

MUSTERLÖSUNG SS 2018

Technologien und Methoden der Softwaresysteme I

Softwaretechnik I

- Alle Angaben ohne Gewähr -

Name:	Vorname:	Matrikel-Nr:

Prüfungstag: 24.09.2018

Prüfungsdauer: 120 Minuten

Maximalpunktzahl: 120 Punkte

Prüfungsumfang: 4 Aufgaben (22 Seiten)

Hilfsmittel: Alle, außer Kommunikationsgeräte

- Hinweise:**
- 1.) Verlangt sind alle Aufgaben.
 - 2.) Verwenden Sie **keinen Rotstift / Grünstift / Bleistift**.
 - 3.) Tragen Sie auf dem **Deckblatt** Ihren **vollen Namen** und Ihre **Matrikelnummer** ein.
 - 4.) Tragen Sie auf **allen Blättern** Ihre **Initialen** (1. Buchstabe von Vor- und Nachnamen) ein.
 - 5.) Es werden **nur** die Blätter **gewertet**, die mit Ihren **Initialen** und versehen sind.
 - 6.) Verwenden Sie für Ihre Lösung unbedingt die **jeweiligen Lösungsfelder**.
 - 7.) Geben Sie dieses **Deckblatt zusammen** mit Ihren in der **richtigen Reihenfolge sortierten Aufgabenblättern** ab.

----- Nicht von Studierenden auszufüllen -----

Aufgabe	1	2	3	4				
----------------	----------	----------	----------	----------	--	--	--	--

Maximal	38	22	30	30				
Erreicht								

Aufgabe 1: Analyse und Entwurf (38 Punkte)

Sie arbeiten als Softwareentwickler für das IAS, dieses will ein multifunktionales Küchengerät „COOKIAS“ entwickeln. Als einer der Entwickler sollen Sie zunächst die Analyse der neuen Steuerungssoftware durchführen. Da am IAS innovativ entwickelt wird, soll dies objektorientiert erfolgen. Abbildung 1 zeigt den schematischen Aufbau der Maschinenhardware zu der Sie die Softwaresteuerung entwickeln sollen. Abbildung 2 zeigt das Bedienpanel des Küchengeräts.

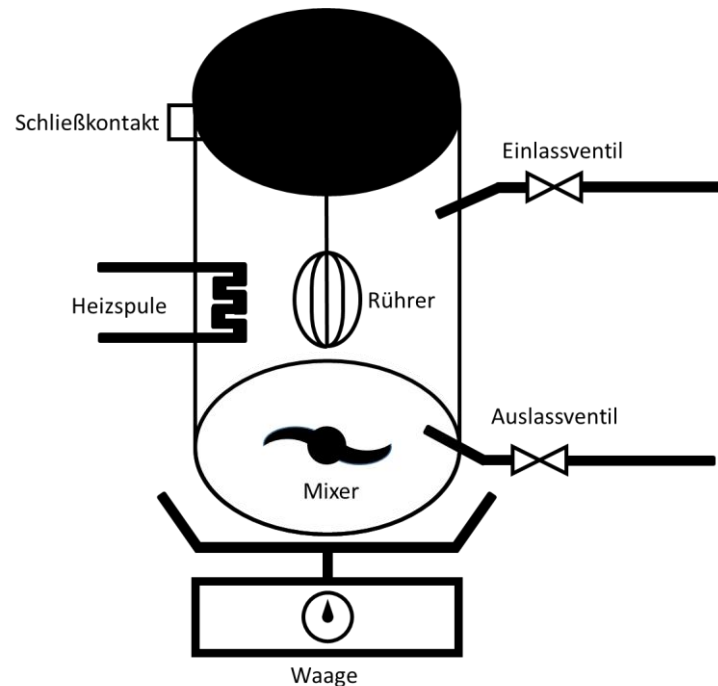


Abbildung 1: Schematischer Aufbau des Küchengeräts

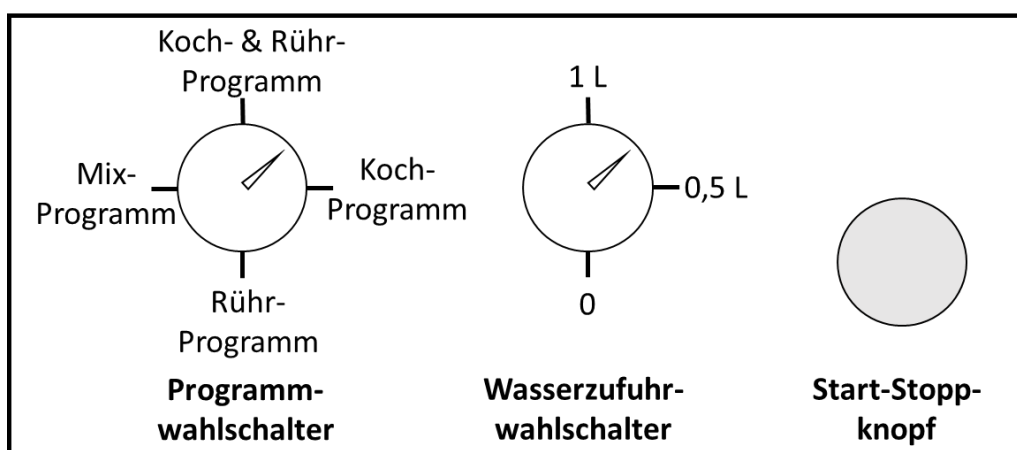


Abbildung 2: Bedienpanel des Küchengeräts

Das Küchengerät besitzt folgende Grundfunktionalitäten: *Rühren, Mixen, Heizen, Wiegen, Wasser einlassen und Wasser auslassen.*

Folgende Programme sollen wählbar sein:

Koch- & Rühr-Programm:

Nach dem Betätigen des Start-Stopp-Knopfs wird, je nach Schalterstellung des Wasserzufuhrwahlschalters zunächst das Wasser in den Behälter eingelassen. Sollte die Zufuhr von Wasser ausgewählt sein, so wird hierzu mit Hilfe der Waage festgestellt, wann die benötigte Menge an Wasser im Behälter ist und die Wasserzufuhr wird gestoppt. Anschließend erhitzt die Heizspule den Behälterinhalt, bringt diesen zum Kochen und hält die benötigte Temperatur. Außerdem beginnt parallel dazu der Rührer-Motor damit, den Rührer gleichmäßig zu drehen. Ein erneutes Betätigen des Start-Stopp-Knopfs beendet den Heiz- sowie den Rührvorgang. Wird der Start-Stopp-Knopf dabei 3 Sekunden lang gedrückt gehalten und wurde zuvor Wasser zugeführt, so wird anschließend die zugeführte Menge Wasser wieder abgelassen. Um festzustellen wann die zugeführte Menge an Wasser wieder abgelassen wurde, wird wiederum die Waage verwendet.

Koch-Programm:

Der Unterschied zum *Koch- & Rühr-Programm* besteht darin, dass kein Rührvorgang stattfindet.

Rühr-Programm:

Der Unterschied zum *Koch- & Rühr-Programm* besteht darin, dass kein Heizvorgang stattfindet.

Mix-Programm:

Der Unterschied zum *Koch- & Rühr-Programm* besteht darin, dass weder Heiz- noch Rührvorgang stattfinden. Stattdessen wird, nach dem optionalen Zuführen von Wasser, der Mixer-Motor gestartet, dieser dreht den Mixer solange gleichmäßig bis der Start-Stopp-Knopf erneut (wahlweise kurz oder 3 Sekunden lang) betätigt wird.

Zusätzlich sind folgende Randbedingungen zu berücksichtigen:

- Es muss sichergestellt sein, dass der Deckel bei allen Vorgängen geschlossen ist.
- Es darf nie mehr als ein Ventil gleichzeitig geöffnet sein.

Hinweise:

- Vereinfachend sollen Effekte wie das Verdampfen von Wasser nicht berücksichtigt werden.
- Das Zuführen von Zutaten (mit Ausnahme des Wassers) geschieht vor der Programmauswahl manuell und soll daher nicht weiter betrachtet werden.
- Die Motoren besitzen für den Rühr- bzw. Mix-Vorgang ein definiertes Programm, das direkt von der Steuerung angewählt werden kann.
- Alle Ventile besitzen genau zwei Zustände (offen und geschlossen).
- Die Heizspule regelt selbst die vorgegebene Solltemperatur ein, d.h. die Steuerung muss lediglich die Heizung ein- bzw. ausschalten.
- Der Programmwahlschalter besitzt vier Wahlmöglichkeiten: *Koch- & Rühr-Programm*, *Koch-Programm*, *Rühr-Programm* und *Mix-Programm*.
- Der Wasserzufuhrwahlschalter besitzt drei Wahlmöglichkeiten: 1 L, 0,5 L und 0 L. Er kann völlig unabhängig vom Programmwahlschalter eingestellt werden, d.h. alle Programme können unter Zufuhr von allen gegebenen Wassermengen benutzt werden.

Aufgabe 1.1: Use Cases (10 Punkte)

Die Softwaresteuerung des Küchengeräts soll ausgehend von obiger verbaler Beschreibung mit Hilfe von Use Cases modelliert werden. Gehen Sie dabei wie folgt vor:

- a) Identifizieren Sie 4 (abgeschlossene) Use Cases und die dazugehörigen Akteure. (2 Punkte)

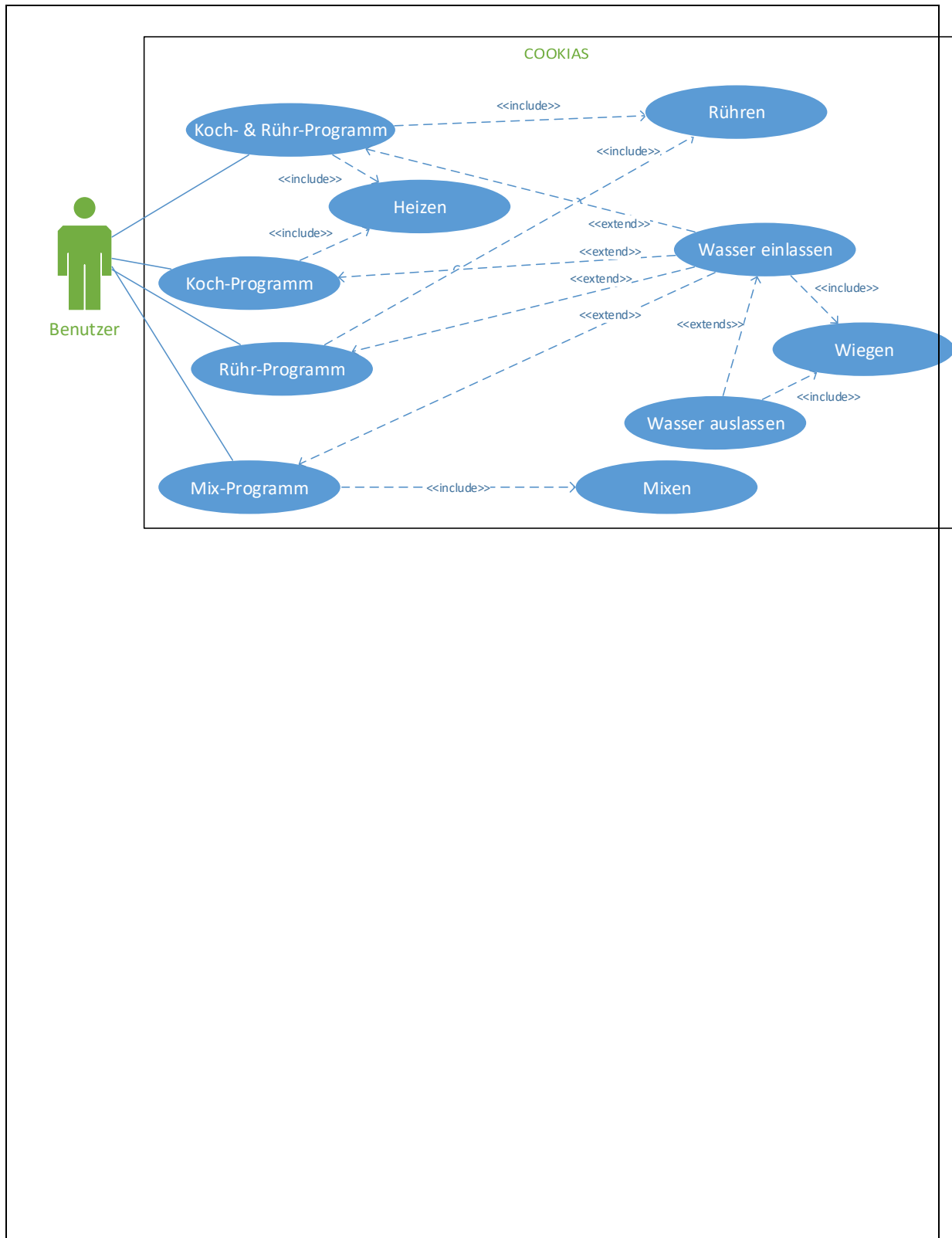
Koch- & Rühr-Programm;
Koch-Programm;
Rühr-Programm;
Mix-Programm;

Akteur für alle Use Cases: Benutzer

- b) Die Use Cases aus a) können mit Hilfe von 6 enthaltenen (partiellen) Use Cases realisiert werden. Identifizieren Sie diese 6 Use Cases. (3 Punkte)

Rühren, Mixen, Heizen, Wiegen, Wasser einlassen und Wasser auslassen

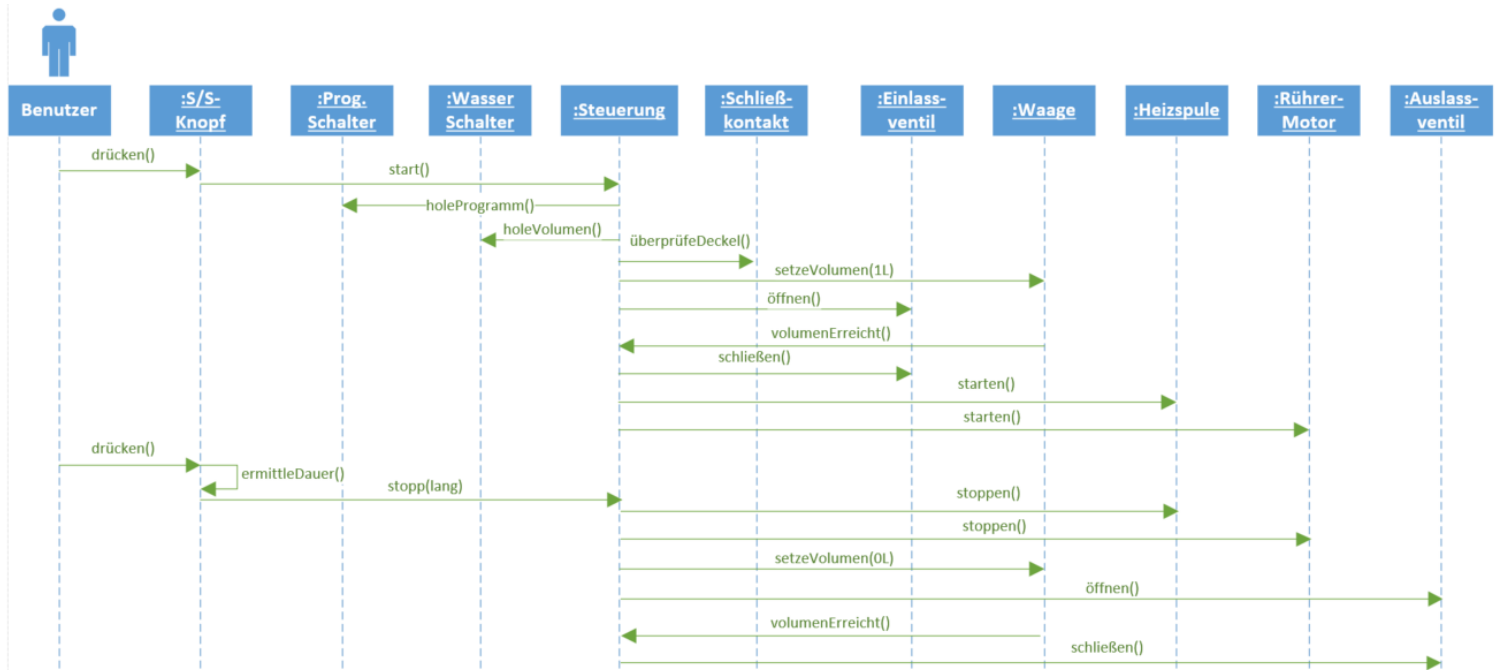
- c) Zeichnen Sie das entsprechende Use Case Diagramm. Verwenden Sie `<<include>>` bzw. `<<extend>>`-Beziehungen. Verwenden Sie die UML-Notation. (5 Punkte)



Aufgabe 1.2: Sequenzdiagramm (14 Punkte)

Identifizieren Sie relevante Objekte für die Steuerungssoftware und zeichnen Sie ein Sequenzdiagramm, nach UML-Notation, für das Programm "Koch- & Rühr-Programm mit 1 L Wasserzufuhr" (Der Programm- und der Wasserzufuhrwahlschalter sind bereits entsprechend eingestellt, die Zutaten befinden sich im Behälter). Bei der zweiten Betätigung des Start-Stopp-Knopfs wird dieser 3 Sekunden lang gedrückt gehalten.

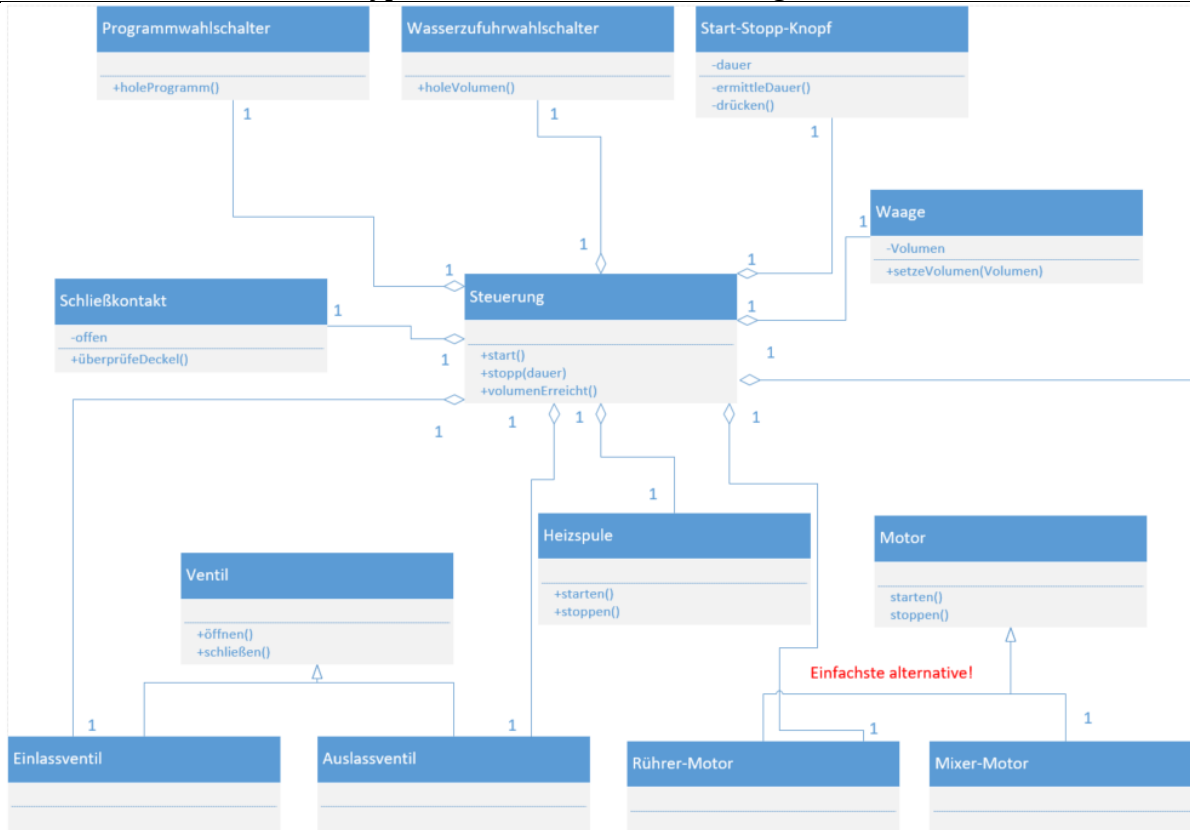
Bilden Sie alle benötigten physikalischen Entitäten aus Abbildung 1 und 2 ab.



Aufgabe 1.3: Klassendiagramm (14 Punkte)

Zeichnen Sie ausgehend von Aufgabe 1.2 ein Klassendiagramm der Softwaresteuerung des Küchengeräts nach UML-Notation. Identifizieren Sie alle relevanten Klassen und ihre Attribute und Methoden, sowie die Beziehungen untereinander mit den entsprechenden Multiplizitäten. Verwenden Sie Vererbungsbeziehungen. Geben Sie die Sichtbarkeiten an.

Hinweis: Die Datentypen müssen nicht berücksichtigt werden.

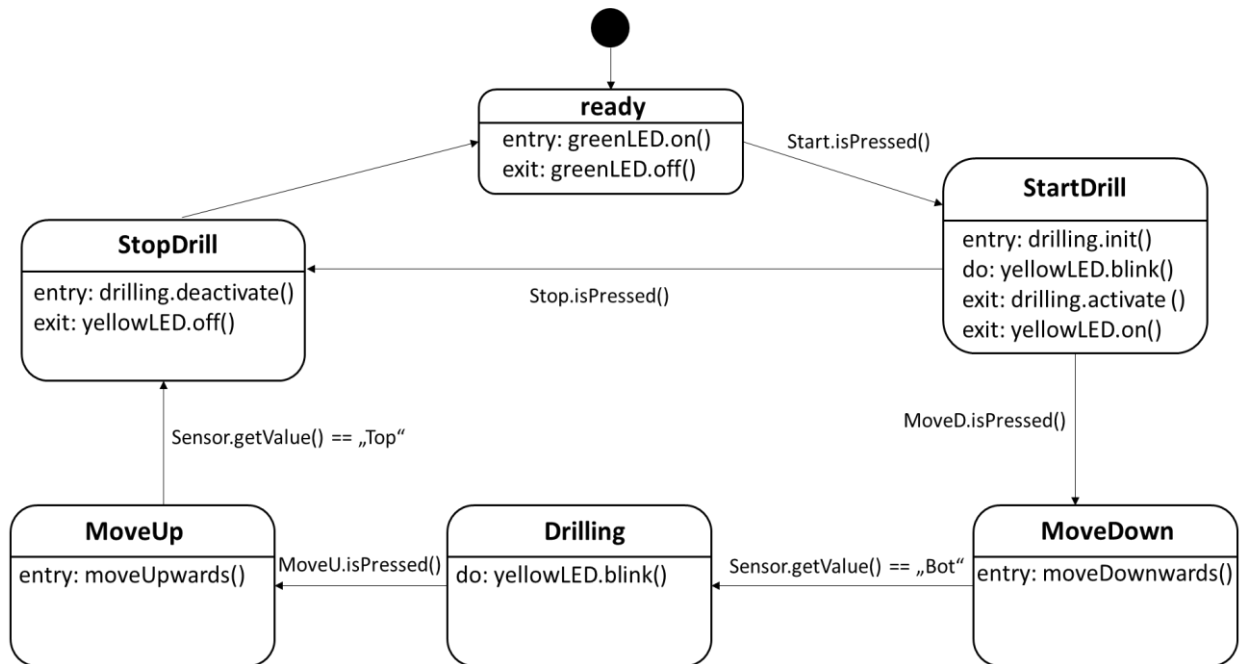


Korrekturhinweis: alle Klassennamen müssten EXAKT mit den denen im Sequenzdiagramm übereinstimmen um die Konsistenz zu wahren.

Aufgabe 2: Implementierung und Test (22 Punkte)

Aufgabe 2.1: Implementierung (5 Punkte)

Gegeben ist der folgende Zustandsautomat:



Implementieren Sie den Zustand **StartDrill** in Pseudocode, vervollständigen Sie hierzu den nachfolgend angegebenen Pseudocode. Machen Sie im Code Entry-, Do- und Exit-Funktionen durch Kommentare kenntlich.

```
SWITCH currentState
    CASE „StartDrill“:
        drilling.init() //Entry-Actions
        WHILE NOT Stop.isPressed() AND NOT MoveD.isPressed()
            yellowLED.blink() //Do-Actions
        ENDWHILE
        IF Stop.isPressed() THEN
            currentState := "StopDrill"
        ENDIF
        IF MoveD.isPressed() THEN
            currentState := "MoveDown"
        ENDIF
        drilling.activate() //Exit-Actions
        yellowLED.on()
    BREAK
...
```

Aufgabe 2.2: Test (17 Punkte)

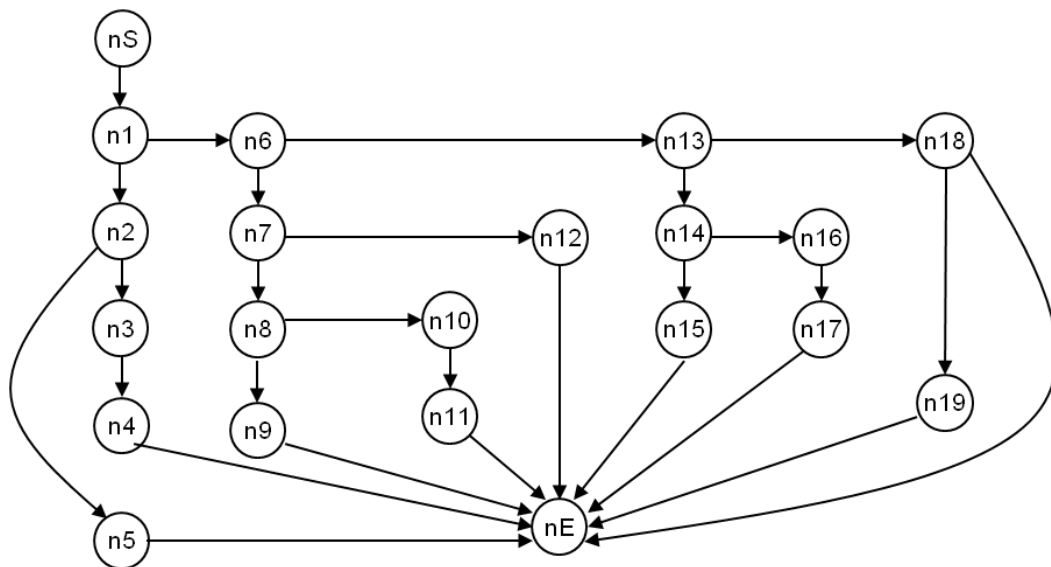
Sie sollen eine Funktion *heatcontrol()* eines Kaffeeautomaten testen, diese ist in Abbildung 3 dargestellt. Die Funktion steuert eine Heizspule in Abhängigkeit der Parameter vom aktuellen Zustand *zustand*: Werte „Bereit“, „Heizen“, „Produzieren“, „Reinigen“, dem Interrupt *interrupt*: „true“ oder „false“ und der Temperatur *temperatur*: „double im Bereich [0.0..100.0]“ und hat den Folgezustand als Rückgabewert.

Nach Ausführung eines return-Statements springt die Ausführung ans Ende der Methode (Knoten nE).

```
Zustand heatcontrol(Zustand zustand, double temperatur, bool interrupt) nS
{
    if(zustand == „Heizen“) n1
    {
        if(temperatur > 90.0) n2
        {
            heizspule.aus(); n3
            return „Bereit“; n4
        }
        else
            return „Heizen“; n5
    }
    else if(zustand == „Produzieren“) n6
    {
        if(interrupt) n7
        {
            if(temperatur > 90.0) n8
                return „Bereit“; n9
            else
            {
                heizspule.ein(); n10
                return „Heizen“; n11
            }
        }
        else
            return „Produzieren“ n12
    }
    else if(zustand == „Bereit“) n13
    {
        if(temperatur > 90.0) n14
            return „Bereit“; n15
        else
        {
            heizspule.ein(); n16
            return „Heizen“; n17
        }
    }
    else if(zustand == „Reinigen“) n18
    {
        //keine weitere Aktion, Kaffeeautomat wird ausgeschalten n19
    }
} nE
```

Abbildung 3: Funktion heatcontrol in Pseudocode

a) Zeichnen Sie den Kontrollflussgraphen für die Funktion *heatcontrol()* (5,5 Punkte)



- b) Stellen Sie Testfälle auf, mit denen Sie eine vollständige C1-Abdeckung erreichen.
Ist eine vollständige C2-Abdeckung möglich? Begründen Sie Ihre Antwort. (8,5 Punkte)

Man benötigt acht Testfälle, für jeden if/else-Zweig einen.

z. B. (Parameter ohne Wertzuweisung sind für jeweiligen Testfall irrelevant und können, innerhalb des gültigen Bereiches, beliebig gewählt werden):

- 1) zustand = "Heizen", temperatur = 100.0
- 2) zustand = "Heizen", temperatur = 80.0
- 3) zustand = "Produzieren", interrupt = false
- 4) zustand = "Produzieren", interrupt = true, ", temperatur = 100.0
- 5) zustand = "Produzieren", interrupt = true, ", temperatur = 80.0
- 6) zustand = "Bereit", temperatur = 100.0
- 7) zustand = "Bereit", temperatur = 80.0
- 8) zustand = "Reinigen"
- 9) zustand = <undefiniert>

Ja, eine vollständige C2-Abdeckung ist möglich.

Begründung: Die oben genannten Testfälle decken neben C1 auch C2 vollständig ab. (Hinweis/alternativ: dies ist insbesondere durch das Fehlen von Schleifen, welche u.U. unendlich viele Testfälle verursachen können, möglich.)

- c) Nennen Sie die Definitionen des C0-, C1- und C2-Tests und erklären Sie was diese jeweils abdecken. (3 Punkte)

C0: Anweisungsüberdeckungstest

- Alle Anweisungen müssen mindestens einmal ausgeführt werden.

C1: Zweigüberdeckungstest

- Alle Zweige wurden mindestens einmal durchlaufen.

C2: Pfadüberdeckungstest

- Alle möglichen Pfade vom Startknoten bis zum Endknoten wurden mindestens einmal durchlaufen.

$\Sigma =$

Aufgabe 3: Projektmanagement (30 Punkte)

Sie arbeiten als Projektleiter in einem Softwareunternehmen. Ihr Unternehmen will eine neue Software zur Verwaltung von Blockchain-basierten Finanzprodukten auf den Markt bringen. Sie sind mit der Projektplanung und -durchführung betraut. Bei einer ersten Projektleitersitzung wurden die in Tabelle 1 dargestellten Arbeitspakete (AP) identifiziert.

Tabelle 1: Auflistung der Arbeitspakete

AP	Beschreibung	Aufwand [PM]	Vorgänger
A	Marktanalyse im Finanzsektor	3	-
B	Analyse der Anforderungen an die Datenhaltung	6	-
C	Analyse von Sicherheitsaspekten der Blockchaintechnologie	6	-
D	Erstellen des Systemmodells	8	A
E	Erstellen des Datenmodells	4	B
F	Entwurf der Sicherheitsarchitektur	4	C
G	Entwurf der grafischen Benutzungsoberfläche	4	D, B
H	Einrichten der Dateninfrastruktur	4	E
I	Implementierung der Sicherungsalgorithmen	3	F
J	Implementierung der grafischen Benutzungsoberfläche	4	G
K	Belastungstest der Datenhaltung	2	H, F
L	Test und Zertifizierung der Sicherungsmaßnahmen	2	I

In Tabelle 2 sind alle Mitarbeiter, die Ihnen für das Projekt zur Verfügung stehen, mit Kosten pro Personenmonat (PM) sowie der Anzahl verfügbarer Mitarbeiter aufgelistet. Jeder Mitarbeitertyp ist nur für bestimmte Arbeitspakete zuständig (z. B. der Analyst bearbeitet Arbeitspakete A, B und C).

Tabelle 2: Auflistung der verfügbaren Mitarbeiter

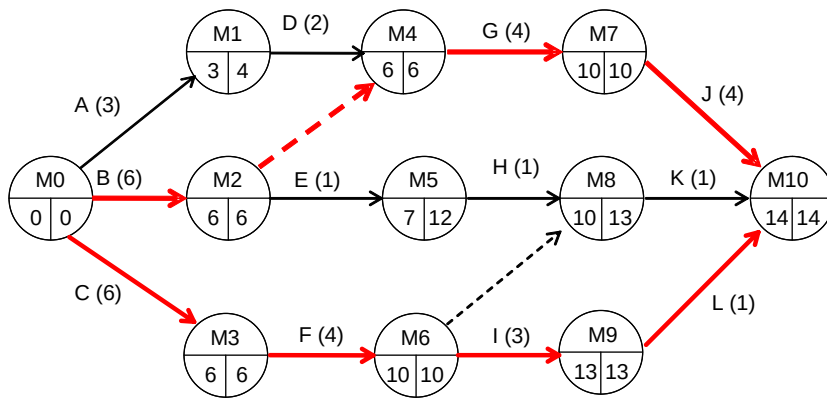
	Kosten pro PM	Anzahl verfügbarer Mitarbeiter	Arbeitspakete:
Analyst	8.000 €	3	A, B, C
SW-Entwickler	7.000 €	4	D, E, H
GUI-Entwickler	7.000 €	1	G, J
Sicherheitsexperte	10.000 €	1	F, I
Qualitätssicherer	6.000 €	2	K, L

Zur Planung des Mitarbeiterereinsatzes gelten folgende Randbedingungen:

- Alle Mitarbeiter stehen für die gesamte Projektlaufzeit zur Verfügung. Allerdings meldet einer der drei Analysten bereits vor Projektbeginn Elternzeit an. Es ist bekannt, dass er nur die ersten 3 Monate des Projektes zur Verfügung steht.
- Alle Arbeitspakete werden zum frühestmöglichen Termin begonnen.
- Können Arbeitspakete parallel bearbeitet werden, dann werden die maximal verfügbaren Mitarbeiter gleichmäßig auf die Arbeitspakete verteilt.

Aufgabe 3.1: Netzplan (11 Punkte)

Zeichnen Sie einen Netzplan in Vorgangspfeiltechnik (Critical Path Method, CPM) für die in Tabelle 1 angegebenen Arbeitspakete, Aufwände und Vorgängerbeziehungen unter Berücksichtigung der verfügbaren Mitarbeiter und geben Sie den kritischen Pfad bzw. die kritischen Pfade an. Verwenden Sie Scheinvorgänge nur dann, wenn diese zwingend erforderlich sind.



Kritische Pfade:

- 1) B, G, J
- 2) C, F, I, L



Aufgabe 3.2: Kostenberechnung (5 Punkte)

Berechnen Sie die Kosten für die Durchführung des Projekts. Gehen Sie dabei davon aus, dass die Kosten pro Personenmonat nach 12 Monaten aufgrund einer jährlichen Gehaltsanpassung um 5% steigen.

Arbeitspaket, das komplett im 2. Jahr bearbeitet wird: AP L (2 PM)

Arbeitspakete, die teilweise im ersten und teilweise im 2. Jahr bearbeitet werden: AP I (2 PM im ersten Jahr, 1 PM im 2. Jahr) und AP J (2 PM im ersten Jahr, 2 PM im 2. Jahr)

Berechnung der Kosten pro Mitarbeitertyp:

$$K_{\text{Analyst}} = 8.000\text{€/PM} * (3 \text{ PM} + 6 \text{ PM} + 6 \text{ PM}) = 120.000 \text{ €}$$

$$K_{\text{SW-Entw}} = 7.000\text{€/PM} * (8 \text{ PM} + 4 \text{ PM} + 4 \text{ PM}) = 112.000 \text{ €}$$

$$K_{\text{GUI-Entw}} = 7.000\text{€/PM} * (4 \text{ PM} + 2 \text{ PM} + 2 \text{ PM} * 1,05) = 56.700 \text{ €}$$

$$K_{\text{Sicherheitsexperte}} = 10.000\text{€/PM} * (4 \text{ PM} + 2 \text{ PM} + 1 \text{ PM} * 1,05) = 70.500 \text{ €}$$

$$K_{\text{QS}} = 6.000\text{€/PM} * (2 \text{ PM} + 2 \text{ PM} * 1,05) = 24.600 \text{ €}$$

$$\text{Gesamtkosten } K_{\text{ges}} = K_{\text{Analyst}} + K_{\text{SW-Entw}} + K_{\text{GUI-Entw}} + K_{\text{Sicherheitsexperte}} + K_{\text{QS}} = 383.800 \text{ €}$$

Analog können auch die Kosten pro Arbeitspaket berechnet und addiert werden.

Aufgabe 3.3: Personalplanung (5 Punkte)

Der Sicherheitsexperte wird dringend für ein anderes Projekt benötigt und steht erst 9 Monate nach Projektbeginn zur Verfügung.

Hinweise:

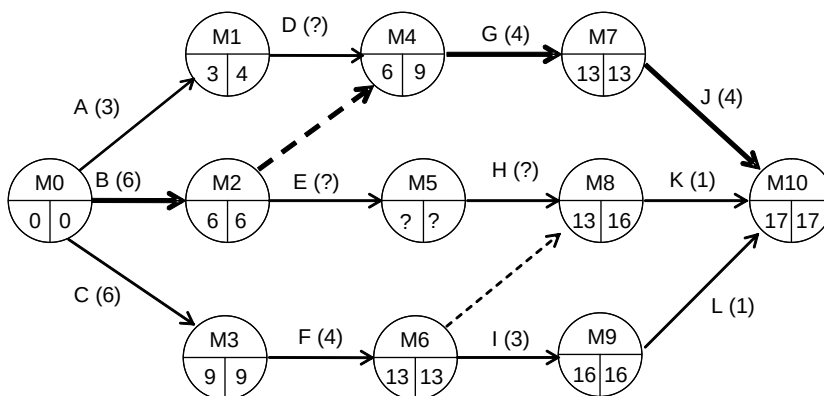
- Geben Sie sinnvoll gerundete Ergebnisse an und begründen Sie ihre Antwort.

a) Um wie viele Monate verschiebt sich der Endtermin des Projektes? (1 Punkt)

Endtermin verschiebt sich um 3 Monate, da Arbeitspaket auf kritischem Pfad liegt und AP F somit erst nach 9 Monaten gestartet werden kann (anstatt nach 6 Monaten).

b) Wie viele SW-Entwickler können Sie durch diese Verzögerung für andere Projekte freistellen, ohne dass sich der Projektendtermin weiter verschiebt? (4 Punkte)

Skizze nicht gefordert, vereinfacht aber das Nachvollziehen der Lösung:



APs E und H muss nach spätestens 16 Monaten abgeschlossen sein, Start frühestens nach 6 Monaten. Der Aufwand von 8 PM könnte prinzipiell von einem SW-Entwickler innerhalb von 10 Monaten bewältigt werden (keine mögliche Parallelität zu AP D berücksichtigt). AP D muss nach spätestens 9 Monaten abgeschlossen sein, Start nach 3 Monaten. Um 8 PM innerhalb von 6 Monaten zu leisten, sind 2 Entwickler notwendig.

Ergebnis: 2 der 4 Entwickler für andere Projekte freigestellt werden.

(Formel: Einsparbare MA = ursprüngliche MA-Anzahl – (Aufwand AP / maximale Dauer AP) – danach abrunden)

Aufgabe 3.4: Personalplanung (2) (5 Punkte)

Fünf Monate nach Projektbeginn teilt Ihnen die Geschäftsleitung mit, dass Sie einen zusätzlichen GUI-Entwickler sowie einen zusätzlichen Sicherheitsexperten einstellen dürfen. Gleichzeitig fordert die Geschäftsleitung, dass das Projekt nach spätestens 12 Monaten abgeschlossen sein muss. Ist diese Frist einzuhalten? Begründen Sie Ihre Antwort.

Gehen Sie zur Bearbeitung dieser Aufgabe davon aus, dass die Suche nach einem zusätzlichen GUI-Entwickler einen Monat und die Suche nach einem zusätzlichen Sicherheitsexperten zwei Monate benötigt. Der erste Sicherheitsexperte steht in diesem Szenario wieder über die gesamte Projektlaufzeit zur Verfügung.

Nach 6 Monaten stehen 2 GUI-Entwickler zur Verfügung, diese bearbeiten Arbeitspakete G und J.

Nach 7 Monaten steht der 2. Security-Experte zur Verfügung, diese bearbeiten Arbeitspakete F und I.

Durch die GUI-Entwickler wird AP G nach 8 Monaten beendet, AP J nach 10 Monaten.

Durch den Sicherheitsexperten wird AP F nach 8,5 Monaten beendet, I nach 10 Monaten.

Dadurch gibt sich eine Gesamtlaufzeit von 11 Monaten für das Projekt. Die Frist von 12 Monaten wird eingehalten.

Aufgabe 3.5: Fehlerentwicklung (4 Punkte)

Der Belastungstest der Datenhaltung (AP K) ergibt, dass die Anzahl der durchführbaren Blockchain-Transaktionen pro Sekunde zu gering ist, da die Anforderungen an die Datenhaltung nicht vollständig waren.

- a) Welche Arbeitspakete müssen aufgrund der vergessenen Anforderung an die Datenhaltung erneut bearbeitet werden? (1 Punkt)

Fehler in Arbeitspaket B, daher müssen AP B, AP E, AP H und AP K wiederholt werden.

Optional: falls Änderungen der Datenhaltung auch zu Änderungen an der GUI führen, müssen AP G und AP J ebenfalls erneut bearbeitet werden.

- b) Erklären Sie anhand dieses Beispiels, warum die Kosten für Fehler, die in späten Projektphasen erkannt werden, höher sind, als Fehler die bereits früh erkannt werden. (1,5 Punkte)

Vorherige Arbeitspakete müssen wiederholt werden, dabei müssen vorige Ergebnisse angepasst oder verworfen werden. Im Lauf des Projekts entstehen zahlreiche Abhängigkeiten, die spät erkannte Fehler teurer machen.

- c) Nennen Sie 3 Möglichkeiten, um Fehler bereits frühzeitig zu erkennen. (1,5 Punkte)

Beispiele:

- Reviews in frühen Projektphasen
- Frühe Modul-/Komponententest
- Scheduling-Performancetests
- Proof of concepts/Protoypen für kritische Anforderungen
- ...

Aufgabe 4: Kurzfragen (30 Punkte)

Wird nicht veröffentlicht