



MINISTERIUM FÜR KULTUS, JUGEND UND SPORT
BADEN-WÜRTTEMBERG

ABITURPRÜFUNG AM BERUFLICHEN GYMNASIUM IM SCHULJAHR 2012/2013

Hauptprüfung	AUFGABEN FÜR DAS FACH
1.5.2	Informationstechnik (TG)

Arbeitszeit	270 Minuten
Hilfsmittel	eingeführte Formelsammlung und lokale Formelsammlung zugelassener Taschenrechner
Stoffgebiet	<p>Teil 1: Hardware</p> <p>Aufgabe 1: Informationsverarbeitende Systeme (2 Seiten) Aufgabe 2: Informationsverarbeitende Systeme (3 Seiten)</p> <p>Teil 2: Software</p> <p>Aufgabe 3: Objektorientierte Analyse und Design (6 Seiten) Aufgabe 4: Objektorientierte Analyse und Design (6 Seiten)</p> <p>Teil 3: Systeme</p> <p>Aufgabe 5: Datenbank- und Betriebssysteme / vernetzte Systeme (3 Seiten) Aufgabe 6: Datenbank- und Betriebssysteme / vernetzte Systeme (3 Seiten)</p>
Bemerkungen	<p>Aus den Teilen 1 und 2 (Pflichtbereich) wählt die Fachlehrerin/der Fachlehrer je eine Aufgabe aus. Beide sind zu bearbeiten.</p> <p>Aus Teil 3 (Wahlbereich) wählt die Schülerin/der Schüler eine Aufgabe aus.</p> <p>Sie sind verpflichtet, jeden Aufgabensatz umgehend auf seine Vollständigkeit zu überprüfen und fehlende Seiten der Aufsicht führenden Lehrkraft anzuzeigen. Jede Aufgabe ist mit einem neuen Blatt zu beginnen. Bei Verstößen gegen die angemessene Darstellungsform kann ein Punkteabzug erfolgen.</p>

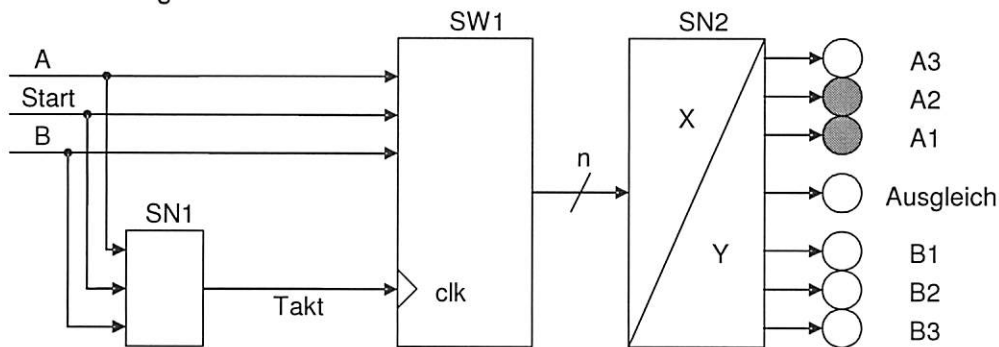
Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Hardware)	
	Teil: 1 (Pflichtbereich)	Aufgabe 1: (2 Seiten)

Punkte

Drei gewinnt

Zu Trainingszwecken wird bei Ballspielen oft eine veränderte Zählweise eingesetzt. In diesem Fall soll ein Spiel so lange dauern, bis ein Team (A oder B) mit drei Punkten führt. Dazu soll folgendes Anzeigesystem entwickelt werden.

Mit dem Signal Start leuchtet die mittlere LED Ausgleich (Spielstand A:B = 0:0). Erzielt danach Team A einen Punkt so leuchtet A1 (Spielstand 1:0). Erzielt Team A einen weiteren Punkt, so leuchtet A1 und A2 (Balkenanzeige, Spielstand 2:0, siehe Abb.). Erzielt danach Team B einen Punkt, so bekommt Team A wieder einen Punkt abgezogen, womit dann wieder nur noch A1 leuchtet (Spielstand 1:0). Würde Team B daraufhin in Folge 4 Punkte erzielen, so wäre der Spielstand 0:3 (B1, B2 und B3 leuchten). Team B hätte somit gewonnen. Eine weitere Betätigung von A oder B hat dann keine Auswirkung mehr. Mit dem Startsignal kann die Anzeige jederzeit wieder auf den Zustand 0:0 gebracht werden.



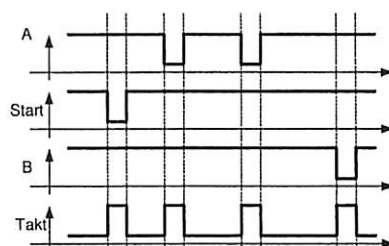
1.1 Schaltnetz 1 (SN1)

Die Eingabe von A, B oder Start erfolgt durch prellfreie Taster, die bei Betätigung einen **Low-Pegel** (0) liefern! Das Schaltnetz SN1 hat die Aufgabe, bei jeder Betätigung eines Tasters ein Taktsignal (positive Flanke) für das Schaltwerk SW1 zu liefern. Es kann davon ausgegangen werden, dass immer nur einer der drei Taster (A, B, Start) aktiviert ist.

1.1.1 Stellen Sie den Zusammenhang von SN1 in einer Wahrheitstabelle (Funktionstabelle) dar.

1.1.2 Geben Sie die Funktionsgleichung für das Taktsignal an.

Beispielimpulsfolge:



2

1

1.2 Schaltwerk 1 (SW1)

Im Einschaltmoment (Power On bzw. Reset) sollen zur Funktionskontrolle alle 7 LEDs an sein. Mit der Betätigung von Start leuchtet dann nur die LED Ausgleich. Das Spiel kann beginnen (vgl. Text oben - lowaktive Taster).

1.2.1 Zeichnen Sie ein Zustandsdiagramm mit Angabe der 7 Ausgangssignale B3 bis A3.

5

1.2.2 Ermitteln Sie die minimal erforderliche Anzahl der Bit-Speicher (D-FFs).

1

1.2.3 Kodieren Sie die Zustände (minimale Anzahl an D-FF) und stellen Sie in einer kodierten Zustandsfolgetabelle den nachfolgenden Ablauf dar:

Start → A macht einen Punkt → B macht einen Punkt → ...

2

1.2.4 Bestimmen Sie die Anzahl der Zeilen zur vollständigen Zustandsübergangstabelle und begründen Sie ihre Lösung.

2

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Hardware)	
	Teil: 1 (Pflichtbereich)	Aufgabe 1: (2 Seiten)

Punkte

Mikrocontroller

Das System ist nun mit einem Mikrocontroller zu realisieren. Zur Funktionskontrolle der LEDs sollen mit der Betätigung des Starttasters zuerst alle 7 LEDs für 3 Sekunden an sein. Danach leuchtet nur noch die mittlere LED Ausgleich (Spielstand A:B = 0:0) und die Anzeige arbeitet wie zuvor beschrieben. Der Punktgewinn von A und B wird mittels prellfreier low-aktiver Taster TA und TB realisiert, wobei immer nur einer der beiden Taster betätigt wird.

1.3 Zeitprogramm 3 Sekunden

Die 3 Sekunden für die Funktionskontrolle der LEDs sollen mit einer verschachtelten Zeitschleife realisiert werden.

- 1.3.1 Entwickeln Sie auf der Basis verschachtelter Schleifen in Assembler ein Unterprogramm wait_3s, das ca. 3 Sekunden dauert. Kommentieren Sie die Programmzeilen sinnvoll und begründen Sie die Startwerte der Zählregister. Wählen Sie als Zeitbasis den von Ihnen verwendeten Mikrocontroller wie z.B. den µC8051 mit einer Maschinenzkluszeit von 1 µs. 4
- 1.3.2 Warum ist eine entsprechende Realisierung in einer höheren Programmiersprache wie z.B. C nicht möglich, bzw. schwierig. 1
- 1.3.3 Nennen und Begründen Sie eine alternative Lösung, bei der die Wahl der Sprache keine Rolle spielt. 1

1.4 LED-Ausgabe mittels Tabelle (Feld, array)

Der aktuelle Spielstand ("Punktendifferenz" A / B) wird mit einer Zählvariablen, bzw. einem Register (im Folgenden X genannt) beschrieben. Mit Betätigung der Starttaste wird X auf 3 (Ausgleich) gesetzt. Ist X bei 0 angelangt, hat Mannschaft A 3 Punkte Vorsprung und gewonnen, bei X=6 entsprechend Mannschaft B. Bei Spielende (X=0 oder X=6) zeigen die Taster A und B keine Wirkung. Schreiben Sie ein Unterprogramm **ausgabe** in Assembler oder eine Funktion **ausgabe()** in C, das die Ausgabe der LEDs anhand von X mittels Tabelle (Feld/array) realisiert. Geben Sie dabei auch die Tabelle (Feld/array) an und kommentieren Sie ihre Programmzeilen sinnvoll. 4

Es gilt die folgende Pinzuordnung:

Anzeige	nc.	B3	B2	B1	Ausgleich	A1	A2	A3
Port N	N.7	N.6	N.5	N.4	N.3	N.2	N.1	N.0

1.5 Tasterabfrage mittels externem Interrupt

Die beiden Taster TA und TB sollen mittels externem Interrupt erfasst werden.

- 1.5.1 Ordnen Sie beide Taster geeigneten Portpins ihres µCs zu und schreiben Sie die Initialisierung beider Interrupts. Kommentieren Sie ihre Befehlszeilen sinnvoll. 3
- 1.5.2 Beide Interrupt-Service-Routinen (ISR) haben die Aufgaben den Wert X (siehe Aufgabe 1.4) bis minimal 0 um 1 zu erniedrigen (Taster TA) bzw. bis maximal 6 um 1 zu erhöhen (Taster TB) und das veränderte Ergebnis mit dem Unterprogramm (der Funktion) **ausgabe** anzuzeigen. 3
- Zeichnen Sie einen PAP oder ein Struktogramm, für eine der beiden ISRs.
- 1.5.3 Erklären Sie den Begriff Vektoradresse (Einsprungsadresse) und dessen Bedeutung für die Programmierung von Interrupts. Worin besteht diesbezüglich der Unterschied zu einem Unterprogramm (einer Funktion)? 1

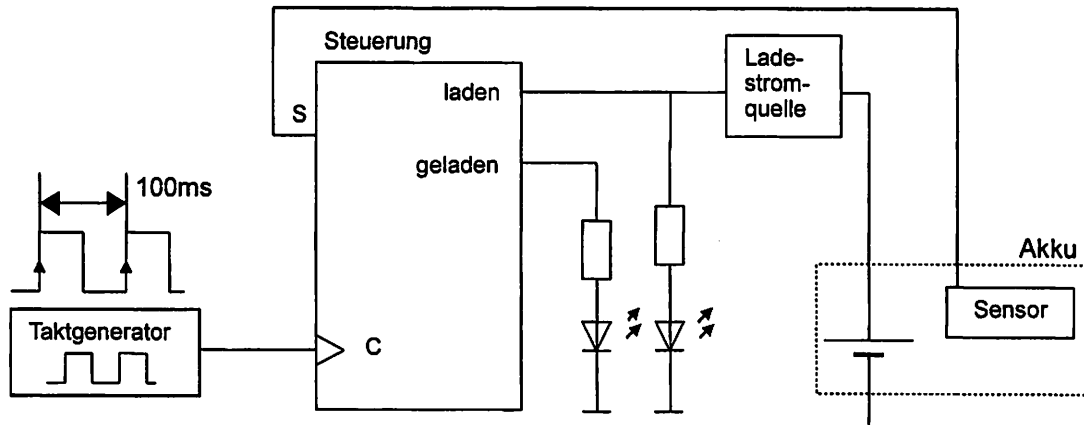
30

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Hardware)	
	Teil: 1 (Pflichtbereich)	Aufgabe: 2 (3Seiten)

Punkte

Ladegerät

Ein Akkuhersteller möchte für seine Akkus ein eigenes Impulsladegerät herstellen. Hierzu wurde folgendes Blockschaltbild entwickelt:

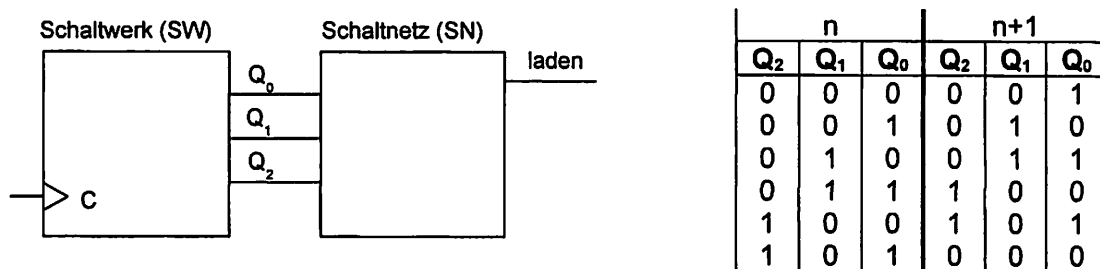


Das Ladegerät besteht aus einem **Taktgenerator**, der **Steuerung**, **LEDs** zur Anzeige und einer **Ladestromquelle**. Die Akkus werden an das Ladegerät angeschlossen. Ein Sensor am Akku dient zur Überwachung und meldet wenn der **Akku voll** ($S=1$) geladen ist.

2.1.1 Berechnen Sie die **Taktfrequenz** des Taktgenerators ($T = 100\text{ms}$).

1

2.1.2 Der erste Entwurf der Steuerung berücksichtigt die Steuerleitung S nicht (\Rightarrow zyklisches Laden mit Ladepause – ohne Abschaltung). Er besteht aus einem Schaltwerk und einem Schaltnetz und wird durch folgendes Blockschaltbild und einer Zustandsübergangstabelle beschrieben:



Wie viele Zustände hat das Schaltwerk?

1

2.1.3 Der Ausgang „laden“ soll **100ms** aus- und **500ms** angeschaltet sein.

3

Geben Sie die Funktionstabelle und die Funktionsgleichung für das **Schaltnetz**, das mit den Eingangssignalen $Q_0 \dots Q_2$ dem Ausgang „laden“ an.

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Hardware)	
	Teil: 1 (Pflichtbereich)	Aufgabe: 2 (3Seiten)

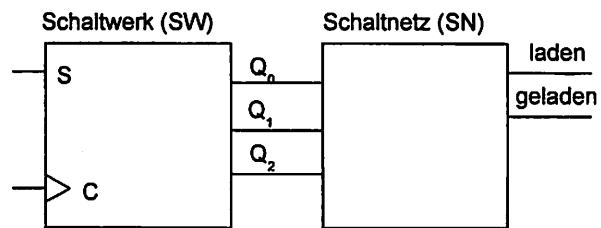
Punkte

- 2.1.4 In einem zweiten Entwurf wird auch der Eingang „S“ des Schaltwerks einbezogen. Der Akku soll solange geladen werden, bis der Sensor ein „1“-Signal abgibt. Nach diesem „1“-Signal soll sofort nach der nächsten positiven Taktflanke die LED „laden“ aus - und die LED „geladen“ angehen.

5

Das Ladegerät soll dann im Zustand „geladen“ bleiben, auch wenn der Sensor nach einiger Zeit wieder ein 0-Signal abgibt.

Zeichnen Sie ein Zustandsdiagramm mit Angabe der **Ausgabewerten** in den einzelnen Zuständen, das diesen Entwurf beschreibt.



- 2.1.5 Bei Experimenten mit dem Ladegerät hat sich ergeben, dass es besser ist einen Lade_Impuls von **1s** und eine Lade_Pause von **50ms** zu haben.

2

Beschreiben und begründen Sie, durch welche Veränderungen am Ladegerät dieses Impuls-Pausenverhältnis möglich ist.

Mikrocontollertechnik

Die Steuerung des Ladegeräts wird nun mit einem Mikrocontroller durchgeführt. Dabei soll der Ladeimpuls **1s** dauern und die Zeit für die Ladepause **50 ms** betragen.

Der Ladeimpuls (**1s**) und die Ladepause (**50ms**) soll durch die Interruptserviceroutine eines Timers bestimmt werden.

- 2.2.1 Beschreiben Sie zunächst stichwortartig, wie man mit Hilfe eines 8- oder 16-Bit Timers bei einer Zeitbasis von 1µs Zeiten im Sekundenbereich (z.B. 1 Sekunde) realisieren kann.

3

Erstellen Sie die folgenden Programmteile:

Hinweis: Geben Sie den Programmcode (in Assembler oder „C“) an und kommentieren Sie Ihr Listing ausführlich und sinnvoll.

- 2.2.2 Initialisierung des Timers und Freigabe des zugehörigen Interrupts

2

- 2.2.3 Interruptserviceroutine (ISR) für das geforderte Impuls-Pausenverhältnis. Diese ISR ist auch für den Ausgang „laden“ verantwortlich!

2

- 2.2.4 Beschreiben Sie stichwortartig, wie ein Timerinterrupt in einem Mikrocontroller abgearbeitet wird.

2

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Hardware)	
	Teil: 1 (Pflichtbereich)	Aufgabe: 2 (3Seiten)

Punkte

- 2.3 Die Temperatur des Akkus wird mit Hilfe eines A/D-Wandlers gemessen und steht am einem 8-Bit Port (z.B. P1) zur Verfügung.

Temperaturtabelle:

Akkutemperatur in °C	Wert an Port1
10	E8h
20	C3h
30	9Bh
40	78h
50	63h

Ein Unterprogramm <Pruef_Temp> soll ein Bit <Temp_OK> auf „1“ setzen wenn die Akkutemperatur zwischen 10°C und 40°C liegt.

3

Sonst soll das Bit <Temp_OK> auf „0“ gesetzt werden.

Entwerfen Sie einen Programmablaufplan (PAP) oder ein Struktogramm für das Unterprogramm <Pruef_Temp>.

- 2.4 Für einen anderen Akkutyp wird das Ladegerät weiterentwickelt. Die Abschaltung des Ladestroms soll zusätzlich über eine Spannungsmessung erfolgen. In jeder Ladepause wird eine Spannungsmessung durchgeführt.

Die Messwerte (8-Bit) werden im Datenspeicher des Mikrocontrollers wie folgt abgelegt.

Speicherorganisation:

AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8	AD9	AD10
neu									alt

} aktuellster Wert
} ältester Wert

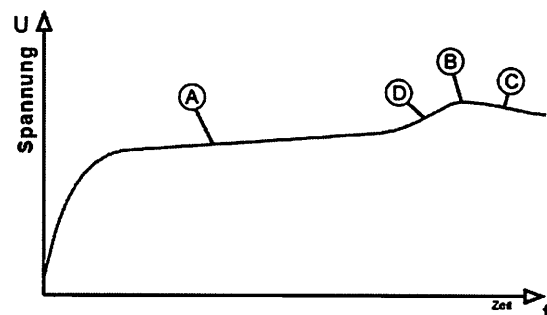
- 2.4.1 Erstellen Sie ein Unterprogramm <U_speichern> in Assembler oder „C“, das den Spannungsmesswert, der am Port zur Verfügung steht in den Datenspeicher schreibt. Berücksichtigen Sie, dass immer die zehn aktuellsten Messwerte zur Verfügung stehen müssen.

4

- 2.4.2 Im Schaubild ist der Verlauf der Akkuspannung während eines Ladevorgangs über der Zeit aufgetragen.

Die wichtigen Punkte der Akkuladung sind im nebenstehenden Schaubild markiert (A, B, C, D).

Der Ladevorgang soll zwischen Punkt B und Punkt C beendet werden.



Entwerfen Sie mit Hilfe der im Datenspeicher zur Verfügung stehenden Spannungsmesswerte einen Algorithmus zur Abschaltung des Ladevorgangs. Beschreiben Sie diesen mit ihren eigenen Worten.

2

30

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (6 Seiten)

Punkte

Aufgabe 3 Computerspiel 'Shut the Box'

Das Computerprogramm 'Shut the Box' soll für zwei Spieler entworfen werden. Mit diesem Programm sollen Kinder das Addieren von ganzen Zahlen spielerisch erlernen. Das Bild 1 zeigt die Spieloberfläche mit den Spieldaten ❶, den Klappen ❷, den Tasten 'würfeln' und 'Spielerwechsel' ❸ sowie der Würfelanzeige ❹. Die Klappen sind von 1 bis 9 durchnummeriert. Jeder Klappe ist der Wert zugeordnet, der auf ihr angezeigt ist. Im Bild unten hat ein Spieler bereits die Klappe 8 geschlossen, somit wird auf dieser nur ein 'X' angezeigt.

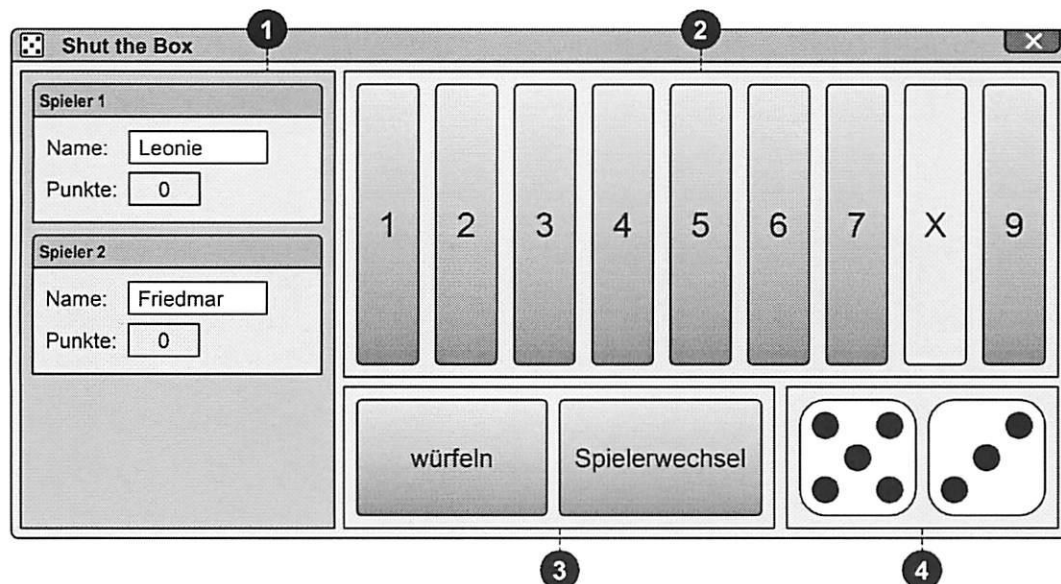


Bild 1

Spielziel: Ziel des Spiels ist es, alle Klappen zu schließen.

Spielstart: Alle Klappen von 1 bis 9 sind geöffnet, die Werte (1 - 9) aller Klappen werden angezeigt. Die Punkteähler der zwei Spieler sind auf Null gestellt.

Spielverlauf: Der Startspieler klickt auf die Taste 'würfeln' und generiert damit zwei Zufallszahlen, die mit der Würfelanzeige dargestellt werden. Die Würfeltaste wird deaktiviert. Nun muss der Spieler eine oder mehrere Klappen anklicken. Dabei muss die Summe der auf den Klappen angegebenen Werte mit der Augensumme der Würfel übereinstimmen. Die Klappen werden nach dem Anklicken geschlossen und sind danach nicht mehr anklickbar. Der Spieler entscheidet, auf welche Klappen er die Augensumme der Würfel aufteilt. **Beispiele:**

Augenzahlen	mögliche Klappen, die geschlossen werden können
1 + 1	2
1 + 2	1 + 2 oder 3
2 + 2	1 + 3 oder 4
2 + 3	1 + 4 oder 2 + 3 oder 5
3 + 3	1 + 5 oder 2 + 4 oder 1 + 2 + 3 oder 6

Hat der Spieler durch das Schließen der Klappen die passende Augensumme erreicht, wird die Würfeltaste wieder aktiviert und er kann erneut würfeln. Die bereits geschlossenen Klappen bleiben geschlossen. Ist es dem Spieler nicht möglich, die Augensumme der Würfel auf die noch offenen Klappen aufzuteilen oder sind alle Klappen geschlossen, so ist der Zug beendet. Er kann nur noch auf die Taste 'Spielerwechsel' klicken. Die Summe der Werte auf den noch offenen Klappen wird dem Spieler als Minuspunkte angerechnet und auf der Punkteanzeige dargestellt. Nun ist der nächste Spieler am Zug, alle Zahlenklappen werden wieder geöffnet. Kann auch dieser Spieler die Augensumme der Würfel nicht mehr auf die noch offenen Klappen aufteilen oder hat er alle Klappen geschlossen, so ist das Spiel beendet. Der Spieler mit den wenigsten Minuspunkten hat gewonnen.

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (6 Seiten)

Punkte

Beispiel: Spieler 1 würfelt die Augensumme 9, siehe Bild 2. Es sind nur noch die Klappen 1, 3, 4 und 7 offen. Mit diesen Klappen kann die angezeigte Augensumme nicht summiert werden. Spieler 1 klickt die Taste 'Spielerwechsel', die Werte der noch offenen Klappen ($1 + 3 + 4 + 7 = 15$) werden anschließend dem Spieler 1 als Minuspunkte angerechnet. Danach werden für Spieler 2 alle Klappen wieder geöffnet und er kann würfeln.



Bild 2

3.1 Analyse des Klassendiagramms

3.1.1 Der Name der Spieler darf während eines Spiels nicht mehr geändert werden. Erläutern Sie anhand der Klasse *Spieler* (Arbeitsblatt 1, Seite 5), wie diese Vorgabe umgesetzt wird. 2

3.1.2 Erläutern Sie den Begriff Datenkapselung anhand der Klasse *Spieler*. 1

3.2 Klassendiagramm ergänzen

Das Sequenzdiagramm im Bild 3 wurde bereits für die folgenden Szenarien entwickelt:

- Spieler 1 klickt auf die Taste 'würfeln' und würfelt die Zahlen 3 und 5.
- Danach klickt Spieler 1 auf die offene Klappe mit der Zahl 8.

Mögliche Nebenszenarien sind nicht eingezeichnet.

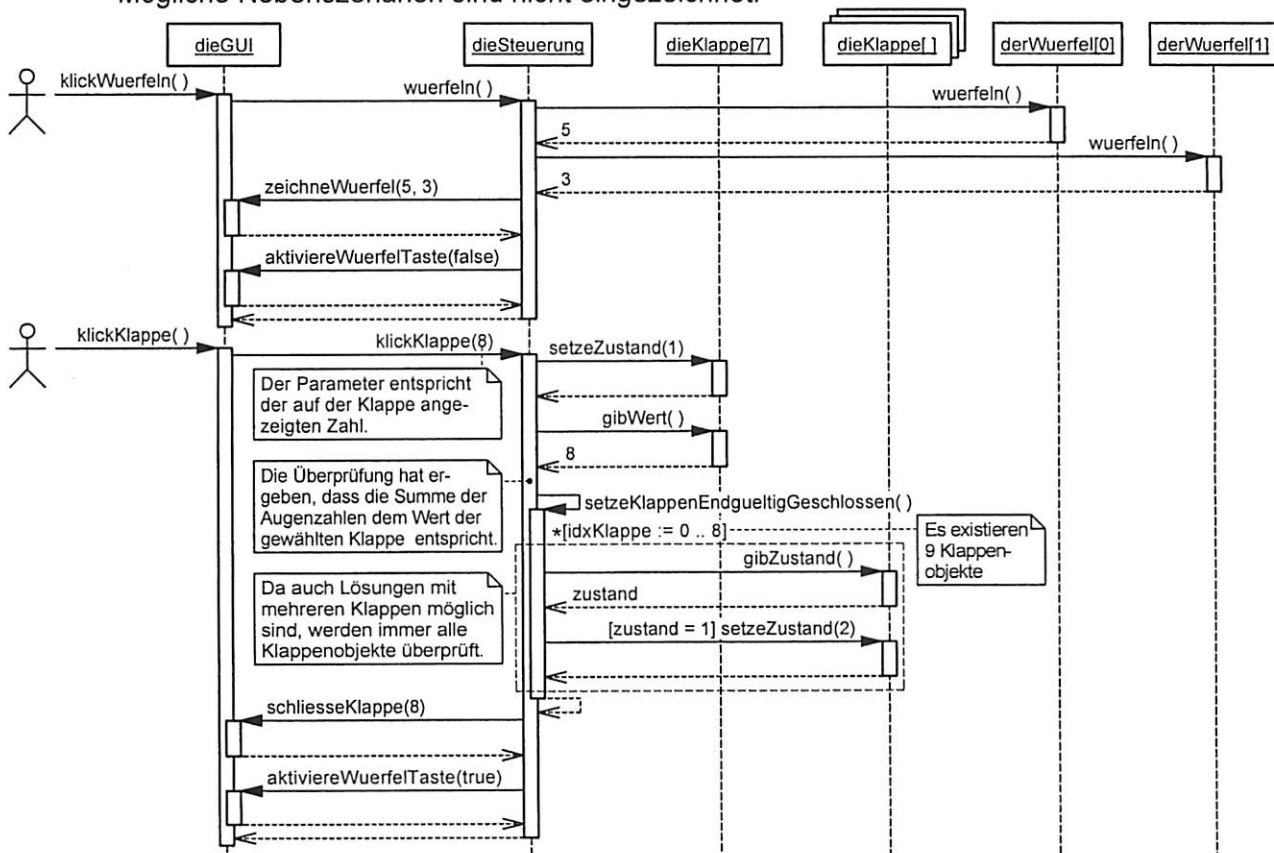


Bild 3

Zum Verständnis: Klappenobjekte besitzen drei Zustände. Offene Klappen haben den Zustandswert 0. Wurde eine Klappe geschlossen und der Spielzug ist noch nicht vorbei ($\sum \text{Klappenwerte} < \sum \text{Augenzahlen}$), erhält diese den Zustandswert 1. Damit wird sichergestellt, dass keine falschen Werte in die Punkteberechnung einfließen. Eine endgültig geschlossene Klappe erhält den Zustandswert 2.

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (6 Seiten)

Punkte

- 3.2.1 Ein Objekt der Klasse *Steuerung* erhält bei seiner Erzeugung eine Referenz auf die GUI sowie die beiden Spielernamen. Die Objekte der Klasse *Klappe* erhalten bei ihrer Erzeugung ihren Wert (1..9). Ergänzen Sie auf dem Arbeitsblatt 1 die Konstruktoren der Klassen *Steuerung* und *Klappe* mit vollständiger Signatur und Sichtbarkeit. 2
- 3.2.2 Überprüfen Sie, ob alle Botschaften des Sequenzdiagramms (Bild 3) im Klassendiagramm auf dem Arbeitsblatt 1 angegeben sind. Ergänzen Sie auf dem Arbeitsblatt die im Klassendiagramm fehlenden Operationen. Dabei ist die vollständige Signatur, der Rückgabebetyp und die Sichtbarkeitskennzeichnung anzugeben. Die Sichtbarkeit öffentlich ist nur dann zu vergeben, wenn das für die Kommunikation zwischen den Objekten notwendig ist. 3
- 3.2.3 Ergänzen Sie auf dem Arbeitsblatt 1 die Assoziationen, die aus dem Sequenzdiagramm im Bild 3 ersichtlich sind. Die Assoziationen sind mit Navigationsrichtungen, Rollennamen und Kardinalitäten anzugeben. Begründen Sie Ihre Wahl der Richtung und Kardinalität der Assoziation. 3
- 3.3 **Struktogramm aus Sequenzdiagramm ableiten** 2
- Leiten Sie den Algorithmus der Operation *setzeKlappenEndgueltigGeschlossen()* mit Hilfe der Informationen des Sequenzdiagramms im Bild 3 ab und stellen Sie diesen in einem Struktogramm dar.
- 3.4 **Sequenzdiagramm aus Struktogramm ableiten** 5

Ergänzen Sie das Sequenzdiagramm auf dem Arbeitsblatt 2 für das Szenario „Spieler 1 kann keine weiteren Klappen mehr schließen und klickt auf die Taste 'Spielerwechsel' ". Für den aktiven Spieler 1 gilt: *dieSteuerung.aldxAktiverSpieler* = 0. Spieler 1 hat 21, Spieler 2 hat 25 (Minus-) Punkte. Für Spieler 1 erhöhen sich für dieses Szenario die Punkte auf 36. Leiten Sie das zu erstellende Sequenzdiagramm aus dem Struktogramm der Operation *Steuerung.wechsleSpieler()* ab, siehe Bild 4. Tragen Sie, soweit möglich, für die Parameter und Rückgabewerte der Botschaften im Sequenzdiagramm konkrete Werte ein.

Struktogramm zur Operation *Steuerung.wechsleSpieler()*

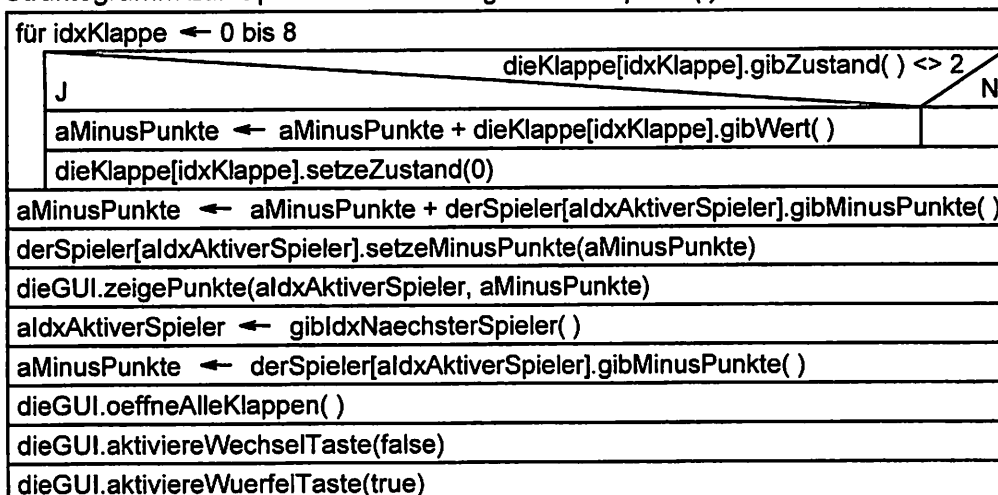


Bild 4

3.5 Weiterentwicklung des Klassendiagramms

In einer neuen Spielvariante soll der Computer einen Spieler ersetzen, falls kein menschlicher Spieler zur Verfügung steht. In dieser Version soll eine neue Klasse *Spieler* Superklasse für die Subklassen *Computer* und *Mensch* sein. Die Klasse *Computer* erhält ein neues Attribut *aSpielstaerke* (Wertebereich 1 .. 3), Objekte der Klasse *Computer* können jedoch keinen Namen speichern. Die Klasse *Mensch* besitzt alle Eigenschaften der Klasse *Spieler* auf dem Arbeitsblatt 1.

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (6 Seiten)

Punkte

3.5.1 Entwerfen Sie auf einem Lösungsblatt die Vererbungshierarchie für die Klassen *Spieler*, *Computer* und *Mensch*. Beachten Sie folgende Regeln: 3

- Alle Attribute werden bei der Objekterzeugung initialisiert.
- Die Attributwerte für die Spielstärke und für den Spielernamen können nach der Objekterzeugung nicht mehr geändert werden.
- Alle anderen Attributwerte können nach der Objekterzeugung verändert werden.

3.5.2 Superklassen können abstrakt oder konkret deklariert sein. Begründen Sie, weshalb Sie Ihre Superklasse *Spieler* in Aufgabe 3.5.1 abstrakt oder konkret deklariert haben. 2

3.6 Struktogramm weiter entwickeln 7

In der Spielstärke 1 kann der Computer nur zwei sehr einfache Lösungen für das Schließen der Klappen errechnen. Die zwei folgenden Beispiele zeigen die beiden Lösungsmöglichkeiten der Spielstärke 1 für die Augenzahlen 3 und 4 auf:

1. Lösungsmöglichkeit: Klappenwert = Augensumme
Augenzahlen 3 + 4 → Computer schließt Klappe 7
2. Lösungsmöglichkeit: Klappenwerte = Augenzahlen
Augenzahlen 3 + 4 → Computer schließt Klappe 3 und Klappe 4
Weitere Lösungen, z.B. das Schließen der Klappen 2 und 5, werden bei Spielstärke 1 nicht erkannt.

computerZug() : Boolean (nicht vollständig)

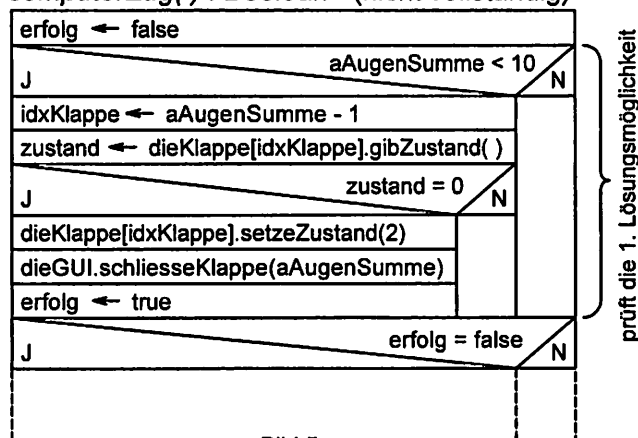


Bild 5

Der Algorithmus der Operation *computerZug()* : Boolean der Klasse *Steuerung* soll für die Spielstärke 1 weiter entwickelt werden.

Das Bild 5 zeigt den Anfang des Algorithmus mit der Prüfung der 1. Lösungsmöglichkeit. Übertragen Sie die im Bild 5 unten gegebene Verzweigung auf Ihr Lösungsblatt und ergänzen Sie den Algorithmus zur Prüfung der 2. Lösungsmöglichkeit sowie die Wertrückgabe.

Folgende Aspekte sind zu beachten:

- Die aktuelle Augensumme ist im Attribut *aAugenSumme* der Klasse *Steuerung* gespeichert. Die Augenzahlen der Würfel sind bei den Würfelobjekten zu erfragen.
- Augensummen größer als 9 können nicht mit einer einzigen Klappe erreicht werden.
- Nur Klappen, die den Zustand 0 besitzen, können geschlossen werden.
- Besitzen beide Würfel die gleiche Augenzahl (z.B. 4 + 4), so kann die 2. Lösungsmöglichkeit nicht angewandt werden.
- Die Operation *GUI.schliesseKlappe(pWert : GZ)* schließt die Klappe mit dem im Parameter *pWert* übergebenen Wert.
- Der Zustand einer zu schließenden Klappe muss auf 2 gesetzt und die Klappe auf der Oberfläche geschlossen werden.
- Konnte der Computer die Aufgabe lösen ($\sum \text{Klappenwerte} = \sum \text{Augenzahlen}$), so wird *true* zurückgegeben, sonst *false*.

Zu- und Vorname:

Hauptprüfung 2012/2013	
1.5.2	Informationstechnik

Schulnummer	Schülernummer

Hauptprüfung 2012/2013	
1.5.2	Informationstechnik
Arbeitsblatt 1	Teil: 2 Aufgabe: 3

Schulnummer	Schülernummer

Arbeitsblatt 1

Spieler

- aName : Text

- aMinusPunkte : GZ

+ Spieler(pName : Text) <<constructor>>

+ gibName() : Text

+ gibMinusPunkte() : GZ

+ setzeMinusPunkte(pMinusPunkte : GZ)

Wuerfel

- aLetzterWurf : GZ

+ wuerfeln() : GZ

+ gibLetzterWurf() : GZ

Klappe

- aWert : GZ

- aZustand : GZ

+ gibZustand() : GZ

2
derSpieler[]

GUI

...

+ GUI(pName1 : Text, pName2 : Text) <<constructor>>

+ zeigePunkte(pIdxSpieler : GZ, pMinusPunkte : GZ)

+ oeffneAlleKlappen()

+ schliesseKlappe(pWert : GZ)

+ aktiviereWuerfelTaste(pAktiv : Boolean)

+ aktiviereWechselTaste(pAktiv : Boolean)

+ klickWechsleSpieler()

+ klickWuerfeln()

...

Steuerung

- aAugenSumme : GZ

- aKlappenSumme : GZ

- aIdxAktiverSpieler : GZ

- aMinusPunkte : GZ

...

+ wechsleSpieler()

- gibIdxNaechsterSpieler() : GZ

- setzeKlappenEndgueltigGeschlossen()

+ computerZug() : Boolean

...

Zu- und Vorname:

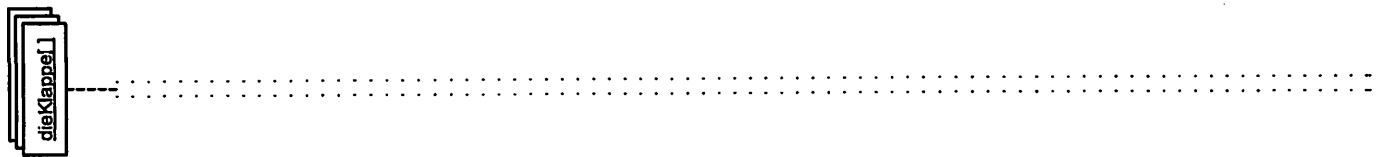
Hauptprüfung 2012/2013	
1.5.2	Informationstechnik

Schulnummer	Schülernummer

Hauptprüfung 2012/2013	
1.5.2	Informationstechnik
Arbeitsblatt 2	Teil: 2 Aufgabe: 3

Schulnummer	Schülernummer

Arbeitsblatt 2



Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 4 (6 Seiten)

Punkte

4.1 Grundlagen der Objektorientierung

Ein System, das durch das Klassendiagramm in **Bild 1** beschrieben wird, verfügt über die in **Bild 2** dargestellten Objekte.

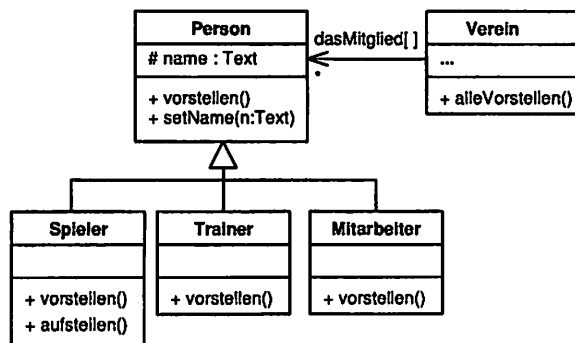


Bild 1

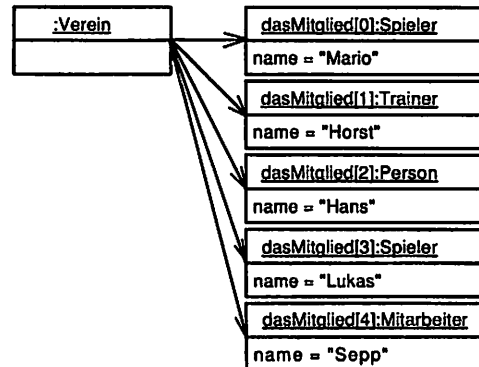


Bild 2

Die Struktogramme in **Bild 3** beschreiben die Implementierungen einiger Operationen.

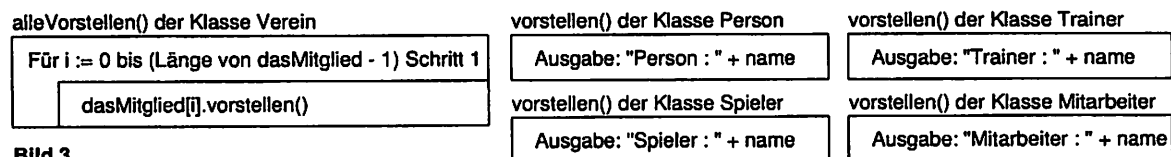


Bild 3

- 4.1.1 Ermitteln Sie, welche Ausgabe auf dem Bildschirm erfolgt, wenn die Operation *alleVorstellen()* aufgerufen wird. 1
- 4.1.2 Nennen und beschreiben Sie das Grundkonzept der Objektorientierung, das die Ausgabe in 4.1.1 ermöglicht. 2
- 4.1.3 Erläutern Sie, warum das Compilieren eines Programms mit der Botschaft *dasMitglied[i].aufstellen()* nicht möglich wäre. 2
- 4.1.4 "Wenn die Klasse *Person* abstrakt deklariert wäre, würde das Objektdiagramm in Bild 2 anders aussehen". Beurteilen Sie diese Aussage. 1
- 4.1.5 "Eine abstrakte Deklaration der Klasse *Person* wäre sinnvoll". Nehmen Sie zu dieser Aussage Stellung. 2

4.2 Computerspiel „Bundesligamanager“

In einer stark vereinfachten Version des Spiels „Bundesligamanager“ führt der Benutzer die Geschehnisse eines Bundesligafußballvereins und seiner genau 18 Fußballspieler. Dazu stellt der Benutzer vor jedem neuen Bundesligaspiel 11 seiner 18 Spieler auf. Die Benutzungsoberfläche hat den in **Bild 4** gezeigten Aufbau.

In der ersten Zeile steht der Name des eigenen Vereins, hier SC Freiburg. Darunter werden tabellarisch die 18 Spieler aufgelistet. Die Tabelle enthält folgende Spalten:

- Nummer: fortlaufende Nummerierung der Spieler
- Name: Name des Spielers
- Position: T für Torwart, V für Verteidiger, M für Mittelfeldspieler, S für Stürmer
- Fitness: Fitnesswerte von 0 bis 99. Je höher der Wert, desto leistungsfähiger der Spieler
- Gelb: Anzahl der Gelben Karten seit Saisonbeginn oder der letzten Gelbsperre

MINISTERIUM FÜR KULTUS, JUGEND UND SPORT BADEN-WÜRTTEMBERG

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 4 (6 Seiten)

Punkte

SC Freuberg							Nächstes Spiel gegen FC Schlacke 05 Spiel starten
Nr.	Name	Pos.	Fitness	Gelb	Status	Aufgestellt	
1	Dohmen	T	82	1	ok	✓	
2	Leone	T	77	2	ok		
3	Glaser	V	56	4	ok	✓	
4	Graziano	V	0	3	V		
5	Becker	V	98	0	ok	✓	
6	Lott	V	78	0	ok	✓	
7	Riese	V	44	4	ok	✓	
8	Knoop	V	61	5	GK		
9	Leser	M	0	0	V		
10	Haug	M	77	1	ok		
11	Klemm	M	81	1	ok	✓	
12	Adler	M	83	3	ok	✓	
13	Vierthaler	M	80	1	ok	✓	
14	Böhler	M	71	0	ok	✓	
15	Ruch	S	72	0	ok		
16	Götz	S	75	3	ok	✓	
17	Baro	S	74	4	ok		
18	Brombacher	S	99	2	ok	✓	

Bild 4

Status:

Status des Spielers, dabei steht

- "ok" für einsetzbar
- "V" für verletzt, deshalb nicht einsetzbar (z.B. Spieler Graziano und Leser)
- "GK" für Gelbsperre und deshalb nicht einsetzbar (z.B. Spieler Knoop)
- "RK" für Rotsperre und deshalb nicht einsetzbar

Aufgestellt: Hier markiert der Benutzer, welche Spieler im nächsten Spiel (im Beispiel gegen den FC Schlacke 05) aufgestellt werden sollen.

Sobald der Benutzer die Schaltfläche "Spiel starten" betätigt, wird aus der Stärke seiner Mannschaft (also aus den Fitnesswerten der elf aufgestellten Spieler) und aus der Stärke des Gegners das Endergebnis berechnet und ausgegeben. Wie dies geschieht, ist für diese Aufgabe nicht von Belang. Ein mögliches Ergebnis gibt Bild 5 wieder.

SC Freuberg							Endergebnis: SC Freuberg - FC Schlacke 05 3:1 Neu Aufstellen
Nr.	Name	Pos.	Fitness	Gelb	Status	Aufgestellt	
1	Dohmen	T	79	1	ok	✓	
2	Leone	T	80	2	ok		
3	Glaser	V	52	4	ok	✓	
4	Graziano	V	20	3	ok		
5	Becker	V	93	0	ok	✓	
6	Lott	V	77	0	ok	✓	
7	Riese	V	41	4	ok	✓	
8	Knoop	V	64	0	ok		
9	Leser	M	0	0	V		
10	Haug	M	79	1	ok		
11	Klemm	M	77	2	ok	✓	
12	Adler	M	80	3	ok	✓	
13	Vierthaler	M	77	1	ok	✓	
14	Böhler	M	70	0	ok	✓	
15	Ruch	S	75	0	ok		
16	Götz	S	72	3	ok	✓	
17	Baro	S	76	4	ok		
18	Brombacher	S	0	2	V	✓	

Bild 5

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 4 (6 Seiten)

Punkte

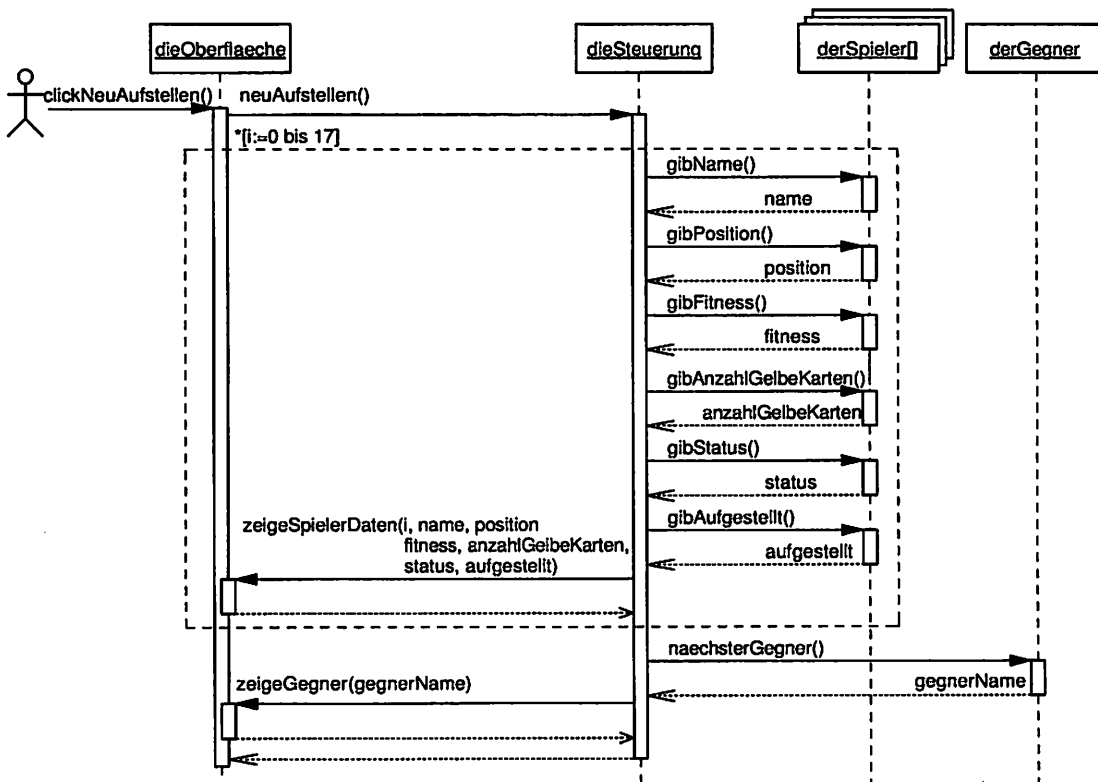
Die Schaltfläche "Spiel starten" ist nun deaktiviert. Die Fitness der aufgestellten Spieler hat sich reduziert, während sich die Fitness der pausierenden Spieler erhöht hat. Der Spieler Graziano ist wieder gesund, während Leser weiterhin verletzt ist. Neu verletzt hat sich der Spieler Brombacher. Der Spieler Knoop hat seine Gelbsperre abgesehen und ist im nächsten Spiel wieder einsetzbar. Der Spieler Klemm hat im Spiel gegen den FC Schlacke 05 eine gelbe Karte kassiert.

Nach Drücken der Befehlsschaltfläche "Neu Aufstellen" kann der Benutzer die elf Spieler für das nächste Bundesligaspiel aufstellen.

4.2.1 Klassendiagramm anhand eines Sequenzdiagrammes weiterentwickeln

Im **Sequenzdiagramm 1** wird das Szenario "Der Benutzer betätigt die Befehlsschaltfläche 'Neu Aufstellen'" gezeigt.

- 4.2.1.1 Ergänzen Sie im Klassendiagramm auf dem **Arbeitsblatt 1** alle fehlenden Operationen, die für das **Sequenzdiagramm 1** benötigt werden. Alle Parameter und Datentypen müssen ebenso wie die Sichtbarkeiten vollständig angegeben werden. Von außen dürfen nur die Operationen sichtbar sein, die für die Kommunikation zwischen den Objekten benötigt werden. 4
- 4.2.1.2 Ergänzen Sie auf dem **Arbeitsblatt 1** die Assoziationen mit Rollennamen und Kardinalitäten, die sich aus dem Sequenzdiagramm ergeben. 3



Sequenzdiagramm 1

Hauptprüfung 2012/2013	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik (Software)	
	Teil: 2 (Pflichtbereich)	Aufgabe: 4 (6 Seiten)

Punkte

4.2.2 Algorithmus entwickeln

8

Die Operation *ueberpruefeAufstellung()* der Klasse *Steuerung* überprüft nach dem Betätigen der Befehlsschaltfläche "Spiel starten", ob die aktuelle Aufstellung des Benutzers korrekt ist. Die Information, ob ein Spieler vom Benutzer aufgestellt wurde, befindet sich in der Benutzungsoberfläche und muss noch beim Spieler gespeichert werden. Die Aufstellung ist korrekt, wenn alle folgenden Bedingungen erfüllt sind:

- Es wurden genau 11 Spieler aufgestellt.
- Es wurden nur Spieler aufgestellt, deren Status "ok" ist.
- Es wurde genau ein Torwart aufgestellt.

Nach diesen Kriterien ist die Aufstellung in **Bild 4** korrekt. Die Aufstellung in **Bild 5** wäre nicht korrekt, da mit Brombacher ein verletzter Spieler aufgestellt wurde. Im Falle einer korrekten Aufstellung liefert *ueberpruefeAufstellung()* *true* zurück, andernfalls *false*.

Entwerfen Sie basierend auf diesen Angaben und den Operationen des Klassendiagramms den Algorithmus für die Operation *ueberpruefeAufstellung()* als Struktogramm.

Hinweise: Zur Vereinfachung können Sie davon ausgehen, dass immer mindestens 11 einsetzbare Spieler zur Verfügung stehen. Es ist auch immer mindestens ein Torhüter einsetzbar.

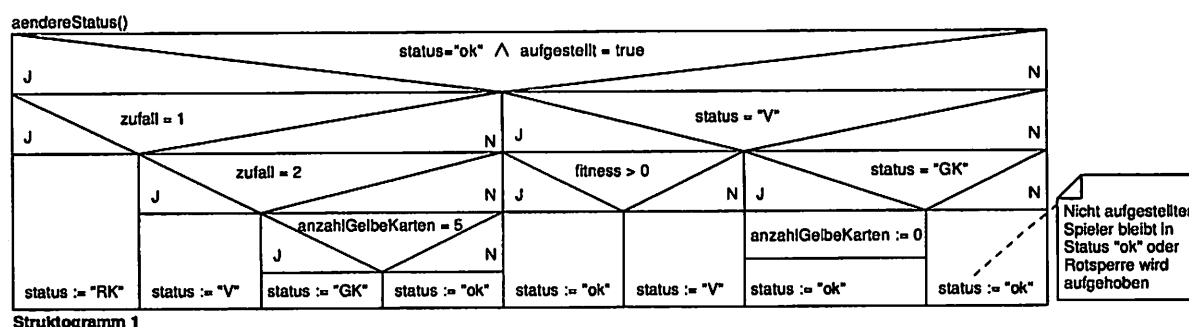
4.2.3 Zustandsdiagramm entwickeln

7

Der Status (Zustand) eines Spielers kann zwischen den Werten "ok", "V" (verletzt), "GK" (gesperrt wegen fünf Gelber Karten) und "RK" (gesperrt wegen Roter Karte) wechseln. Nach jedem Spieltag kann sich der Zustand des Spielers ändern. Bei der Simulation eines Spieles werden für jeden Spieler neue Werte für die Attribute *fitness*, *anzahlGelbeKarten* und *zufall* durch einen Algorithmus ermittelt. Die Details dieses Algorithmus sind für die Lösung der Aufgabe nicht relevant.

Die Operation *aendereStatus()* legt dann für jeden einzelnen Spieler den Status aufgrund der drei oben genannten Attributwerte fest. Der Algorithmus dieser Operation ist in

Struktogramm 1 dargestellt.



MINISTERIUM FÜR KULTUS, JUGEND UND SPORT BADEN-WÜRTTEMBERG

Zu- und Vorname:

Hauptprüfung 2012/2013	
1.5.2	Informationstechnik (Software)

Schulnummer	Schülernummer

X

Hauptprüfung 2012/2013	
1.5.2	Informationstechnik (Software)
Arbeitsblatt	Teil: 2 Aufgabe: 4

Schulnummer	Schülernummer

Punkte

Arbeitsblatt 1

Spieler
<ul style="list-style-type: none"> - name : Text - position : Text - fitness : GZ - anzahlGelbeKarten : GZ - status : Text - aufgestellt : Boolean - zufall : GZ
<pre> <<create>> + Spieler(pName : Text, pPos : Text) + gibName() : Text + gibPosition() : Text + gibFitness() : GZ + setzeAufgestellt(pAufgestellt : Boolean) + spiele() + aendereStatus() + ermittleFitness() - ermittleAnzahlGelbeKarten() - ermittleZufall() </pre>

Gegner
<ul style="list-style-type: none"> - name[] : Text

Oberflaeche
...
<pre> <<create>> + Oberflaeche(dieSteuerung : Steuerung) + clickNaechstesSpiel() + clickNeuAufstellen() + gibAufgestellt(pNr : GZ) : Boolean + zeigeFalscheAufstellung() + zeigeSpielErgebnis(pTore:GZ, pGegentore:GZ) + zeigeVereinsnamen(pVName : Text) </pre>

Steuerung
...
<pre> <<create>> + Steuerung() + initialisiere() + naechstesSpiel() + neuAufstellen() + berechneTore() : GZ + berechneGegentore() : GZ + erzeugeMeldung(pTore : GZ, pGegentore : GZ) + ueberpruefeAufstellung() : Boolean </pre>

MINISTERIUM FÜR KULTUS, JUGEND UND SPORT BADEN-WÜRTTEMBERG

Zu- und Vorname:

Hauptprüfung 2012/2013	
1.5.2	Informationstechnik (Software)

Schulnummer	Schülernummer

✂

Hauptprüfung 2012/2013		Schulnummer	Schülernummer
1.5.2	Informationstechnik (Software)		
Arbeitsblatt	Teil: 2	Aufgabe: 4	

Punkte

Arbeitsblatt 2

