \text{Bus} \}$   
 $u\ddot{A}K_4: t \notin \{ \text{PKW, LKW, Bus} \}$



|||||



### Aufgabe 10

5  
5 P.

Entscheiden Sie welche der folgenden Aussagen jeweils für einen Publish-Subscribe Ansatz oder das Beobachter Muster passen, in dem Sie entweder einen Haken (*passt*) oder ein Kreuz (*passt nicht*) in das entsprechende Tabellenfeld schreiben. Aussagen können für eines, keines oder beide der beiden Ansätze passen.

	Pub-Sub	Beobachter
Erlaubt eine Entkopplung von Klassen	✓ -	✓ -
Bietet eine Form der Gruppenkommunikation über <i>Topics</i> an	✓ -	X -
Sender und Empfänger einer Nachricht kennen sich nicht gegenseitig	✓ -	✓ -
Ermöglicht das dynamische Hinzufügen von neuem Verhalten zu einem Objekt	X -	X -
Unterstützt eine 1-zu-n Kommunikation	✓ -	✓ -

### Aufgabe 11

3  
3 P.

Was ist der Unterschied zwischen statischen und dynamischen Tests?

Bitte antworten Sie in ganzen Sätzen.

Bei statischen Tests testet man die Testobjekte "an sich", sie werden also nicht aufgeführt. Bei dynamischen Tests testet man die Testobjekte zur Laufzeit und es wird eine Laufzeitumgebung benötigt.



**Aufgabe 12**

3  
4 P.

Ein Tester kommt in eine Kneipe. Bestellt ein Bier. Bestellt 0 Bier. Bestellt 999999 Biere. Bestellt eine Eidechse. Bestellt -1 Bier. Bestellt ein sckfjasbnfö-so...

Erklären Sie mit ganzen Sätzen was das mit dem Entwurf von konkreten Testfällen zu tun hat. Handelt es sich um einen guten Tester?

Der Tester Es hat mit dem Entwurf von konkreten Testfällen zu tun, da der Tester Tests mit konkreten Werten bildet.

Bei einer Kneipe kann man Bier in einem bestimmten Wertebereich testen, z.B. 1 bis 500. Man könnte diskutieren ob die 0 auch dazugehören sollte oder nicht. > Deshalb ist es gut, dass der Tester auch diesen Fall abdeckt.

Zu testen -1, 0, 1 und 999999 Biere als Bestellung was sinnvoll ist.

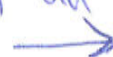
~~Je nach dem ob 0 ein gültiger Wert Bier eine gültige Bestellung sind, dann hätte man die 0 auch weglassen können, da die 1 schon eine gültige Zahl darstellt.~~

~~0 Bier~~

Dem Fall mit 0 Bier hätte man weglassen können, da man mit -1 Bier sowieso schon einen ungültigen Wert hat.

Die Fälle mit den Strings ("Eidechse" und "sck...") hätte der Tester weglassen sollen, da diese einem

0 "wird er kein testen!"



völlig anderem Datentypen angehören. \*

~~Bestand ist~~ e  
Ich komme zu dem Schluss, dass der  
Tester ok ist, aber nicht gut.

\* Es kann sinnvoll sein, nicht jede  
Sprache ist statisch getypt.

---

~~...~~

~~...~~

~~...~~

~~...~~