



21. Bundeswettbewerb Informatik 2002/2003

Die Aufgaben der zweiten Runde

Allgemeine Hinweise

An dieser Runde können nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben mindestens 12 Punkte erreicht haben oder einer Gruppe angehört haben, der dieses gelungen ist. Gruppenarbeit ist nicht zulässig. **Einsendeschluss ist der 28. April 2003**, wobei das Datum des Poststempels gilt. Bitte adressieren Sie Ihre Einsendungen wieder an den

Bundeswettbewerb Informatik, Ahrstraße 45, 53175 Bonn

Es gibt drei Aufgaben, die alle in die Bewertung mit einbezogen werden. Wir erwarten zunächst eine einfache, nachvollziehbare und vollständige Lösung aller Teilaufgaben inklusive der geforderten Beispiele (teilweise in der Aufgabenstellung vorgegeben). Pluspunkte für eine höhere Bewertung können Sie darüber hinaus erreichen, wenn Sie die Aufgaben an Stellen, an denen dies möglich und sinnvoll ist, eigenständig weiterentwickeln. Dabei bevorzugen wir inhaltliche Erweiterungen und Verbesserungen, etwa von Datenstrukturen und Algorithmen, und legen keinen Wert auf aufwendige Tricks, z.B. zur reinen Verschönerung der Mensch-Computer-Interaktion oder der grafischen Ausgabe. Begründen Sie für jede Erweiterung, warum sie sinnvoll ist.

Denken Sie bitte daran, dass zur Bewertung möglicherweise nur Ihre Papier-Unterlagen herangezogen werden können. Diese sollten also einen lückenlosen und nachvollziehbaren Nachweis des Leistungsumfangs und der Funktionstüchtigkeit Ihrer Programme geben. Dem Umfang Ihrer Einsendung setzen Sie bitte Grenzen, denn eine gute Dokumentation vermittelt kurz aber präzise alles Nötige, insbesondere Ihre wesentlichen Lösungsideen. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die nur die Aufgabenstellung kennen, wissen müssen, um Ihre Lösungsidee zu verstehen und Ihre Realisierung dieser Idee nachzuvollziehen. Generell sind gute und originelle Ideen zwar entscheidend, aber die Dokumentation hat schon oft den Ausschlag für oder gegen das Weiterkommen gegeben.

Grundsätzlich gelten die Gliederungs- und Dokumentationsrichtlinien der 1. Runde weiter. Zu jeder Teilaufgabe gehört also die Lösungsidee und die Dokumentation der Lösung sowie des



dazugehörigen Programms (eine Beschreibung, wie die Idee z.B. in eine Datenstruktur umgesetzt oder im Programm realisiert wurde, Hinweise auf Nutzungsgrenzen, Besonderheiten usw.). Dabei sind (halb-)formale Notationen besser als Programmausschnitte. Für die geforderten Programme erwarten wir zusätzlich Programmablaufprotokolle (kommentierte Probeläufe des Programms); senden Sie uns bitte Ergebnisse von Programmläufen mit unterschiedlichen Daten. Komplettiert wird das Papier-Material durch den Programmtext, wobei unwichtige und automatisch generierte Teile nicht ausgedruckt werden sollen.

Schicken Sie uns alles in lesbarer Form auf Papier. Bitte schicken Sie uns Ihre Unterlagen auf einzelnen, gelochten Blättern im Format DIN A 4 (Hüllen mit Lochrand nur bei ausreichender Stabilität verwenden; keine Heftstreifen oder Mappen) und geben Sie auf jedem Blatt Verwaltungsnummer, Vorname, Name und Seitennummer an. Sie finden die Verwaltungsnummer auf Ihrer Teilnahmebescheinigung der ersten Runde. Bitte gliedern Sie Ihre vollständige Einsendung in (a) Allgemeines, (b) Unterlagen zur Aufgabe 1, (c) Unterlagen zur Aufgabe 2 und (d) Unterlagen zur Aufgabe 3.

Außerdem senden Sie uns bitte die Programmtexte und lauffähigen Programme auf einer 3,5“-Diskette oder CD (lesbar unter DOS/Windows98; die Programme dürfen natürlich auch für andere Betriebssysteme entwickelt worden sein).

Für Fragen zu den Aufgaben gibt es wieder eine Hotline, und zwar unter 0228 - 37 86 46 jeweils von 17-19h am 7.1., 26.2., 31.3. und 23.4.2003 oder unter bwinf@bwinf.de. Die Antwort auf E-mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf unseren Webseiten (www.bwinf.de). In der newsgroup fido.ger.bwinf wird sicher wieder über die Aufgaben diskutiert werden – ohne dabei Lösungsideen auszutauschen.

Allen Teilnehmern der zweiten Runde wird Anfang Juni die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die vom 29.9.-2.10.2003 im Heinz Nixdorf Museumsforum in Paderborn ausgerichtet werden wird. Dort werden die Bundessieger und Preisträger ermittelt und am letzten Tag ausgezeichnet. Bundessiegern winkt die Aufnahme in die Studienstiftung des deutschen Volkes. Allgemein werden Geld- und Sachpreise vergeben, unter anderem ein Aufenthalt an einer Sommerschule im Ausland. Der Rechtsweg ist ausgeschlossen.

Auf einige Dinge möchten wir Sie noch hinweisen, die Sie vielleicht zusätzlich motivieren können, sich an der zweiten Runde aktiv zu beteiligen. In vielen Bundesländern gibt es Anerkennungen für gute Teilnehmer an der zweiten BWINF-Runde. Außerdem ist prinzipiell möglich, eine Einreichung zur zweiten Runde des Bundeswettbewerbs Informatik als so genannte besondere Lernleistung in die Abiturwertung einfließen zu lassen. Genaueres erfahren Sie an Ihrer Schule. Jüngere Zweitrundenteilnehmer haben die Chance, zu einer Schülerakademie eingeladen zu werden. Der Hauptpreis für eine herausragende Einsendung dieser zweiten Runde ist aber

ein iBook von Apple.

Viel Spaß und viel Erfolg!



MCI-Sonderpreis

Beim Bundeswettbewerb Informatik spielt die Bedienschnittstelle der eingesandten Programme bei der Bewertung prinzipiell keine Rolle. Für die Benutzbarkeit von Informatiksystemen ist diese Komponente aber von ganz entscheidender Bedeutung, und so wird auch bei den Einsendungen zum BWINF teilweise erhebliche Mühe auf den Interaktionsaspekt verwendet.

Diese Mühe soll belohnt werden: Zum dritten Mal schreibt der Bundeswettbewerb Informatik gemeinsam mit dem Fachbereich Mensch-Computer-Interaktion der Gesellschaft für Informatik (GI) einen Sonderpreis aus für Programme mit besonders gelungenen Bedienschnittstellen. Zu gewinnen gibt es ein Paket der Übersetzungssoftware Personal Translator, das von Linguattec gestiftet wurde. Verliehen wird dieser Preis voraussichtlich auf der Tagung „Mensch & Computer“, die vom 7.-10.9.2003 in Stuttgart stattfinden wird.

Dafür wird Folgendes erwartet: Geben Sie an, mit welchem der zu den Wettbewerbsaufgaben eingesandten Programme Sie sich um den MCI-Sonderpreis bewerben wollen. Beschreiben Sie dann in einem separaten Dokument die Bedienschnittstelle des Programms. Geben Sie aber nicht nur eine Bedienungsanleitung, sondern erläutern Sie Entwurfskonzept und -entscheidungen.

Das Programm und seine Interaktionskomponente werden an Hand einer Reihe von Kriterien bewertet, die unten näher erklärt sind. Sie richten sich mehrheitlich nach einigen Dialogprinzipien und wichtigen Eigenschaften der Informationspräsentation, die in Normen für Mensch-Computer-Schnittstellen gefordert werden. Dazu kommen zwei Aspekte, die für den BWINF wichtig sind. Bei den angewandten Kriterien handelt es sich also um

Dialogkriterien: Der Dialog eines Benutzers mit einem System ist

selbsterklärend, wenn jeder Dialogschritt durch Reaktionen des Systems sofort verständlich wird oder dem Benutzer auf Anforderung erklärt wird;

kontrollierbar, wenn Benutzer Richtung und Geschwindigkeit der Interaktion bestimmen können, bis ihr Ziel erreicht ist;

fehlertolerant, wenn trotz offensichtlicher Fehler in der Eingabe das beabsichtigte Ergebnis ohne bzw. mit nur geringem Eingreifen der Benutzer erzielt werden kann.

Präsentationskriterien: Die Präsentation von Information ist

klar und präzise, wenn der Informationsgehalt schnell und genau und ohne überflüssige Informationen vermittelt wird;

übersichtlich, wenn die Menge der Informationen knapp und geordnet dargestellt wird;

lesbar, wenn die (visuelle) Darstellung der Information leicht zu lesen bzw. erkennen ist.

BWINF-Kriterien: Ein Programm und seine Bedienschnittstelle ist

originell, wenn die Programmfunktionalität auf überraschende Weise nutzbar gemacht wird;

inspizierbar, wenn die Bedienschnittstelle Zugang zur Funktionalität erlaubt und die Bewertung des Programms erleichtert.



Aufgabe 1: Restgeld

In Fantasien gibt es Münzen und Geldscheine im Wert von 1 Fant, 2 Fanten, 3 Fanten usw. bis 100 Fanten. In den Bahnhöfen Fantasien stehen Automaten, an denen Fahrkarten zum Preis von 1 Fant, 2 Fanten, 3 Fanten usw. bis 100 Fanten erworben werden können. Wenn der Kunde z. B. eine Fahrkarte zum Preis von 21 Fanten kauft und mit einem 86-Fanten-Schein bezahlt, gibt der Automat z. B. einen 47-Fanten-Schein und zwei 9-Fanten-Münzen als Restgeld heraus, denn natürlich kann er nicht immer alle hundert verschiedenen Münzen und Scheine in ausreichender Menge vorrätig halten.

Aufgabe

1. Schlage eine Modellierung der Automaten als Objekte einer Klasse vor. Welche Attribute soll ein Automatenobjekt besitzen, und welche Methoden soll es zur Verfügung stellen? Eingeworfenes und herauszugebendes Geld befindet sich in einem Speicher, dessen Fassungsvermögen wir der Einfachheit halber als genügend groß annehmen.
2. Beschreibe eine Methode zur Berechnung der bei einem Fahrkartenkauf herauszugebenden Münzen und Scheine. Beschreibe zuerst die genauen Eingabe- und Ausgabevereinbarungen deiner Methode und danach den Algorithmus, der diese Methode realisiert.
3. Wenn die Möglichkeit besteht, dass der Automat beim nächsten Fahrkartenkauf kein passendes Restgeld herausgeben kann, soll eine Warnanzeige aufleuchten. Entwirf eine Methode, mit der die Warnanzeige gesteuert werden kann. Beschreibe wieder die Eingabe- und Ausgabevereinbarungen deiner Methode und den benutzten Algorithmus.
4. Sehr häufig hat der Automat mehrere Möglichkeiten für die Geldrückgabe. Obwohl sie alle im Prinzip korrekt sind, sind einige Möglichkeiten in der Praxis und auf Dauer besser als andere. Beschreibe und diskutiere verschiedene Kriterien einer gut gewählten Geldrückgabe und verschiedene Strategien, die der Automat benutzen könnte, um die Kriterien möglichst zu erfüllen. Zum Beispiel ist es wahrscheinlich keine gute Strategie, immer nur 1-Fant-Münzen herauszugeben, sofern genug vorhanden sind: Die Kunden werden nicht erfreut sein, und der Automat hat bald fast keine 1-Fant-Münzen mehr.



Aufgabe 2: Brandubh

Im 6. und 7. Jahrhundert war in Skandinavien und Irland unter verschiedenen Namen ein Brettspiel sehr populär. In Irland war es u.a. unter der Bezeichnung brandubh (gesprochen branduhw), d.h. schwarzer Rabe, bekannt. Es wird auf einem Brett von 7x7 Feldern gespielt.

Ein König und 8 Getreue bilden die weiße, 16 Belagerer bilden die schwarze Partei. Die Aufgabe der weißen Partei ist es, dem König bei der Flucht von seinem Thron genau in der Mitte des Spielfeldes auf eines der Randfelder zu helfen. Die schwarze Partei versucht dieses zu verhindern.

Die Startaufstellung ist: (**X**: weißer Getreuer, **K**: König, **O**: Belagerer)

	O	O		O	O	
O			X			O
O			X			O
	X	X	K	X	X	
O			X			O
O			X			O
	O	O		O	O	

Die Regeln für das Spiel sind:

- Auf einem Feld kann sich nur eine Figur befinden.
- Gezogen werden darf in senkrechter und in waagerechter Richtung. Es darf beim Ziehen keine Figur übersprungen werden.
- Der Thron darf nur vom König besetzt werden. Wenn er nicht besetzt ist, darf er übersprungen werden.
- Wenn eine Figur durch einen gegnerischen Zug zwischen zwei Gegner gerät, wird sie geschlagen, also aus dem Feld genommen. Wird eine Figur zwischen zwei gegnerische Figuren gezogen, passiert nichts. Der König kann nicht auf diese Weise geschlagen werden.
- Schwarz gewinnt, wenn der König gefangen ist. Dies ist der Fall, wenn:
 - auf den vier Nachbarfeldern gegnerische Figuren stehen oder
 - auf drei Nachbarfeldern gegnerische Figuren stehen und das vierte Feld das Thronfeld ist.
- Weiß gewinnt, wenn der König ein Randfeld erreicht hat.
- Wird dreimal die gleiche Stellung wiederholt bzw. ist kein Zug mehr möglich, endet das Spiel unentschieden.



Aufgabe

Entwirf und realisiere eine Spielumgebung, die interaktiv mindestens das Folgende leistet:

- Zwei menschliche Spieler können gegeneinander spielen.
- Spielzüge werden entgegengenommen, überprüft und bei Fehlern zurückgewiesen.
- Das Schlagen von Figuren wird übernommen. Spielende, Unentschieden und Sieg werden erkannt.

Die Spielumgebung soll auch ein Hilfesystem enthalten. Dies soll mindestens

- einführende Informationen zum Spiel zur Verfügung stellen,
- auf fehlerhaft eingegebene Spielzüge geeignet reagieren (z.B. durch Anzeige relevanter Regeln) und
- auf Anfrage die regelgemäßen Möglichkeiten für den nächsten Zug aufzeigen. Hierbei ist explizit nicht verlangt, dass das Hilfesystem unter den Möglichkeiten mit Hilfe einer eigenen Spielstrategie auswählen soll.

Es wird insbesondere auf die Qualität des Hilfesystems Wert gelegt, nicht auf eine ausgefeilte grafische Darstellung.



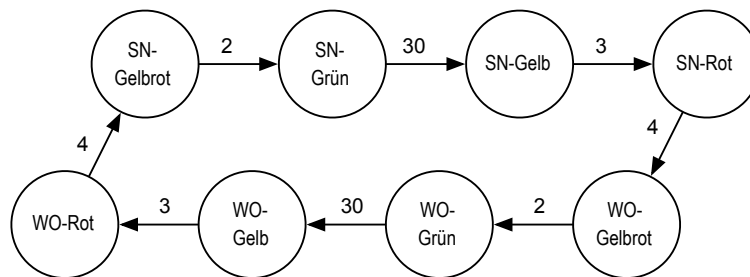
Aufgabe 3: Ampelsteuerung

An einer Kreuzung zweier Straßen, die westöstlich (WO) und süd-nördlich (SN) verlaufen, gibt es in jeder Richtungsachse Ampeln für Fußgänger (F) und Ampeln für Autos (A). Die Ampelleuchten können mit den Steuerbefehlen EIN und AUS geschaltet werden.

Die Ampeln der beiden Richtungsachsen sind je Verkehrsteilnehmertyp stets gleichgeschaltet. Das heißt, dass zum Beispiel der Steuerbefehl (EIN WO F GRÜN) das Aufleuchten der vier „grünen Menschlein“ für jene Fußgänger bewirkt, welche die Kreuzung von Ost nach West oder von West nach Ost überqueren wollen.

Die Ampelanlage dieser Kreuzung kann für den Zweck ihrer programmierten Steuerung als ein System von *Zuständen* modelliert werden, wobei die durch ein *Ereignis* ausgelöste Ausführung von Steuerbefehlen ihrerseits einen *Zustandsübergang* verursacht. Ein solches Modell bezeichnen Informatiker gerne als *Zustandsautomat*.

Im Normalbetrieb soll das Modell unserer Ampelanlage acht Zustände haben, deren Namen sich daran orientieren, welche Ampelleuchten für die Autos einer Richtungsachse eingeschaltet sind. SN-ROT ist also zum Beispiel der Zustand, in dem die zwei Auto-Ampeln der SN-Richtungsachse auf Rot stehen. Die Übergänge zwischen den Zuständen beim Normalbetrieb sind mit der folgenden Grafik modelliert:



Die Werte an den Übergangspfeilen geben an, nach wieviel Sekunden der vorherige Zustand in den nachfolgenden Zustand gesteuert werden soll. Dazu verfügt die Ampelsteuerung über eine Countdown-Uhr, die beim Eintritt eines Zustands auf den entsprechenden Sekundenwert gestellt wird. Die Ampelsteuerung wartet dann auf das Ereignis, dass die Countdown-Uhr den Wert Null erreicht. Tritt dieses Ereignis ein, steuert die Ampelsteuerung die Ampelanlage in den nachfolgenden Zustand.

Die erste Grafik modelliert nur den Normalbetrieb der Ampelanlage. Nun wollen wir eine Steuersprache *S* vorstellen, mit der diese und andere Betriebsarten programmiert werden können. *S* kennt sechs Steuerbefehlstypen:

Typ 1: (#ZUSTANDSDEKLARATION <zustandsname>)

Ein isoliertes Zustandsobjekt mit Namen <zustandsname> wird generiert.

Typ 2: (#EREIGNISDEKLARATION <ereignisname>)



Ein mögliches Ereignis wird definiert. Beispiel:
(#EREIGNISDEKLARATION SN-KNOPFDRUCK)

Typ 3: (#EREIGNIS <ereignisname>)
Ein Ereignis tritt ein. Beispiel: (#EREIGNIS SN-KNOPFDRUCK)

Typ 4: (#COUNTDOWN <sekundenwert>)
Setzt die Countdown-Uhr auf den Wert <sekundenwert>. Der Sekundenwert muss mindestens 1 und darf höchstens 60 betragen. Beispiel: (#COUNTDOWN 19)

Typ 5: (#EIN|#AUS #WO|#SN #A #ROT|#GELB|#GRÜN)
(#EIN|#AUS #WO|#SN #F #ROT|#GRÜN)
Dies sind die Steuerbefehle für die Ampelleuchten. Das Symbol | trennt die jeweils möglichen Steueralternativen.

Typ 6: (#ÜBERGANGSDEKLARATION <zustandsname-vorher>
<ereignis> <steuerbefehl>* <zustandsname-nachher>)
Ein Zustandsübergang wird definiert, der dann eintritt, wenn sich die Ampelanlage im Zustand <zustandsname-vorher> befindet und das Ereignis <ereignis> eintritt. Dann werden kein Steuerbefehl, ein Steuerbefehl oder mehrere Steuerbefehle hintereinander ausgeführt (dies meint das hintergestellte Symbol Asterisk *). Dann geht die Ampelanlage in den Zustand <zustandsname-nachher>. Beispiel:

```
(#ÜBERGANGSDEKLARATION SN-GRÜN COUNTDOWN-IST-NULL
      (#AUS #SN #F #GRÜN) (#EIN #SN #F #ROT)
      (#AUS #SN #A #GRÜN) (#EIN #SN #A #GELB)
      (#COUNTDOWN 3) SN-GELB)
```

Aufgabe

1. Programmiere den Normalbetrieb der Ampelanlage als Programm-1 in der Sprache S.
2. Um die Anlage flexibler zu machen, haben die Ingenieure den Fußgängerampeln Druckknöpfe und den Fahrbahnen der Autos Induktionsschleifen vor den Ampeln hinzugefügt. Damit soll die Ampelsteuerung dem Verkehrsaufkommen angepasst werden können. Erweitere das Programm-1 entsprechend zu Programm-2. Dabei sollen die Meldungen der Sensoren (Druckknöpfe und Induktionsschleifen) als Ereignisse modelliert werden. Die Verkehrssicherheit muss aber stets gewährleistet sein und kein Verkehrsteilnehmer soll beliebig lang an einer Ampel warten müssen. Beschreibe uns genau, wie das Programm dies sicherstellt.
3. Erweitere das Programm-2 so, dass die Ampelsteuerung auch auf andere Betriebsarten wie zum Beispiel „Straßenbauarbeiten“, „Verkehrsunfall“ und „Marathonveranstaltung“ umgeschaltet werden kann. Die Polizeizentrale hat dazu eine Möglichkeit, selbst Ereignisse an die Ampelsteuerung zu schicken.



Die Testumgebung für S

Auf den Webseiten des BWINF (www.bwinf.de, Die Aufgaben, 2. Runde) steht ein Interpreter zur Verfügung (voraussichtlich ab Anfang Januar), mit dem Du die Programme testen kannst. Dort finden sich dann auch weitere Benutzungshinweise für den Interpreter und eine vollständige formale Beschreibung der Sprache S in erweiterter Backus-Naur-Form (EBNF).

Hinweise zum Interpreter

Der Interpreter visualisiert die Ampelanlage und nimmt ein Programm als ASCII-Textdatei entgegen. Nach seinem Programmstart befindet sich der Interpreter im vordeklarierten Zustand INTERPRETER-START. Vordeklariert ist außerdem das Ereignis COUNTDOWN-IST-NULL, das bei Ablauf der durch #COUNTDOWN gesetzten Uhr eintritt.

Der Interpreter arbeitet das Programm Befehl für Befehl ab und führt die Befehle sofort aus. Der Interpreter stoppt, wenn er einen Programmfehler erkennt. Der Interpreter erkennt zwei Fehlertypen:

- Fehlertyp 1 ist ein Syntaxfehler, also ein Verstoß gegen die Grammatik von S.
- Fehlertyp 2 ist der Versuch, einen `<identifizier>` in einer Zustandsdeklaration oder Ereignisdeklaration ein zweites Mal zu verwenden.