

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

Kennzahl: \_\_\_\_\_

**Herbst**

Kennwort: \_\_\_\_\_

**2001**

**46114**

Arbeitsplatz-Nr.: \_\_\_\_\_

---

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**  
**- Prüfungsaufgaben -**

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Algorithmen/Datenstrukt./Progr.-meth.**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 6

Bitte wenden!

## Thema Nr. 1

### Vorbemerkung:

Als Programmiersprache kann bei der Lösung der folgenden Aufgaben Pascal, Modula, C, C++ oder Java verwendet werden.

### Aufgabe 1

An einer Rundstrecke befinden sich 20 Zapfsäulen. Vorausgesetzt ist, dass die  $i$ -te Zapfsäule eine Benzinmenge  $a_i$  enthält und dass eine Benzinmenge  $b_i$  nötig ist, um die nächste Zapfsäule zu erreichen. Es soll ein Programm erstellt werden, das die Nummern aller Zapfsäulen ausgibt, von denen aus ein Auto mit anfangs leerem (aber genügend großem) Tank die Rundstrecke abfahren kann, bis der Ausgangspunkt wieder erreicht ist. Das Programm soll zunächst die Werte  $a_i$  und  $b_i$  in geeignete Felder einlesen.

### Aufgabe 2

In einer Wetterstatistik werden für 30 europäische Großstädte die täglichen Durchschnittstemperaturen des Jahres 2000 erfasst.

- a) Deklarieren Sie geeignete Datentypen oder Klassen (zunächst ohne Methoden) zur Darstellung der Wetterstatistik, der Großstädte und der Messungen, so dass
  - die Wetterstatistik über eine Liste aller Großstädte verfügt,
  - jede Großstadt einen Namen hat und über eine Liste ihrer Messungen im Jahr 2000 verfügt,
  - jede Messung einen Wert (die Durchschnittstemperatur) und einen Tag und einen Monat (zur Bestimmung des Messdatums) hat!
- b) Geben Sie eine Funktion (bzw. Prozedur) oder eine Methode an, die für eine Großstadt die Durchschnittstemperatur im Jahr 2000 berechnet! Falls die Implementierung in C++ oder Java angegeben wird, soll die Methode zur Klasse „Großstadt“ gehören, ansonsten soll die Prozedur einen formalen Parameter vom Typ „Großstadt“ verwenden.
- c) Geben Sie eine Funktion (bzw. Prozedur) oder eine Methode an, die den Namen der (bzw. einer) Großstadt mit maximaler Durchschnittstemperatur im Jahr 2000 berechnet! Falls die Implementierung in C++ oder Java angegeben wird, soll die Methode zur Klasse „Wetterstatistik“ gehören, ansonsten soll die Prozedur einen formalen Parameter vom Typ „Wetterstatistik“ verwenden.

Fortsetzung nächste Seite!

### Aufgabe 3

Im Folgenden soll die Datenstruktur der endlichen Teilmengen ganzer Zahlen implementiert werden.

- a) Deklarieren Sie einen Datentyp „SetInt“ oder eine Klasse „SetInt“ (zunächst ohne Methoden), so dass endliche Mengen ganzer Zahlen durch einfach verkettete Listen (Zeigerstrukturen) dargestellt werden!
- b) Es sollen die Operationen „isempty“, „contains“, „add“ und „remove“ implementiert werden. „isempty“ stellt fest, ob eine Menge  $s$  leer ist; „contains“ stellt fest, ob ein Element  $x$  zu einer Menge  $s$  gehört, „add“ fügt ein Element  $x$  zu einer Menge  $s$  hinzu (falls  $x$  noch nicht zu  $s$  gehört) und „remove“ entfernt ein Element  $x$  aus einer Menge  $s$ .

Falls Sie in Teil a) einen Datentyp „SetInt“ deklariert haben, sollen die Operationen durch geeignete Funktionen bzw. Prozeduren mit Parametern  $s$  und (ggf.)  $x$  implementiert werden.

Falls Sie in Teil a) eine Klasse „SetInt“ deklariert haben, sollen die Operationen durch geeignete Methoden dieser Klasse (ggf. mit Parameter  $x$ ) implementiert werden.

### Aufgabe 4

Gegeben ist die folgende rekursive Funktionsdefinition:

```
function f (x: integer) : integer;  
  if x = 1 then 0  
  else if x = 2 then 1  
    else f(x-2) - f(x-1)  
  endif  
endif
```

- a) Welche Werte liefert die Funktion für die Aufrufe  $f(3)$ ,  $f(4)$ ,  $f(5)$  und  $f(6)$ ? Wie viele rekursive Aufrufe erfolgen jeweils?
- b) Bestimmen Sie die Menge aller Integer-Zahlen  $x$ , für die der Aufruf  $f(x)$  terminiert!
- c) Geben Sie eine rekursive Funktionsdefinition mit Kopf

```
function anzahl (x: integer) : integer;
```

an, die für jede Integer-Zahl  $x$ , für die der Aufruf  $f(x)$  terminiert, die Anzahl der rekursiven Aufrufe von  $f(x)$  berechnet!

## Thema Nr. 2

## 1. Programmiermethodik

Wir betrachten die Implementierung einer endlichen Menge. Sie sei softwaremäßig als ein Modul gekapselt, der nach außen die folgenden Funktionen bereitstellt:

<b>empty_set:</b>		→	<i>set</i>	kreiert eine neue (leere) Menge
<b>add:</b>	<i>set x item</i>	→	<i>set</i>	fügt ein Element in die Menge ein
<b>remove:</b>	<i>set x item</i>	→	<i>set</i>	entfernt ein Element, sofern es in der Menge enthalten ist
<b>is_empty:</b>	<i>set</i>	→	<i>bool</i>	prüft, ob die Menge leer ist
<b>is_in</b>	<i>item x set</i>	→	<i>bool</i>	prüft, ob das Element in der Menge ist

a) Die Semantik der Funktion **is\_empty** kann folgendermaßen definiert werden:

```
is_empty(empty_set) = true
is_empty(add(s,e))  = false
```

Definieren Sie in der gleichen Weise die Semantik von **is\_in**!

- b) Wenn Sie die Semantik von **remove** definieren wollen, müssen Sie an die Möglichkeit denken, dass **add** ein Element mehrfach in die Menge aufnimmt. Wie lässt sich das Zusammenspiel von **remove** und **add** definieren? (Es gibt zwei grundlegend verschiedene Möglichkeiten.)
- c) Ergänzen Sie das Beispiel durch Syntax und Semantik einer Funktion **card**, die die Kardinalität der Menge bestimmt! (Beachten Sie Ihre Überlegungen zu b)!)
- d) Eine Funktion  $f : M_1 \rightarrow M_2$  kann als Menge implementiert werden, indem wir  $item = M_1 \times M_2$  setzen und folgende Funktion ergänzen: **val**:  $set \times M_1 \rightarrow M_2$ , die den Funktionswert  $f(x)$  herausucht. Definieren Sie die Semantik von **val**, und passen Sie die Semantik von **add** an die neue Aufgabenstellung an!

## 2. Systementwurf

Ein aktuelles Thema in der Diskussion um die Strukturierung von Softwaresystemen ist die Objektorientierung.

- a) Erläutern Sie an dem folgenden (in einer Phantasiesprache geschriebenen) Beispiel die Begriffe Klasse, Objekt und Methode:

```
class konto is
  eroeffne_konto (kunde);
  einzahlen (betrag);
  auszahlen (betrag);
  kontostand () returns betrag;
end konto;
```

(*betrag* und *kunde* seien geeignet definierte Klassen.) Erläutern Sie den Unterschied zwischen Instanzen- und Klassenvariablen, und geben Sie zu der Klasse *konto* je ein Beispiel an!

- b) Ein zentrales Konzept des objektorientierten Software-Entwurfs ist die Vererbung. Welche Möglichkeiten zur Konstruktion neuer Klassen bietet dieses Konzept?
- c) Einige objektorientierte Programmiersprachen (z. B. C++) erlauben die Mehrfachvererbung, während andere (z. B. Java) sich auf die Einfachvererbung beschränken. Erläutern Sie den Unterschied und diskutieren Sie Vor- und Nachteile beider Möglichkeiten! Welchen Ausweg bietet Java für den Fall, dass die Problemstruktur in natürlicher Weise zu einer Mehrfachvererbung führen würde?
- d) Erläutern Sie, was man in der objektorientierten Programmierung unter virtuellen Funktionen und „dynamischem Binden“ versteht! Welche Vorteile bietet dieses Konzept bei der Systementwicklung?

## 3. Algorithmen und Datenstrukturen

Streuspeicherverfahren (Hash-Verfahren) erlauben die Speicherung einer Menge von Daten, die über Schlüssel aus einer Schlüsselmenge  $K$  identifiziert werden können, wobei nur ein kleiner Teil der potentiell möglichen Schlüssel wirklich auftritt. Man reduziert daher die Größe der Schlüsselmenge mit Hilfe einer Funktion  $h : K \rightarrow A$  auf einen kleineren Adressbereich. Dabei kann es vorkommen, dass zwei oder mehr real auftretende Schlüssel auf die gleiche Adresse abgebildet werden (Kollision).

- a) Bei der „offenen“ Adressierung zur Kollisionsauflösung definiert jeder Schlüssel  $k$  eine Folge von Speicherplätzen  $h(k, i)$ . Beschreiben Sie verbal und durch ein Struktogramm das Einfügen eines neuen Objektes mit dem Schlüssel  $k$  in die Menge!



- b) Beschreiben Sie verbal und durch ein Struktogramm das Suchen nach einem Element mit gegebenem Schlüssel  $k$ ! Wie erkennt Ihr Algorithmus, dass kein Element mit dem Schlüssel  $k$  in der Menge gespeichert ist?

- c) Bei der häufig verwendeten, linearen Kollisionsauflösung

$$h(k,i) = [h_0(k) + ci] \bmod m,$$

wobei  $m$  die Größe des Adressbereichs  $A$  und  $c$  eine Konstante mit  $\text{ggT}(m, c) = 1$  ist, kann es zur „Clusterbildung“ kommen. Was versteht man darunter, wie lässt sich der Effekt formelmäßig charakterisieren, und welche Auswirkung hat er?

- d) Die Anwendung der Streuspeicherung mit offener Adressierung ist nicht unbedingt zu empfehlen, wenn es erlaubt sein soll, dass Daten aus der Menge gestrichen werden können. Beschreiben Sie das auftretende Problem und eine denkbare Lösung!