Fachhochschule Ingolstadt

Studiengang Elektro- und Informationstechnik
Sommersemester 2000

Prüfungsfach:	Grundlagen Informatik
Prüfungsabschnitt:	2. Teilprüfung
Termin:	19.07.2000
Zeit:	9.00 Uhr
Prüfer:	Prof. Dr. B. Glavina, Prof. Dr. J. Schweiger
Prüfungsdauer:	90 Minuten
Hilfsmittel:	4 DIN-A4 Seiten Zusammenfassung Vorlesungsskript

Aufgabe 1: (ca. 30%)

Gegeben seien einfach verkettete Listen in der Programmiersprache C. Die Listen bestehen aus Elementen *elem*, die eine natürliche Zahl *n* als Elementnummer und eine dynamische Zeichenkette *s* speichern.

a) Wie lautet die Typdeklaration der Listenelemente *elem*.
b) Schreiben Sie eine Prozedur *Top,* die in einer Liste L aus Elementen *elem* ein Element aus der Liste mit einer bestimmten Elementnummer an den Anfang der Liste stellt. Die Prozedur erhält als Parameter die Liste L und eine Elementnummer. Die Prozedur liefert als Ausgabe einen Zeiger auf eine Liste zurück, in der sich das Element am Anfang befindet. Falls kein Element mit der übergebenen Elementnummer in L enthalten ist, wird der leere Zeiger zurückgegeben.

Aufgabe 2: (ca. 20 %)

Gegeben sei folgendes Programmstück in C++ mit konventioneller Fehlerbehandlung:

```
class SB
{     private: int x;
      public:  int SubBearbeitung (char IntString[]);
               void FehlerFileTrace ( ); };
class B
{     private: int e;
      public:  int Bearbeitung (char IntString[]); };

void SB::FehlerFileTrace ()
{     // Fehlerausgabe Datei  }

int SB::SubBearbeitung (char IntString[])
{     if (sscanf(IntString, "%d", &x) != 1)  /* Fehler */ return (-1);
      else { /* Reguläre Bearbeitung */ x=0; return (0); }}

int B::Bearbeitung (char IntString[])
{     SB isb;
      e = isb.SubBearbeitung(IntString);
      if (e == 0) /* Reguläre Bearbeitung */ return (0);
      else { /* Fehler */ isb.FehlerFileTrace (); return (e); } }

void main ( )
{     char s[100]; B ib; int res;
      printf ("Zahl eingeben\n");   fflush(stdin);    gets(s);
      res = ib.Bearbeitung(s);
      if (res == -1) printf ("Fehler");
      else printf ("Weitere Bearbeitung"); }
```

Bitte wenden !!!!

Fachhochschule Ingolstadt

Studiengang Elektro- und Informationstechnik
Sommersemester 2000

Wandeln Sie das Programmstück so um, daß die konventionelle Fehlerbehandlung durch das Exception Handling in C++ ersetzt ist. Führen Sie dazu geeignete Fehlerklassen ein. Nach der Umwandlung des Programmstücks liefern die Methoden der Klassen kein Ergebnis zurück.

Aufgabe 3: (ca. 35 %)

In dieser Aufgabe soll ein einfacher Planer für die Vergabe der Rechner der EDV-Labore in C++ erstellt werden. Der Planer speichert die Anzahl der freien Rechner eines Labors über 1 Woche, Montag bis Freitag, im Stundenraster von 8:00 bis 18:00 Uhr. Jede Stunde eines Tages ist mit der Anzahl der verfügbaren Rechner versehen. Jedes EDV-Labor beinhaltet 20 Rechner. Der Planer verhandelt mit zwei anderen Planern die Belegung der freien Rechner.

a) Deklarieren Sie eine Klasse *LaborPlaner*, die die Anzahl der freien Rechner über eine Woche speichert. Der Wochenplan ist außerhalb der Klasse nicht sichtbar und auch für Unterklassen nicht zugänglich. Die Klasse enthält Methoden entsprechend der nachfolgenden Aufgabenstellung.
b) Implementieren Sie einen Konstruktor zur Klasse, der alle Rechner des Labors für alle Stunden als verfügbar kennzeichnet.
c) Implementieren Sie eine Methode *BelegeStunde,* die für eine Stunde eines Tages eine übergebene Anzahl von Rechnern belegt.
d) Implementieren Sie eine Methode *IstFrei,* die für eine Stunde eines Tages die Anzahl freier Rechner bestimmt.
e) Implementieren Sie eine Methode *VerhandleBelegung.* Diese erhält als Parameter ein Feld mit zwei weiteren Laborplanern und eine Anzahl zu belegender Rechner. Sie liefert als Resultat zurück, ob eine Belegung gefunden wurde. Die Methode plant in Zusammenarbeit mit den übergebenen Planern eine Stunde, in der alle Labore gemeinsam die geforderte Anzahl von Rechner bereitstellen können und belegt diese entsprechend in den Wochenplänen.
f) Implementieren Sie eine Unterklasse *PraktikumsPlaner* zur Klasse *LaborPlaner*, in der zusätzlich noch der Namen des Dozenten und die Klasse (beide max. 10 Zeichen) gespeichert werden. Diese zusätzlichen Daten sollen nur innerhalb der Klasse und für weitere Unterklassen verfügbar sein.

Aufgabe 4: (ca. 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 = 1000,5 und B = 0,53125.

a) Wie lauten A und B im Dualsystem (Festkomma-Darstellung)?
b) Wie werden A und B gemäß dem oben genannten IEEE-Standard im Rechner gespeichert? Geben Sie die 32 Bits je Zahl auch als Hexadezimalzahl an.
c) Geben Sie die IEEE-Darstellung von C = A + B an, und zwar als Ergebnis der Addition der beiden IEEE-Darstellungen von A und B.