

Technische Universität Berlin
Institut für Softwaretechnik und Theoretische Informatik
FG Softwaretechnik
Ernst-Reuter-Platz 7
10587 Berlin

Jähnichen
Mehlhase
Rein-Jury

MPGI 3

SLK B

Wintersemester 2011/2012
27. Februar 2012

Prüfen Sie zunächst, ob dieses Exemplar vollständig ist (**14** einseitig bedruckte Blätter inklusive Titelblatt).

Tragen Sie auf diesem Titelblatt und darüber hinaus auf allen Blättern, die Sie für Ihre Niederschrift verwenden, Ihren Namen und Ihre Matrikelnummer ein. Zum Bestehen der Prüfungsleistung „Klausur“ müssen Sie in den beiden schriftlichen Leistungskontrollen (SLKs) insgesamt mindestens **50 Punkte** erreichen. Darüber hinaus müssen Sie in jeder der beiden schriftlichen Leistungskontrollen mindestens je 10 Punkte erreichen. Insgesamt sind in der vorliegenden schriftlichen Leistungskontrolle 50 Punkte möglich. Die Bearbeitungszeit beträgt 75 Minuten. Bevor die Bearbeitungszeit anfängt, haben Sie 15 Minuten Einlesezeit. In der Klausur sind außer eines beschriebenen DIN-A4-Blattes **keine Hilfsmittel** zugelassen. Verwenden Sie ausschließlich das ausgeteilte Klausur-Papier. Es dürfen nur **permanent-schwarze oder -blaue Stifte** zum Lösen der Aufgaben verwendet werden.

Viel Erfolg!

Name, Vorname: _____

Matrikelnummer: _____

Studienrichtung: _____

MPGI3-Übungen habe ich im _____ besucht.
(z.B. WS 2011/12)

Aufgabe	1a	1b	2	Gesamt
Maximal:	17	12	21	50
Erreicht:				

Aufgabe 1 (Objekt-Z)

Mensch ärgere Dich nicht

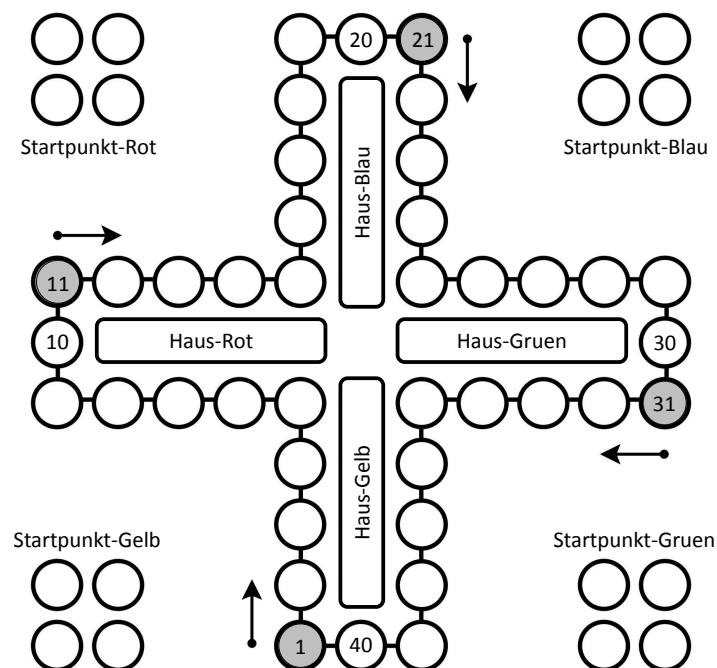
Aufgabenstellung

Eine vereinfachte Version eines Mensch ärgere Dich nicht Brettspiels ist mit Hilfe von Object-Z zu spezifizieren – dafür sind ausschließlich die vorgegebenen Klassen und deren Operationen zu erweitern. Beachten Sie, dass die Spielregeln der Einfachheit halber angepasst wurden. *Es sind exakt die hier angegebenen Regeln zu spezifizieren. Dabei dürfen den vorgegebenen Klassen keine neuen Attribute und keine neuen Operationen hinzugefügt werden!*

Sie müssen keine Sichtbarkeitslisten angeben!

Kurzbeschreibung

Das Spielbrett des Spiels besteht aus vierzig im Kreis angeordneten, durchnummerierten Spielfeldern, wie in der folgenden Grafik dargestellt.



An einem Spiel nehmen immer vier Spieler teil. Jeder der Spieler hat vier Figuren einer Farbe. Anfangs befinden sich die Figuren am *Startpunkt* (also noch nicht auf einem Spielfeld). Es wird reihum mit einem Würfel gewürfelt. Bei einer Sechs darf ein Spieler eine seiner Figuren auf sein *Einsetzfeld* (in der Grafik

grau dargestellt) setzen. Figuren, die sich auf einem Spielfeld befinden, bewegen sich im Uhrzeigersinn weiter – d.h. das Zielfeld eines Zuges errechnet sich aus der Summe des Würfelwurfes und der Nummer des Spielfeldes der zu setzenden Figur (wenn diese Zahl größer als 40 ist, wird 40 davon abgezogen). Auf jedem Spielfeld darf sich maximal eine Figur befinden. Landet eine Figur auf einem Spielfeld, auf dem sich eine Figur einer anderen Farbe befindet, wird diese rausgeworfen (d.h. sie kommt zum Startpunkt zurück). Eine Figur darf nicht auf ein Spielfeld gezogen werden, auf dem bereits eine Figur derselben Farbe steht. Hat eine Figur das komplette Spielbrett einmal umrundet, kommt sie in das *Haus* des entsprechenden Spielers. Das ist der Fall, wenn die Figur in einem Zug das *letzte Feld* passiert (dabei spielt sein Würfelergebnis keine Rolle). Der Spieler, der als Erstes mit allen vier Figuren in seinem Haus ankommt, gewinnt das Spiel.

Vorgaben

Die Farben der Figuren werden in dem folgenden Aufzählungstyp spezifiziert:

Farbe ::= Gelb | Rot | Blau | Gruen

Jedes Spielfeld wird durch eine Nummer (\mathbb{N}) beschrieben. Spielfelder werden im Folgenden auch als „Felder“ bezeichnet.

a) Figur

17 Punkte

Die Klasse *Figur* beschreibt eine Figur des Mensch ärgere Dich nicht Spiels. Eine Figur hat eine feste *farbe* und nimmt an einem *spiel* teil. Die Konstante *einsetzfeld* der Figur enthält das Feld, auf der die Figur vom Startpunkt eingesetzt wird. Im Gegensatz dazu enthält die Konstante *letztesFeld* das letzte Feld vor dem Eintritt in das Haus. Die Belegungen dieser beiden Konstanten sind abhängig von der *farbe* der Figur, wie aus der Grafik oben hervorgeht. Die folgende Tabelle verdeutlicht den zu spezifizierenden Zusammenhang:

<i>farbe</i>	<i>einsetzfeld</i>	<i>letztesFeld</i>
Gelb	1	40
Rot	11	10
Blau	21	20
Gruen	31	30

Eine Figur steht auf einem *feld* oder befindet sich *amStart* oder *imHaus* (boolesche Werte). Die Belegungen der drei sekundären Variablen (*feld*, *amStart* und *imHaus*) leiten sich aus den Belegungen der entsprechenden primären Variablen des *spiels* (s. Klasse *Spiel*, Aufgabe b) ab: Eine Figur befindet sich genau

dann *amStart*, wenn sie sich im *spiel* am *startpunkt* befindet. Analog dazu ist eine Figur genau dann *imHaus*, wenn sie sich im *spiel* im *haus* befindet. Ist eine Figur weder *amStart* noch *imHaus*, entspricht *feld* genau dem Feld, auf dem sie sich im *spiel* befindet (s. Attribut *positionen* der Klasse *Spiel*, Aufgabe b).

Die Operation *kann_einsetzen* beschreibt den Fall, dass eine Figur, die sich *amStart* befindet, durch den *wurf?* einer Sechse eingesetzt werden darf. Sie bekommt als Eingabe das Ergebnis eines Würfelwurfs (*wurf?*) und liefert das *einsetzfeld*, auf das die Figur gestellt werden muss, zurück (*ziel!*).

Eine Figur, die sich auf einem Feld (also weder *amStart* noch *imHaus*) befindet, *kann_ins_haus*, wenn sie, nachdem sie das Spielbrett einmal umrundet hat, mit dem *wurf?* ihr *letztesFeld* passiert – das bedeutet, sie stand vor dem Zug auf oder vor ihrem letzten Feld und überschreitet es mit dem *wurf?*. Die Rückgabe der Operation ist die *figur!* selbst. Als Eingaben für den *wurf?* sind nur die Zahlen eins bis sechs zugelassen.

Die Operation *kann_ziehen* beschreibt, dass die Figur einen Zug durchführen kann. Sie liefert zu einem *wurf?*, entsprechend der Position der Figur, das Zielfeld (*ziel!*) zurück. Als Eingabe für den *wurf?* sind abermals nur die Zahlen von eins bis sechs zugelassen. Weitere Vorbedingungen sind, dass sich (1) die Figur auf einem Feld (also weder *amStart* noch *imHaus*) befindet und (2) die Figur mit dem *wurf?* nicht ihr *letztesFeld* passiert (im Gegensatz zu der letzten Operation) – das bedeutet, sie bleibt entweder davor oder darauf stehen. Das Zielfeld wird durch die Summe von dem aktuellen *feld* und dem *wurf?* berechnet. Beachten Sie, dass es sich um im Kreis angeordnete Felder handelt: Überschreitet eine Figur das vierzigste Feld, muss sie beim ersten Feld weitergehen – dies ist bei der Summenbildung zu beachten.

Die Operation *feld_frei* liefert ein freies Feld (*ziel!*) und die *figur!* selbst zurück. Ein Feld wird als frei bezeichnet, wenn sich keine Figur darauf befindet.

Im Gegensatz dazu steht die Operation *fremde_figur_auf_feld*, die ein Feld (*ziel!*) zurückliefert, auf dem sich eine Figur einer anderen *farbe* befindet. Außerdem wird auch hier die *figur!* selbst (also die Figur, die den Zug durchführt) zurückgeliefert.

Die kombinierte Operation *kann_normal_ziehen* beschreibt, dass die Figur entweder eingesetzt werden kann oder einen Zug von einem Feld auf ein anderes durchführen kann. Vorbedingung der Operation ist, dass das *feld_frei* ist, auf das die Figur gesetzt werden soll.

Die kombinierte Operation *kann_rauswerfen* beschreibt ebenfalls, dass die Figur entweder eingesetzt werden kann oder einen Zug von einem Feld auf ein anderes durchführen kann. Hier gilt jedoch die Vorbedingung, dass das Zielfeld von einer fremden Figur belegt ist, die rausgeworfen werden kann.

Figur

farbe : Farbe
spiel : Spiel
einsetzfeld : \mathbb{N}
letztesFeld : \mathbb{N}

Δ

feld : \mathbb{N}
amStart :
imHaus :

kann_einsetzen

wurf? : \mathbb{N}
ziel! : \mathbb{N}

— *kann_ins_haus* —
wurf? : \mathbb{N}
figur! : *Figur*

— *kann_ziehen* —
wurf? : \mathbb{N}
ziel! : \mathbb{N}

— *feld_frei* —
ziel! : \mathbb{N}
figur! : *Figur*

fremde_figur_auf_feld _____

kann_normal_ziehen $\hat{=}$

kann_rauswerfen $\hat{=}$

b) Spiel

12 Punkte

Das Spiel besteht aus 40 durchnummerierten *spielfeldern* (von 1 bis 40).

Die Menge *startpunkt* enthält alle Figuren (aller Farben), die sich am Startpunkt befinden und auf ihren Einsatz warten. Die Menge *haus* enthält alle Figuren (aller Farben), die sich bereits im Haus befinden. Die partielle Abbildung *positionen* bildet alle *spielfelder*, auf denen sich eine Figur befindet, auf die entsprechende Figur ab. Das Attribut *dran* enthält die Farbe des Spielers, der am Zug ist. Spezifizieren Sie, dass sich von jeder der vier Farben exakt vier Figuren im Spiel (d.h. am *startpunkt*, im *haus* oder auf einem Spielfeld) befinden und dass sich eine Figur nicht gleichzeitig an mehreren Stellen befinden kann – d.h. sie kann nur entweder am *startpunkt*, im *haus* oder auf einem Spielfeld sein.

Initial befinden sich alle Figuren (aller Farben) des Spiels am *startpunkt*.

In der Operation *figur_setzen* wird eine *figur?* von ihrem aktuellen *feld* auf das Zielfeld (*ziel?*) gesetzt. Es sind keine Vorbedingungen nötig, da diese Operation nur innerhalb einer kombinierten Operation benutzt wird. Der Wert des Attributs *dran* ändert sich innerhalb dieser Operation nicht.

In der Operation *figur_setzen_mit_rauswerfen* wird, analog zur vorherigen Operation, eine *figur?* von ihrem aktuellen *feld* auf das Zielfeld (*ziel?*) gesetzt. Zusätzlich wird die sich auf dem Zielfeld (*ziel?*) befindende Figur rausgeworfen – d.h. vom Spielfeld entfernt und auf den *startpunkt* gesetzt. Es sind auch hier keinerlei Vorbedingungen nötig und der Wert des Attributs *dran* ändert sich innerhalb dieser Operation ebenfalls nicht.

Die Operation *haus_betreten* drückt aus, dass eine *figur?* das *haus* betritt (sie wird von ihrem aktuellen Spielfeld entfernt). Auch hier ändert sich der Wert des Attributs *dran* nicht.

Die kombinierte Operation *zug* drückt einen vollständigen Zug einer als Parameter übergebenen Figur aus. Vorbedingung ist, dass der Spieler mit der *farbe* der als Parameter übergebenen Figur *dran* ist. Kann die Figur normal ziehen (also entsprechend des Wurfes auf ein freies Feld gesetzt werden), so wird sie gesetzt. Kann die Figur eine andere rauswerfen, so wirft sie die andere Figur raus und wird dann ebenfalls gesetzt. Kann die Figur ihr Haus betreten, so betritt sie das Haus und wird vom Spielfeld entfernt.

Spiel

spielfelder : $\mathbb{P} \mathbb{N}$

startpunkt : $\mathbb{P} \text{Figur}$

haus : $\mathbb{P} \text{Figur}$

positionen : *spielfelder* \leftrightarrow *Figur*

dran :

INIT

figur_setzen _____
 $\Delta($ _____)
ziel? : \mathbb{N}
figur? : *Figur*

figur_setzen_mit_rauswerfen _____
 $\Delta($ _____)
ziel? : \mathbb{N}
figur? : *Figur*

haus_betreten _____
 $\Delta($ _____)
figur? : *Figur*

zug $\hat{=}$

Aufgabe 2 (Statecharts)

Waschmaschine mit integriertem Trockner

21 Punkte

Aufgabenstellung

Modellieren Sie die im Folgenden beschriebene Waschmaschine mit integriertem Trockner. Dafür sind die parallelen Komponenten *Steuerung*, *Tür*, *Verriegelung* und *Programm* mittels Statecharts zu spezifizieren. Es sollen nur exakt die beschriebenen Eigenschaften modelliert werden.

Benutzen Sie für Ihre Modellierung die in der folgenden Tabelle beschriebenen Ereignisse. Wenn Sie interne Ereignisse benutzen, müssen diese in der Tabelle ergänzt werden! Sie dürfen keine Variablen benutzen!

Die zu modellierende Waschmaschinenkombi besitzt die Tasten *On/Off* und *Programmauswahl*. Die Tür der Maschine ist mit einer elektronischen Verriegelung ausgestattet und kann im unverriegelten Zustand mechanisch geöffnet und geschlossen werden.

NAME	BESCHREIBUNG
EXTERNE EINGABEREIGNISSE	
<i>oo</i>	die Taste <i>On/Off</i> wird betätigt
<i>w</i>	die Taste <i>Programmauswahl</i> wird betätigt
<i>to</i>	die Tür wird geöffnet
<i>ts</i>	die Tür wird geschlossen
INTERNE EREIGNISSE	

Beschreibung

Die zu modellierende Waschmaschinenkombi verfügt über drei *Programme*: „Nur Waschen“, „Nur Trocknen“ und „Waschen und Trocknen“. Anfangs ist das letztgenannte Programm ausgewählt. Durch Drücken der Taste *Programmwahl* kann, wenn sich die Waschmaschine im ausgeschalteten Zustand befindet, zwischen den drei Programmen (in beliebiger Reihenfolge) hin und her gewechselt werden.

Die *Tür* der Maschine kann entweder geöffnet oder geschlossen sein. Sie kann nur geöffnet werden, wenn die *Verriegelung* inaktiv ist. Anfangs ist die Tür geschlossen und die Verriegelung inaktiv.

Die Maschine befindet sich anfangs im ausgeschalteten Zustand. Über einen Druck auf *On/Off* kann sie eingeschaltet werden. Ist die Tür geschlossen, beginnt sie, je nach ausgewähltem Programm, mit dem Waschen oder dem Trocknen – natürlich wird bei dem Programm „Waschen und Trocknen“ zuerst gewaschen und danach getrocknet. Ist die Tür nicht geschlossen, wird die Maschine durch einen Druck auf *On/Off* zwar eingeschaltet, wartet aber mit dem Beginnen des ausgewählten Programmes bis die Tür geschlossen wird – dann beginnt sie automatisch.

Wird eine Wäsche gestartet, wird vor Beginn zunächst die Tür verriegelt. Nachdem die einstündige Wäsche abgeschlossen ist, wird die Tür wieder entriegelt. Ist das Programm „Waschen und Trocknen“ ausgewählt, wird danach automatisch das 30-minütige Trocknen gestartet. Andernfalls geht die Maschine in einen Ruhezustand über (bleibt aber eingeschaltet).

Während des Trocknens ist die Tür der Waschmaschine unverriegelt und kann jederzeit geöffnet werden. In diesem Fall wird das Trocknen unterbrochen. Nachdem die Tür wieder geschlossen wurde, beginnt das 30-minütige Trocknen von vorne. Nach Abschluss des Trocknens geht die Waschmaschine in den Ruhezustand über.

Wird der Knopf *On/Off* im eingeschalteten Zustand betätigt, hat dies, abhängig vom aktuellen Zustand der Maschine, verschiedene Auswirkungen. Ist gerade ein Waschvorgang aktiv, wird dieser abgebrochen und die Maschine wartet drei Minuten bevor sich die Tür entriegelt und die Maschine abschaltet – innerhalb dieser drei Minuten reagiert die Maschine auf keinerlei Ereignisse. Ist gerade kein Waschvorgang aktiv, schaltet sich die Maschine sofort aus. Nach erneutem Einschalten wird das ausgewählte Programm von vorne gestartet.

