



Klausur Software Engineering

Aufgaben und Punkte

Die Bearbeitungszeit der Klausur umfasst 90 Minuten. Es gibt 8 Aufgaben mit insgesamt 80 zu erreichenden Punkten.

Notieren Sie Ihre Lösungen wenn möglich direkt auf dem Aufgabenblatt. Sollte der Platz nicht ausreichen, verwenden Sie zusätzliche Blätter, die Ihnen gestellt werden. Benutzen Sie jede Seite der zusätzlichen Blätter nur für genau eine Aufgabe und notieren Sie Ihren Namen und Ihre Matrikelnummer am oberen Rand des Blattes.

Rechts oben auf jeder Seite der Klausur stehen die Punkte für eine gesamte Aufgabe. Die in Klammern gesetzten Zahlen dahinter geben die Punkte für die einzelnen Teilaufgaben an, von links nach rechts, jeweils beginnend mit Teilaufgabe a).

Persönliche Daten

Notieren Sie im Folgenden Ihre persönlichen Daten. Geben Sie Ihren Namen und Ihre Matrikelnummer außerdem auf jedem weiteren Blatt der Klausur am oberen Rand, sowie auf jedem zusätzlichen Zettel, den Sie benutzen, wie vorher erläutert an.

Vorname: _____

Nachname: _____

Geburtsdatum: _____

Matrikelnummer: _____

Studiengang: _____

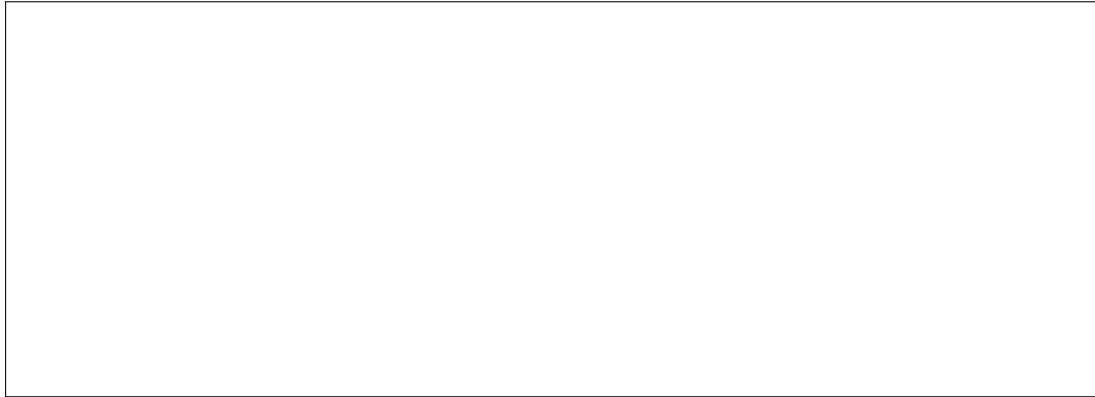
Viel Erfolg!

Matrikelnummer:

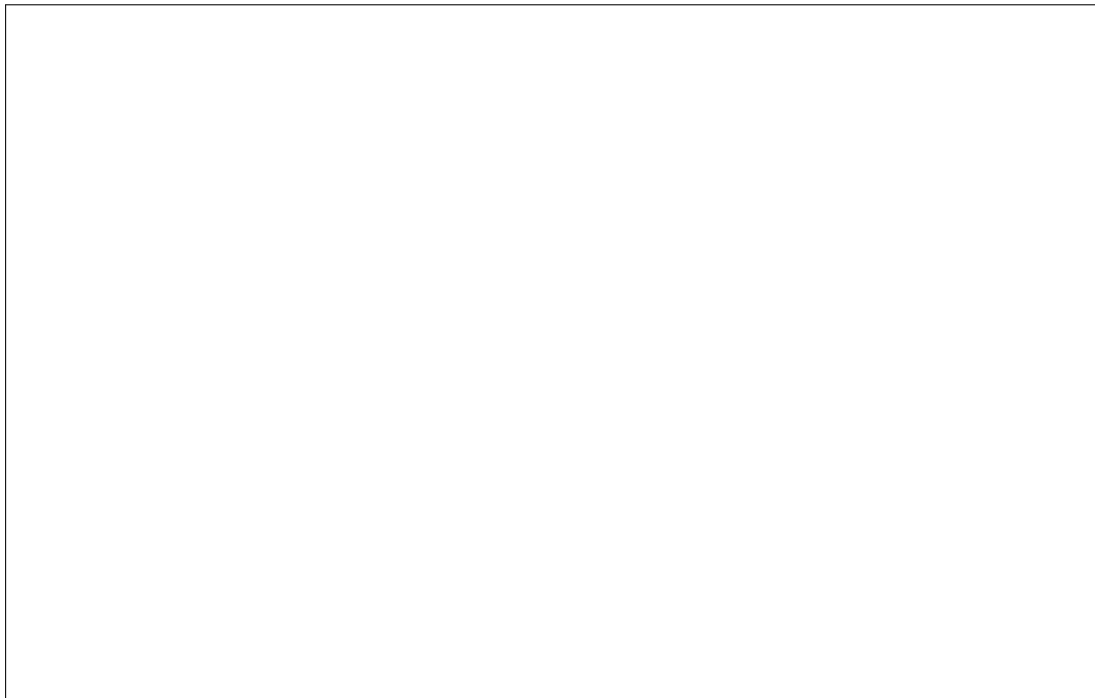
Aufgabe 1: Softwareentwicklungsprozess

9 Punkte (2/3/4)

- a) Was ist der Zweck von Vorgehensmodellen in der Softwareentwicklung?



- b) Nennen Sie ein inkrementelles, sowie ein nicht-inkrementelles Vorgehensmodell und beschreiben Sie, für welche Art von Projekt sich das genannte Modell jeweils eignet.



Matrikelnummer:

- c) Beschreiben Sie das Vorgehen als auch die Vor- und Nachteile beim Einsatz von Extreme Programming (XP).

Aufgabe 2: Anforderungsanalyse

9 Punkte (5/2/2)

Ihre Firma soll einen Geldautomaten inklusive Software für ein großes Bankhaus entwickeln. Der Kunde beschreibt Ihnen die Funktionalität des Automaten wie folgt:

„Ein Kunde kann überhaupt nur Sachen machen, wenn er sich identifiziert. Das passiert, indem er eine Chipkarte einführt. Das muss nicht nur eine EC-Karte sein, unser Bankhaus unterstützt auch Kreditkarten. Nach dem Einführen der Chipkarte, egal von welchem Typ, hat der Kunde dann ein paar Möglichkeiten: Er kann sich den Kontostand anzeigen lassen, oder Geld ausgezahlt bekommen, aber maximal 10.000 € am Tag. Bei mehr müsste er an einen Schalter gehen. Wenn der Kunde aber irgendeine Aktion macht, bei der sein Guthaben belastet wird, reicht die Karte zur Identifikation allein nicht aus, dann muss der Kunde auch noch seinen vierstelligen PIN-Code eingeben. Nach der Auszahlung von Geld hat der Kunde außerdem die Möglichkeit, sich nochmal seinen Kontostand anzusehen.

Und dann gibt's ja auch noch den Fall, dass der Automat gewartet werden muss. Das macht bei uns der Hausmeister. Er füllt frisches Geld nach, kann aber auch ein Softwareupdate aufspielen. Um all diese Sachen machen zu können, hat der Automat eine Art Hausmeister-Modus. Um da rein zu kommen, muss er einen Geheimcode eingeben, das ist bei uns einfach immer 0000. Damit der Hausmeister dann auch alle Funktionen ausprobieren kann, verfügt er bei uns außerdem über ein extra Testkonto.“

- a) Zeichnen Sie ein UML-Anwendungsfalldiagramm für die beschriebene Anwendung.



Matrikelnummer:

- b) Nennen Sie ein weiteres Beschreibungsmittel für Anforderungen und erläutern Sie die Vorzüge gegenüber Anwendungsfalldiagrammen.

- c) Sie haben den Auftrag für den Geldautomaten bekommen und möchten in der späteren Planungsphase den Entwicklungsaufwand abschätzen. Nennen Sie eine Methode, die sich dafür anbietet, und erläutern Sie grob das Vorgehen dabei.

Aufgabe 3: Petri-Netze

12 Punkte (3/1,5/4,5/1/2)

Es sei folgender Netzgraph gegeben:

$$N = (S, T, F)$$

$$S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$$

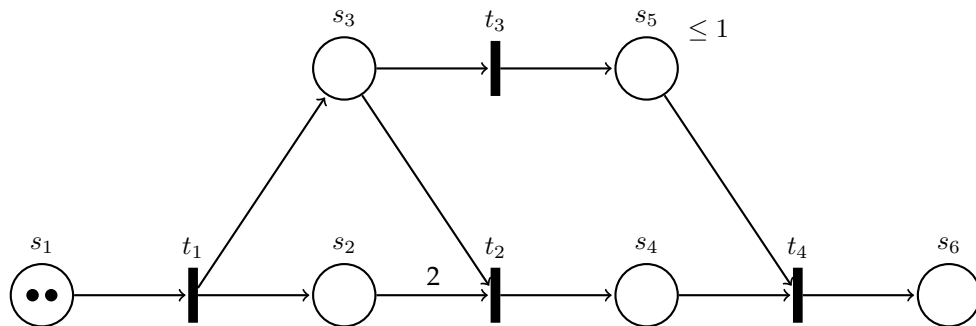
$$T = \{t_1, t_2, t_3, t_4\}$$

$$F = \{(t_1, s_1), (s_2, t_1), (s_5, t_1), (s_3, t_3), (s_4, t_3), (t_3, s_2), (s_2, t_2), (t_2, s_3), (t_2, s_4), (s_5, t_4), (t_4, s_6), (t_4, s_5)\}$$

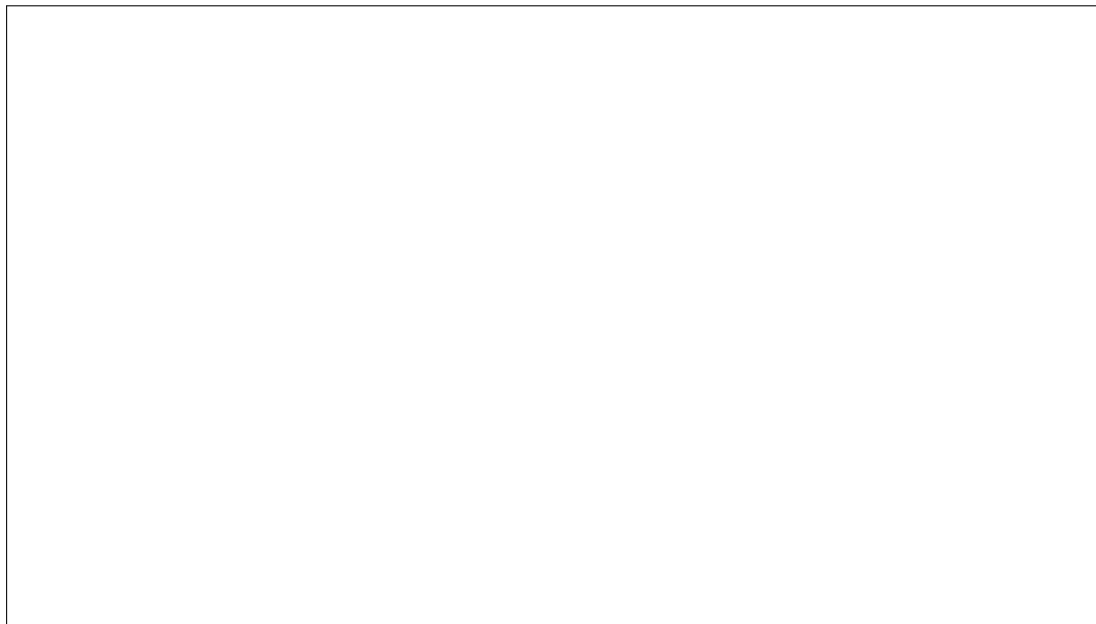
- Geben Sie den Netzgraph in der aus der Vorlesung bekannten Graph-Darstellung an.
- Zeichnen Sie in den Netzgraph eine Anfangsmarkierung ein, von der aus keine Verklemmung erreichbar ist. Gehen Sie dabei davon aus, dass jede Stelle die Kapazität ∞ und jede Kante das Gewicht 1 besitzt.

Matrikelnummer:

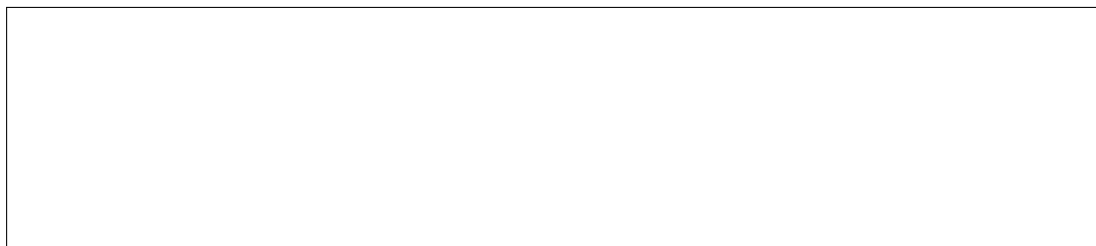
Für Teilaufgaben c) bis e) ist folgendes Petri-Netz gegeben. Beachten Sie: Stelle s_5 besitzt eine Maximalkapazität von 1.



c) Zeichnen Sie den Erreichbarkeitsgraphen des Petri-Netzes.



d) Ist es möglich von der Anfangsmarkierung aus eine Verklemmung zu erreichen? Begründen Sie die Antwort.



e) Das Petri-Netz ist nicht lebendig. Ergänzen Sie in der obigen graphischen Darstellung **eine** Transition, um das Petri-Netz lebendig zu machen.

Aufgabe 4: Lineare Temporallogik

12 Punkte (je 2)

Sei im Folgenden $AP = \{p, q, r\}$ und $\Sigma = 2^{AP}$.

Geben Sie für jede der folgenden LTL-Formeln an:

- Einen unendlichen Lauf über Σ , der die Formel erfüllt, oder begründen Sie kurz, warum es keinen solchen Lauf gibt
- Die Anzahl, wie viele unendliche Läufe über Σ die Formel erfüllen (mit kurzer Begründung)

a) $(p \wedge \mathcal{X} q) \vee \text{false}$

b) $p \wedge \mathcal{X} q \wedge \mathcal{X} \mathcal{X} \mathcal{G}(\neg p \wedge \neg q \wedge \neg r)$

c) $\mathcal{GF} p$

Matrikelnummer:

d) $\mathcal{G} r \wedge \mathcal{F}((\neg r) \mathcal{R} p \wedge q \mathcal{U}(\neg p))$

e) $(q \vee r) \wedge \mathcal{G}(p \rightarrow \mathcal{X} q \wedge q \rightarrow \mathcal{X} p)$

f) $\mathcal{G}(r \rightarrow (\neg q \wedge \neg p)) \wedge \neg \mathcal{F}(q \vee \neg(q \mathcal{U}(\neg r \rightarrow q)))$

Aufgabe 5: Algebraische Spezifikation**12 Punkte**

Im Folgenden soll ein abstrakter Datentyp `BinaryTree` für unsortierte Binärbäume spezifiziert werden. Diese sollen Elemente der Sorte `nat` speichern können. Ein neuer Baum kann dabei aus dem leeren Baum `empty` durch Anwenden der Operation `mkTree` erzeugt werden. `mkTree` erhält dabei ein Element der Sorte `nat` (für den Wert der Wurzel) und zwei Elemente der Sorte `binaryTree` (für den linken und rechten Teilbaum) und generiert ein neues Element der Sorte `binaryTree`. Außerdem gibt es noch vier weitere Operationen:

- Die Operation `size` gibt die Größe eines `binaryTree` zurück, also die Anzahl aller seiner Knoten.
- Die Operation `contains` prüft, ob ein Binärbaum ein bestimmtes Element enthält.
- Die Operation `height` berechnet die Höhe des Baums, also die Länge des längsten Pfades von der Wurzel zu einem Blatt.
- Die Operation `exchange` tauscht jedes Vorkommen eines bestimmten Elements im Baum durch ein anderes Element aus.

Sie dürfen in dieser Aufgabe auf folgende Spezifikationen zugreifen:

```

spec Bool =
sort
  bool = true | false
ops
  not: (bool) bool
  and: (bool, bool) bool
  or: (bool, bool) bool
vars
  x: bool
axioms
  false ≠ true,
  not(true) = false,
  not(false) = true,
  and(true, x) = x,
  and(false, x) = false,
  or(true, x) = true,
  or(false, x) = x
end

spec Nat = Bool then
sort
  nat = zero | succ(nat)
ops
  add: (nat, nat) nat
  max: (nat, nat) nat
vars
  m, n: nat
axioms
  add(zero, n) = n,
  add(succ(m), n) = succ(add(m, n)),
  max(zero, n) = n,
  max(n, zero) = n,
  max(succ(m), succ(n)) = succ(max(m, n))
end

```

Matrikelnummer:

Geben Sie eine algebraische Spezifikation für `BinaryTree` an.

Aufgabe 6: Sequenzdiagramm**12 Punkte**

Ein Praktikant hat für Ihr Projekt eine sortierte Listen-Datenstruktur entwickelt, welche auf dem Entwurfsmuster Composite basiert (siehe Quellcode). Für die anderen Entwickler im Projekt möchten Sie mittels eines Sequenzdiagramms die Funktionsweise dieser Liste verdeutlichen. Beschriften Sie alle Pfeile für Methodenaufrufe mit den Methodennamen und den konkreten Parameterwerten. Geben Sie bei den Antwortpfeilen ebenso Methodennamen und Parameterwerte und zusätzlich den Rückgabewert (falls vorhanden) an. Sie brauchen nur Objekte der Klassen darstellen, die in dieser Aufgabe angegeben sind.

```
public interface ListElement {
    public ListElement insert(int value);
    public boolean contains(int value);
}

public class Fin implements ListElement {
    public ListElement insert(int value) {
        return new DataElement(value, this);
    }

    public boolean contains(int value) {
        return false;
    }
}

public class DataElement implements ListElement {
    private int value;
    private ListElement next;

    public DataElement(int value, ListElement next) {
        this.value = value;
        this.next = next;
    }

    public ListElement insert(int value) {
        if (value < this.value) {
            return new DataElement(value, this);
        } else if (value > this.value) {
            next = next.insert(value);
        }
        return this;
    }

    public boolean contains(int value) {
        if (value == this.value) {
            return true;
        } else if (value < this.value) {
            return false;
        } else {
            return next.contains(value);
        }
    }
}
```

Zeichnen Sie ein UML-Sequenzdiagramm für den Aufruf folgender Main-Methode.

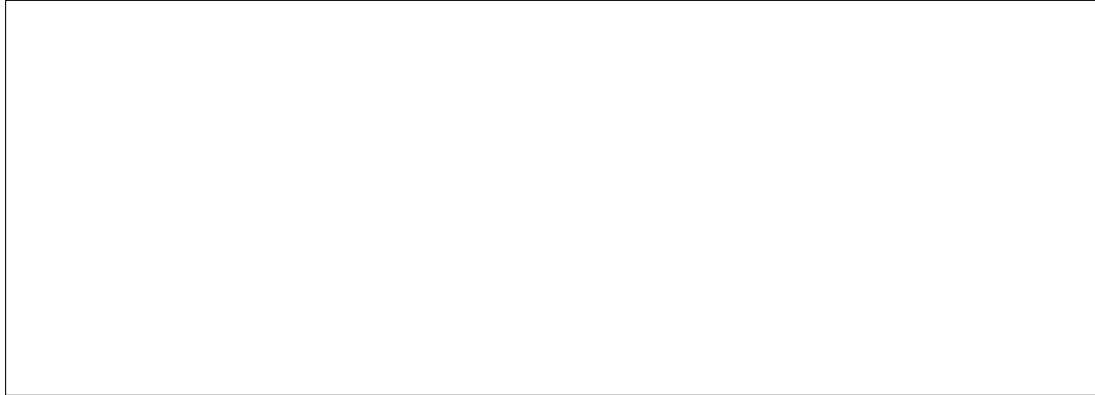
```
public class Main {  
    private static ListElement head;  
  
    public static void main(String[] args) {  
        head = new Fin();  
        head = head.insert(5);  
        head = head.insert(3);  
        System.out.println("List contains 7: " + head.contains(7));  
    }  
}
```



Aufgabe 7: Programmablaufplan

6 Punkte (1/5)

- a) Kann jedes Programm, das mittels eines Programmablaufplans dargestellt werden kann, auch mithilfe eines Struktogramms dargestellt werden? Begründen Sie Ihre Antwort kurz.



Für die nächste Teilaufgabe ist folgendes Java-Programm gegeben:

```
int ggt(int a, int b) {  
  
    int min = 0;  
    if (a < b) {  
        min = a;  
    } else {  
        min = b;  
    }  
  
    int teiler = 1;  
  
    for (int i = 1; i <= min; i++) {  
  
        if (a % i == 0) {  
            if (b % i == 0) {  
                teiler = i;  
            }  
        }  
    }  
  
    return teiler;  
}
```

Matrikelnummer:

- b) Stellen Sie das Programm in einem Programmablaufplan dar.

A large, empty rectangular box with a thin black border, intended for drawing a flowchart (Programmablaufplan) based on the program described in the previous question.

Aufgabe 8: Testen & Runtime Verification**8 Punkte (6/2)**

- a) Vervollständigen Sie die folgenden JUnit-Testfälle an den gekennzeichneten Stellen geeignet. Sie dürfen sowohl JUnit 4 als auch JUnit 5 Syntax nutzen.

`IntList` soll eine Liste von Integern verwalten. Sie verfügt dabei über folgende Funktionalitäten:

- `new IntList()` generiert eine neue, leere Liste.
- `add(i)` fügt das Element `i` hinten an die Liste an.
- `remove(i)` entfernt alle Vorkommen von `i` aus der Liste.
- `addList(l)` fügt eine zweite Liste `l` hinten an die Liste an.
- `size()` gibt die Anzahl der Elemente in der Liste zurück.
- `contains(i)` gibt `true` zurück, falls die Liste mindestens einmal `i` enthält, sonst `false`.

Im Folgenden sollen Unit-Tests für die Methoden `size` und `contains` erstellt werden:

```
public class IntListTest {
    IntList list1;
    IntList list2;

    _____

    public void setUp() {
        list1 = new IntList();
        list2 = new IntList();
    }

    _____

    public void testSize() {
        list1.add(3);
        list2.add(7);
        list2.add(7);

        _____(list1.size(), 1);

        _____(list2.size(), 2);

        _____

        _____(list1.size() == 3);

        list1.remove(7);
        list1.remove(4);

        assertTrue(_____);
    }

    _____

    public void testContains() {
        list1.add(5);
        list2.add(3);
        list1.addList(list2);

        assertTrue(_____);

        assertTrue(_____);

        _____
    }
}
```


Matrikelnummer:

- b) Skizzieren Sie kurz die Funktionsweise von Runtime Verification. Legen Sie insbesondere Gemeinsamkeiten und Unterschiede zu herkömmlichem Testen dar.



Matrikelnummer:

Korrektur

Diese Seite wird von den Korrektoren ausgefüllt.

Aufgabe	Erreichte Punkte	Mögliche Punkte
1. Softwareentwicklungsprozess		9
2. Anforderungsanalyse		9
3. Petri-Netze		12
4. Lineare Temporallogik		12
5. Algebraische Spezifikation		12
6. Sequenzdiagramm		12
7. Programmablaufplan		6
8. Testen & Runtime Verification		8
Gesamtpunktzahl		80

Note: _____