

## Prüfung Grundlagen Informatik 2

Aufgabensteller: Prof. Dr. B.Glavina, Prof. Dr. J.Schweiger  
Prüfungsdauer: 90 Minuten  
Hilfsmittel: vier Seiten Skript-Zusammenfassung

### Aufgabe 1 (Listen; etwa 25%)

Gegeben ist die C-Deklaration des Verbundtyps `student` wie folgt:

```
struct student {  
    char name[40];  
    unsigned int matrikelnr;  
    int note;  
    struct student *next;  
};
```

a) Schreiben Sie eine C-Funktion `select_success`, die eine auf obigem Verbundtyp basierende Liste als Parameter übergeben bekommt und daraus eine neue Liste konstruiert, welche als Funktionsergebnis geliefert wird. Die Ergebnisliste soll nur die Studenten enthalten, die bestanden haben (deren Note nicht 5 ist). Achten Sie bei Ihrer Implementierung darauf, dass die Ursprungsliste nicht verändert wird.

b) Falls die obige Funktion als Prozedur realisiert werden soll, braucht man einen zusätzlichen (zweiten) Parameter (`result`). Geben Sie die entsprechende Kopfzeile der Prozedur `select_success_p` an. Wie müsste nun -- bei ansonsten gleichem Rumpf, den Sie nicht neu hinschreiben sollen! -- die Ergebnisliste an den Ausgabeparameter übergeben werden? Schreiben Sie (nur) den Code für die Zuweisung.

### Aufgabe 2 (Klassendefinition; etwa 15%)

In einem Verwaltungsprogramm für Werkzeuge wird jedes Werkzeug als Objekt gespeichert. Zur eindeutigen Identifikation wird jedes Werkzeug mit einer ganzzahligen Nummer geführt. Die Nummer ist außerhalb des Objekts nicht sichtbar und wird bei der Konstruktion eines neuen Werkzeugs im Verwaltungsprogramms in das Objekt eingetragen. Durch die Methode `getNummer()` kann sie aus dem Objekt ausgelesen werden.

Die Werkzeuge sind im Programm in elektrische und pneumatische Werkzeuge unterteilt. Für die elektrischen Werkzeuge wird die notwendige

Versorgungsspannung im Objekt allgemein verfügbar gespeichert. Für die pneumatischen Werkzeuge wird der notwendige Luftdruck global zugänglich gespeichert.

An elektrischen Werkzeugen können derzeit nur Schleifmaschinen im System verwaltet werden. Zu jeder Schleifmaschine wird die Bezeichnung der verwendbaren Schleifbänder als dynamische Zeichenkette nach außen nicht sichtbar gespeichert. Mit der Methode `setSchleifbaender()` kann Speicherplatz für die Bezeichnung angelegt und die als Parameter übergebene Bezeichnung in das Objekt einkopiert werden. Mit der Methode `getSchleifbaender()` kann die Bezeichnung ausgelesen werden.

An pneumatischen Werkzeugen können derzeit nur ein Schrauber und eine Sprühpistole verwaltet werden. Der Schrauber und die Sprühpistole enthalten global verfügbare Angaben (reelle Zahlen) für das Drehmoment bzw. den Düsendurchmesser.

a) Vereinbaren Sie eine Klassenhierarchie für den obigen Sachverhalt in C++. Die Klassenhierarchie soll, soweit möglich, durch Vererbung redundanzfrei sein.

b) Programmieren Sie den Rumpf der Methode `setSchleifbaender()` in C++. Die Rümpfe der restlichen Methoden müssen nicht programmiert werden.

### Aufgabe 3 (Klassen, Ausnahmen; etwa 45%)

In dieser Aufgabe soll eine einfache Nachrichtenübermittlung in C++ erstellt werden. Ein Nachrichtenverteiler holt von einer Nachrichtenquelle einzelne Nachrichten ab und stellt sie dem adressierten Agenten zu.

a) Implementieren Sie eine Klasse für die Agenten. Jeder Agent besitzt einen allgemein zugänglichen Namen als statische Zeichenkette mit bis zu 100 Einzelzeichen. Jeder Agent besitzt weiterhin, nach außen nicht sichtbar, einen Empfangspuffer als dynamische Zeichenkette und einen Wert für die Kapazität des Empfangspuffer.

Der Konstruktor jedes Agenten erhält den Namen des Agenten und die Kapazität des Empfangspuffers als Parameter und speichert diese entsprechend. Zudem versieht er den Empfangspuffer mit dem angegebenen Speicherplatz.

Jeder Agent besitzt eine allgemein verfügbare Methode `puffernNachricht()`, die als Parameter eine Nachricht als Zeichenkette erhält. Falls im Empfangspuffer noch genügend Platz vorhanden ist, hängt die Methode die Nachricht an die vorhandenen Nachrichten im Empfangspuffer mit einem Zeilenvorschub als Trennzeichen an. Ansonsten wird ein Zustellfehler als Exception ausgeworfen.

Schließlich besitzt jeder Agent eine Methode `bearbeitenNachrichten()`. Diese ist prozedural, allgemein zugänglich und parameterlos. Sie entnimmt alle Nachrichten aus dem Empfangspuffer und bearbeitet sie. Der Rumpf dieser Methode muss nicht implementiert werden.

b) Vereinbaren Sie eine Klasse für die Nachrichtenquelle, die aus der allgemein verfügbaren Methode `empfangen()` besteht. Durch Aufruf dieser Methode kann aus der Nachrichtenquelle eine einzelne Nachricht für einen Adressaten abgeholt werden. Die Variablen für die Adresse und die Nachricht werden als Resultatsparameter an die Methode übergeben. Die Methode erzeugt dynamisch den Speicherplatz für beide Resultate. Die Methode `empfangen()` muss nicht implementiert werden.

c) Implementieren Sie eine Klasse für den Nachrichtenverteiler. Dieser speichert, nach außen nicht sichtbar, eine Reihung mit bis zu zehn Zeigern auf Agenten und einen Zeiger auf eine Nachrichtenquelle. Bei der Konstruktion eines Nachrichtenverteilers wird die Reihung geeignet vorbelegt und die Nachrichtenquelle mit dem Zeiger belegt, der als Parameter an den Konstruktor übergeben wird.

Ein Nachrichtenverteiler enthält zudem die global verfügbare Methode `anmelden()`. Sie erhält einen Agenten als Parameter und speichert ihn, falls möglich, an beliebiger Stelle in der internen Reihung. Sind alle Plätze der Reihung belegt, so wirft die Methode einen Anmeldefehler als Exception aus.

Schließlich enthält ein Nachrichtenverteiler eine allgemein zugängliche, parameter- und ergebnislose Methode `zustellen()`. Sie empfängt aus der Nachrichtenquelle, die im Objekt gespeichert ist, eine Nachricht für einen Adressaten und stellt sie dem entsprechenden Agenten zu, der in der internen Reihung gespeichert ist. Ist der Adressat nicht in der Reihung vorhanden, so gibt sie eine Fehlermeldung am Bildschirm aus.

d) Implementieren Sie das Hauptprogramm der Nachrichtenverteilung. Erzeugen Sie eine Nachrichtenquelle und einen Nachrichtenverteiler für die Quelle. Erzeugen Sie die Agenten `a1`, `a2`, `a3` und melden Sie diese am Nachrichtenverteiler an. Tritt ein Anmeldefehler auf, wird das Programm ohne weitere Aktionen beendet.

Tritt kein Anmeldefehler auf, so rufen Sie solange die Methode `zustellen()` des Nachrichtenverteilers auf, bis ein Zustellfehler entsteht. Danach aktivieren Sie in beliebiger Reihenfolge zu jedem Agenten die Methode `bearbeitenNachrichten()`.

#### Aufgabe 4 (Zahldarstellung; etwa 15%)

Nach dem Standard IEEE-754 werden Gleitkommazahlen einfacher Genauigkeit mit 8 Bit als Biased Exponent (Bias = 127) und 23 Bit als Fraction (=Mantisse ohne Hidden Bit) dargestellt. Gegeben seien die Zahlen  $A = 1,5$  und  $B = 0,015625$  und  $C = 785,0625$ .

a) Wie lauten A, B und C im Dualsystem (Festkomma-Darstellung)?

b) Wie werden A, B und C gemäß dem oben genannten IEEE-Standard im Rechner gespeichert?

(Ende des Aufgabentextes)