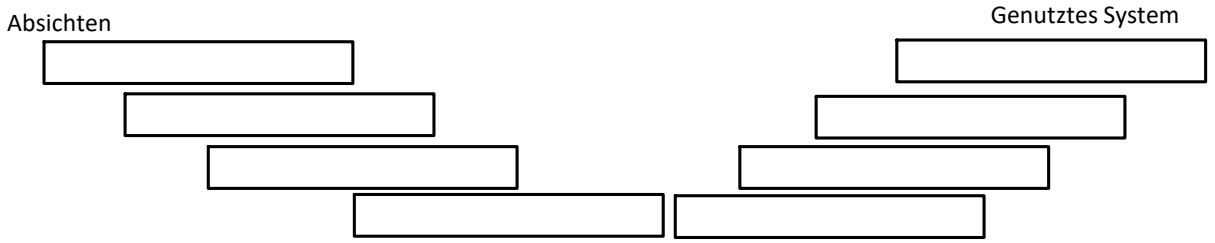
Altklausur 18.09.2014 (Mitschrift aus Übung)

Donnerstag, 19. Juli 2018 10:02

Aufgabe1: Software Lebenszyklus

Ergänzen Sie das V-Modell



Aufgabe 2: Kopplung:

Bestimmen und begründen Sie welche Kopplungsart vorliegt

a)

```
Class A {
  public static
  List<List<Integer>> heaps=
    new ArrayList<List<List<Integer>>();

  static void populate(){
    for( int i=0; i<100; i++){
      List<Integer> heap=
        new ArrayList<Integer>();
      int m= (int) (1000+Math.random());
      for (int j=0; j<m; j++)
        heap.add((int) (3000+Math.random()));

      heaps.add(heap);
}}}
```

```
Class B{
  public static void sort(){
    A.populate();
    for (List<Integer> h: A.heaps){
      if(h.size() > 1000)
        Collections.sort(h);
      else
        A.heaps.remove(h);
}}
```

b)

```
Class A{
Static int op( String opName, int a, int b){
  if(opName.equals("MAX"))
    return(a<=b)?b:a;
  else if (opName.equals("MIN"))
    return(a<=b)?a:b;
  else if (opName.equals("ABS"))
    return (a<0)?-a:a;
  else
    throw new IllegalArgumentException();
```

```
 Class B{
   public static void main(String[] args){
     try{
       A.op("MAX", System.in.read(),200);
       A.op("ABS",System.in.read(),100);
       A.op("MIN",System.in.read(),100);
     }catch (IOException e){}
 }}
```

c)

```
Class A{
  public interface Mitarbeiter{
    String name;
    double gehalt;
  }

  public static double
  gehaltsDurchschnitt(List<Double> gehaelter){
    double sum=0;
    for( double g : gehaelter) sum=sum+g;

    return gehaelter.size()>0?
sum/gehaelter.size():0;
}}
```

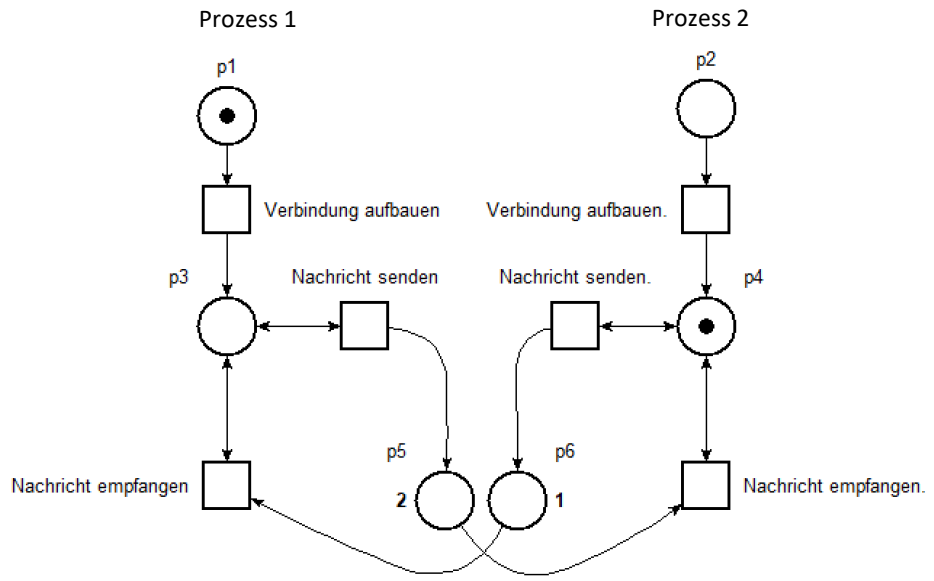
```
Class B{
  static void
  statistic(List<A.Mitarbeiter> list) {
  List<Double> g=new LinkedList<Double>();

  for( A.Mitarbeiter m: liste) g.add(m.gehalt);

  double durchschnitt=A.gehaltsDurchschnitt(g);

  for(A.Mitarbeiterm:liste)
    if(m.gehalt <durchschnitt)
      System.our.println(m.name);
}}
```

Augabe 3: Petri-Netze



Hinweis: Token auf den Plätzen p5 und p6 repräsentieren Nachrichten in dem Kommunikationskanal. Beachten Sie auch die zugehörigen Kapazitätsangaben.

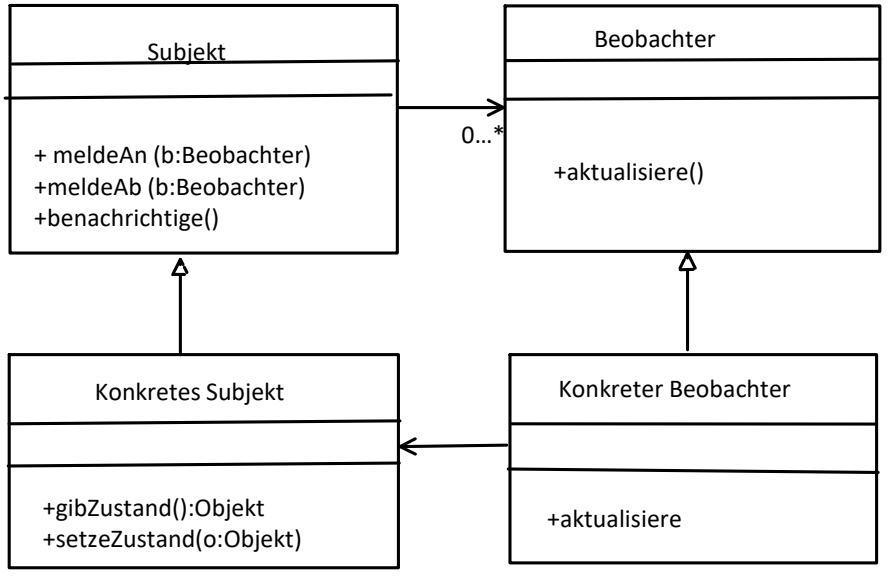
Teilaufgaben

a) Geben Sie den Erreichbarkeitsgraphen des Petri-Netzes an.
b) Erweitern Sie das Petri-Netz um Plätze, Transitionen, Kanten (außer Inhibitor-Kanten) bzw. Token so, dass zusätzlich folgende Eigenschaften erfüllt werden:

1. Jeder Prozess kann nach Aufbau einer Verbindung zum Kommunikationskanal diese Verbindung wieder abbauen. Nur bei Bestehen einer aufgebauten Verbindung können Nachrichten gesendet bzw. empfangen werden.
2. Die Prozesse sind nie gleichzeitig mit dem Kommunikationskanal verbunden.
3. Die Prozesse verbinden sich abwechselnd mit dem Kommunikationskanal, d.h. baut ein Prozess seine Verbindung zum Kommunikationskanal ab, so darf er erst dann eine neue Verbindung aufbauen, wenn der andere Prozess zuletzt mit dem Kanal verbunden war.
4. In dem Kommunikationskanal können sich bis zu maximal 2 Nachrichten zur gleichen Zeit befinden.

Aufgabe 4: Entwurfsmuster

a) Folgendes Klassendiagramm stellt die statische Sicht des Entwurfsmusters "Beobachter dar:



Stellen Sie anhand eines Sequenzdiagramms die dazugehörigen Objekt -Interaktionen für

folgenden Fall dar.

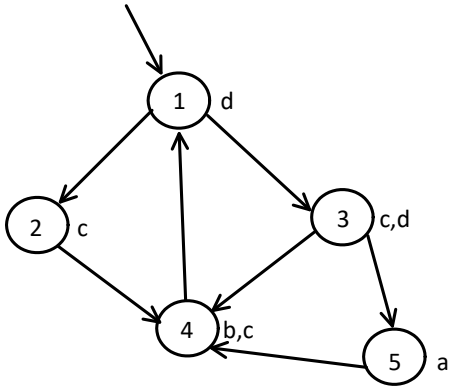
Konkrete Beobachter b1 und b2 melden sich beim konkreten Subjekt s an. Anschließend verändert b2 den Zustand von s.

b) Die Klasse Vertex bietet folgende Operationen zur Konstruktion von Graphen an
   - addNeighbour(v:Vertex)
   - removeNeighbour(v:Vertex)

   Während der Laufzeit soll festgelegt werden können, dass durch Bestätigung einer vorgegebenen Taste eine aus den obigen auszuwählenden Operationen auszuführen sei. Zeigen Sie anhand eines Klassendiagramms wie das Entwurfsmuster "Befehl" (eng. Command) zu diesem Zweck einsetzbar ist.
   Hinweis: Berücksichtigen Sie, dass sowohl addNeighbour als auch removeNeighbour parametrisiert sind!

Aufgabe5: Model Checking

Es seien a,b,c atomare Aussagen. Folgende Abbildung definierte eine CTL -Struktur, wobei an den Zuständen die jeweils gültigen atomaren Aussagen annotiert sind:



Geben Sie für jede der folgenden CTL-Formeln den jeweiligen Wahrheitswert im Startzustand an und begründen Sie Ihre Entscheidung.

a) AF(a∨b) ∧∨¬
b) AG EX c
c) EG(¬d ∨ AX c)
d) A (c∨d ∪ EX a

Lösung

Aufgabe 1

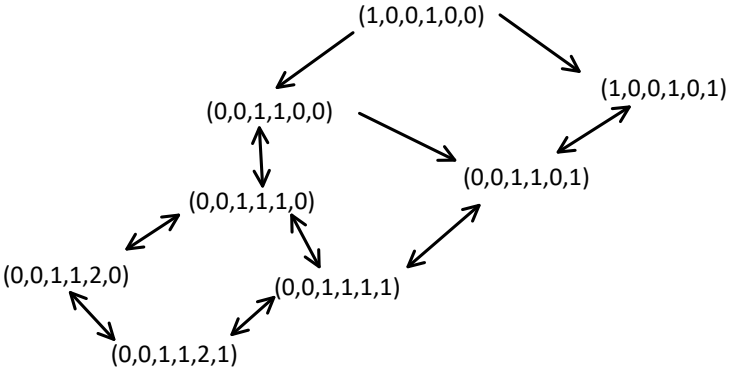
Absichten — Genutztes System
Anforderungsspezifikation — Installiertes System
Systementwurf — Integriertes System
SW-Grobentwurf — Integrierte SW-Module
SW-Feinentwurf — SW Module

Aufgabe2: Kopplung

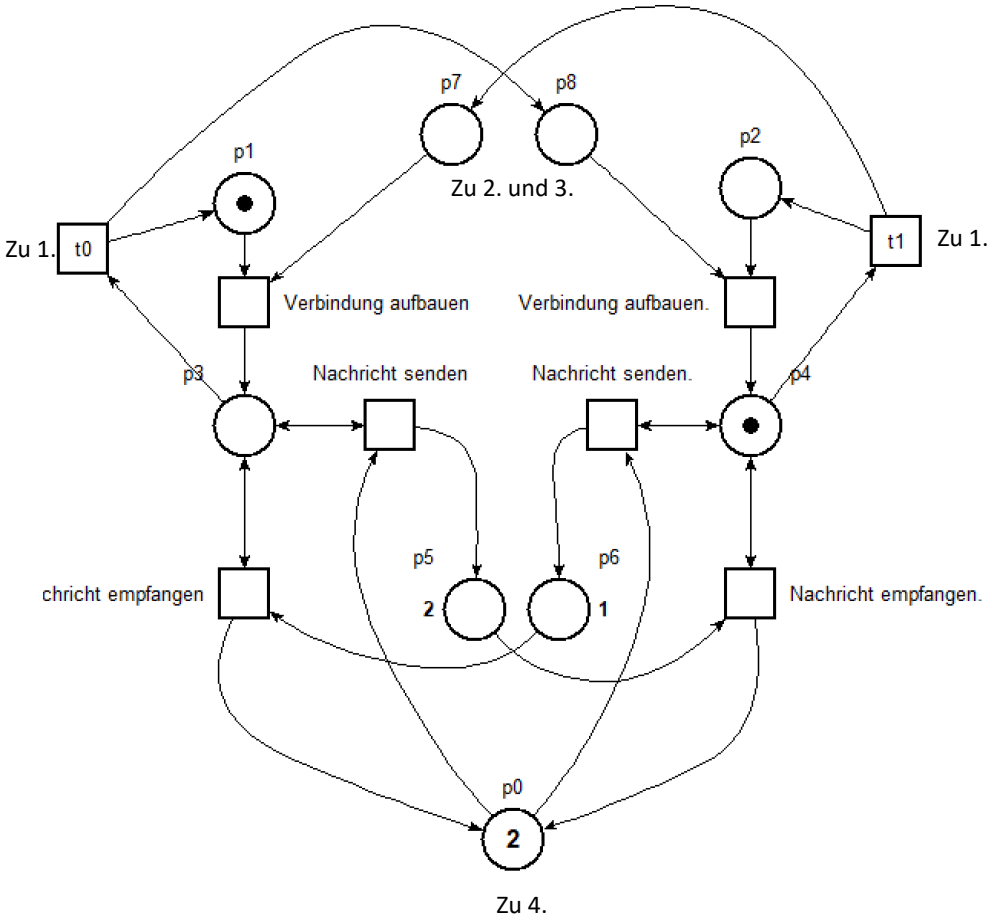
a)Global Kopplung (gemeinsame Datenstruktur Eisendschreiben)
b)Kontrollkopplung (Methode Main in Klasse B steuert mit dem ersten Parameter die Methode Op in der Klasse A)
c)Datenkopplung (alle Daten, die an die Methode Gehaltsdurchschnitt übergeben werden, werden benutzt

Aufgabe 3: Petri-Netze

a)

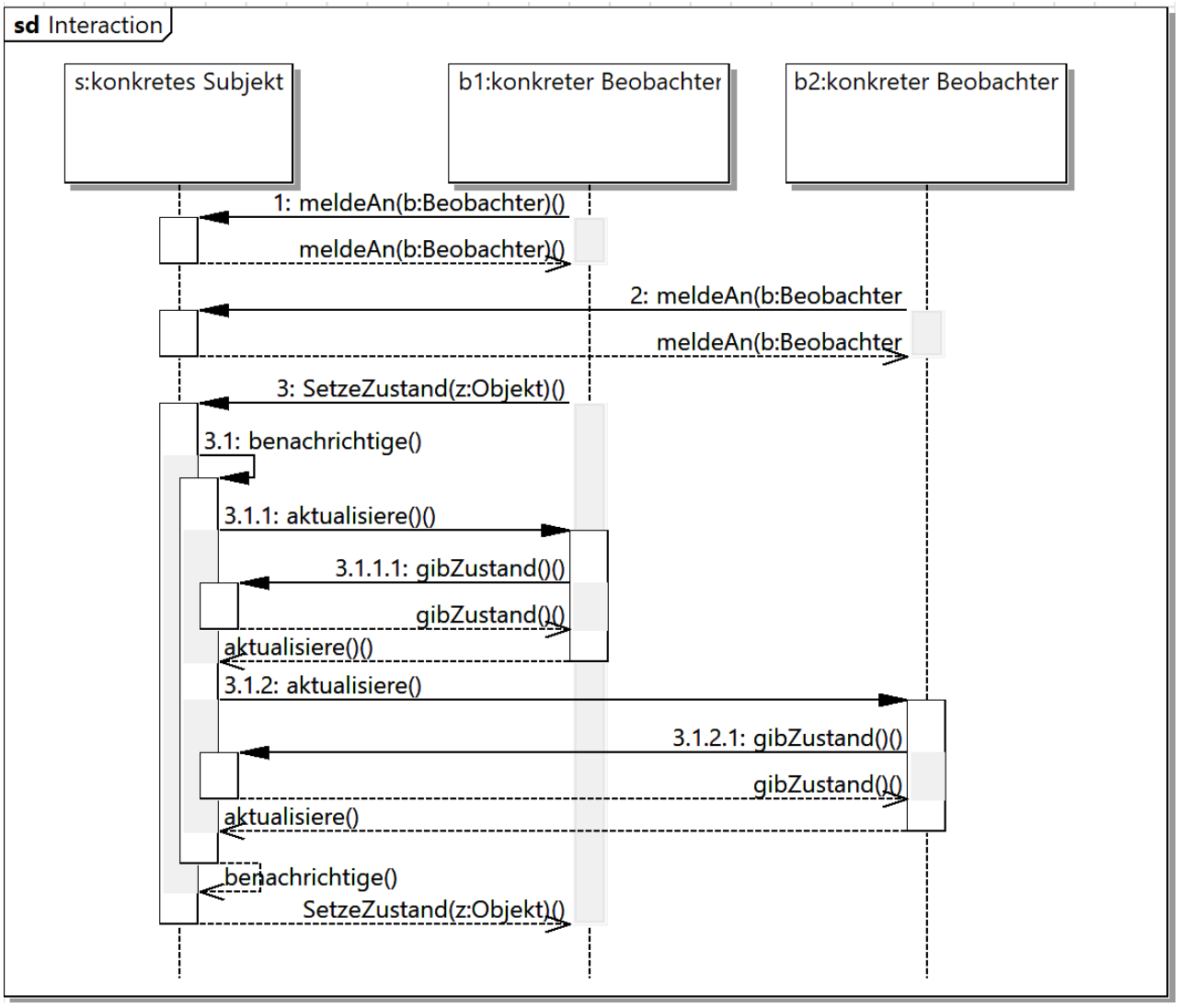


b)



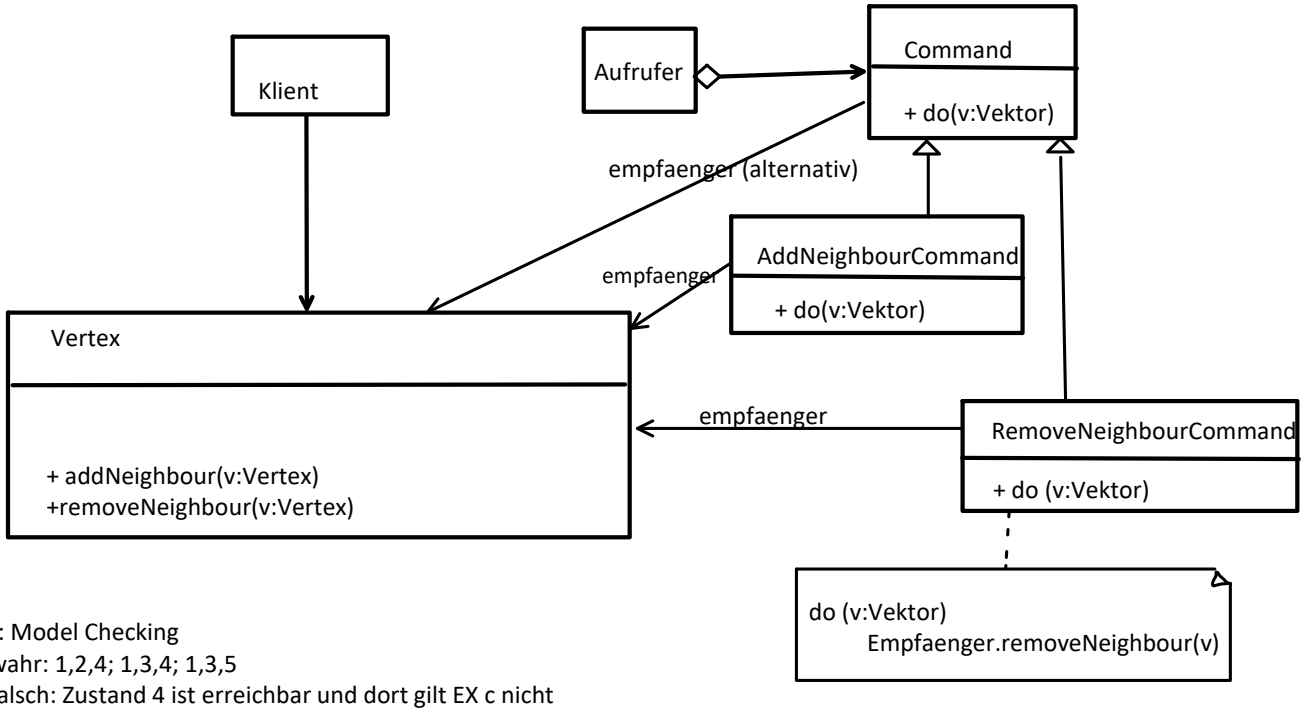
Aufgabe 4: Entwurfsmuster

a)



Methodenname auf Antwortpfeil nicht zwingend nötig

b)



Aufgabe 5: Model Checking

a) wahr: 1,2,4; 1,3,4; 1,3,5
b) falsch: Zustand 4 ist erreichbar und dort gilt EX c nicht
c) Wahr: Schleife 1,2,4,1.
d) falsch: Schleife 1,2,4,1, es wird EX a nie erreicht