

**Fachhochschule Ingolstadt**  
**Studiengang Elektro- und Informationstechnik**  
**Sommersemester 1999**

Prüfungsfach: Grundlagen Informatik  
Prüfungsabschnitt: 2. Teilprüfung  
Termin: 13.7.1999  
Zeit: 9.00 Uhr  
Prüfer: 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 ganze Zahl *x* und ein Einzelzeichen *c* speichern.

- a) Wie lautet die Typdeklaration der Listenelemente *elem*.
- b) Schreiben Sie eine Funktion *InsertListe* die eine Liste L2 in eine Liste L1 einfügt. L1 und L2 bestehen aus Elementen *elem*. Die Liste L2 wird entweder vor einem Element E einhängt oder, falls E in der Liste nicht vorhanden ist, am Ende angefügt. Die Funktion erhält als Parameter die Listen L1 und L2 als Zeiger und E als Element ohne Zeiger. Die Funktion liefert als Ergebnis einen Zeiger auf die zusammengesetzte Liste. Die Funktion kann die Listen L1 und L2 zerstören.

Aufgabe 2: (ca. 20 %)

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

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

int SB::SubBearbeitung (char IntString[])
{
    if (sscanf(IntString, "%d", &x) != 1)
        /* Umwandlungsfehler */ return (-1);
    else if (x > 5 || x < -5)
        /* Bereichsfehler */ return (-2);
    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 */ return (e);
}
```

**Bitte wenden !!!!**

**Fachhochschule Ingolstadt**  
**Studiengang Elektro- und Informationstechnik**  
**Sommersemester 1999**

```
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 ("Umwandlungsfehler");
    else if (res == -2) printf ("Bereichsfehler");
    else printf ("Weitere Bearbeitung");
}
```

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

Aufgabe 3: (ca. 35 %)

In dieser Aufgabe soll ein einfacher beruflicher Terminplaner in C++ erstellt werden. Der Terminplaner speichert Belegungen über 1 Woche, Montag bis Freitag, im Stundenraster von 8:00 bis 18:00 Uhr. Eine Stunde eines Tages ist entweder *belegt* oder *frei*. Der Terminplaner verhandelt einen für ihn günstigen Termin mit zwei anderen Terminplanern.

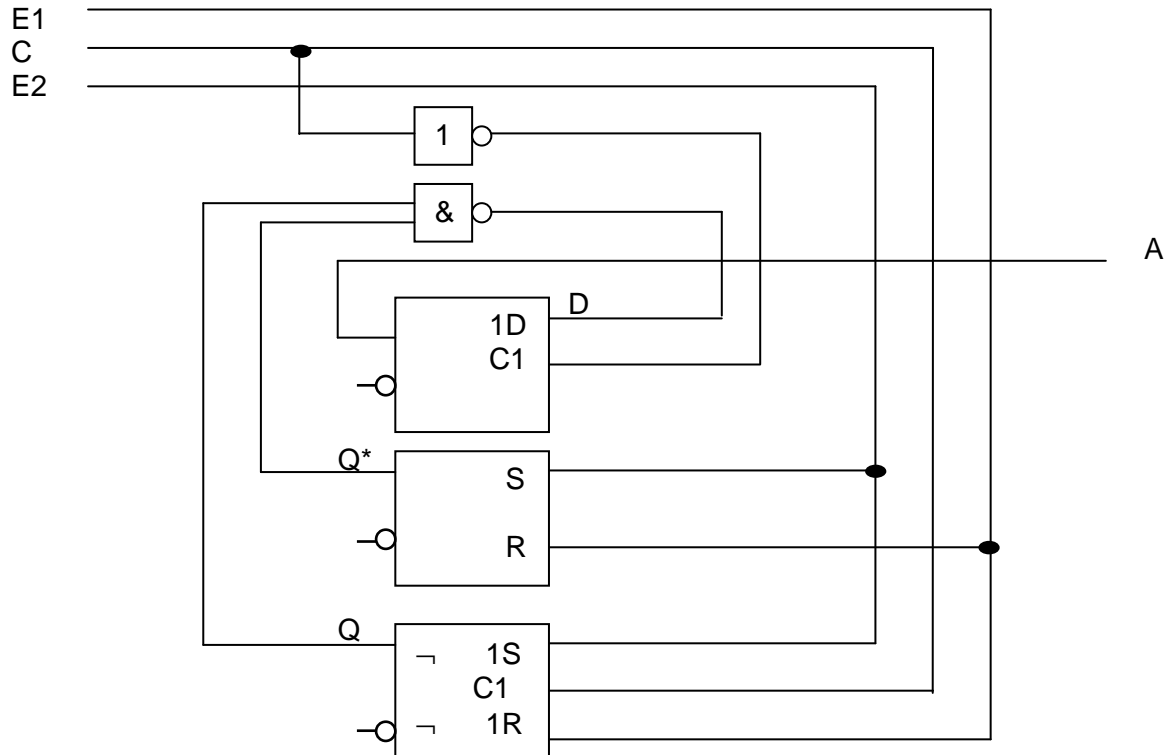
- a) Deklarieren Sie eine Klasse *Terminplaner*, die obige Belegungen für eine Woche speichert. Der Wochenplan ist außerhalb der Klasse nicht sichtbar, aber für Unterklassen zugänglich. Die Klasse enthält u.a. die Methode *VerhandleTermin*. Diese erhält als Parameter ein Feld mit zwei weiteren Terminplanern und liefert als Resultat zurück, ob ein Termin gefunden wurde. Weitere Methoden der Klasse sind nach Bedarf festzulegen.
- b) Implementieren Sie einen Konstruktor zur Klasse, der alle Stunden als *frei* kennzeichnet.
- c) Implementieren Sie die Methode *VerhandleTermin*. Die Methode sucht zunächst nach einer freien Stunde zu Beginn oder am Ende eines Arbeitstages oder zwischen zwei belegten Stunden. Wird keine derartige Stunde gefunden, so wird eine beliebige freie Stunde gewählt. Ist diese Stunde auch bei den anderen beiden Terminplanern frei, so wird sie belegt. Ist diese Stunde bei den anderen Terminplanern bereits belegt, so wird nach obiger Strategie nach einer anderen Stunde gesucht. Die Methode liefert zurück, ob ein Termin gefunden werden konnte.
- d) Implementieren Sie eine Unterklasse *ArbeitskreisPlaner* zur Klasse *Terminplaner*, in der zusätzlich noch der Namen des Vorsitzenden (max. 10 Stellen) und die Anzahl der Mitarbeiter des Arbeitskreises gespeichert werden. Diese zusätzlichen Daten sollen allgemein verfügbar sein.

**Bitte nächstes Blatt !!!!**

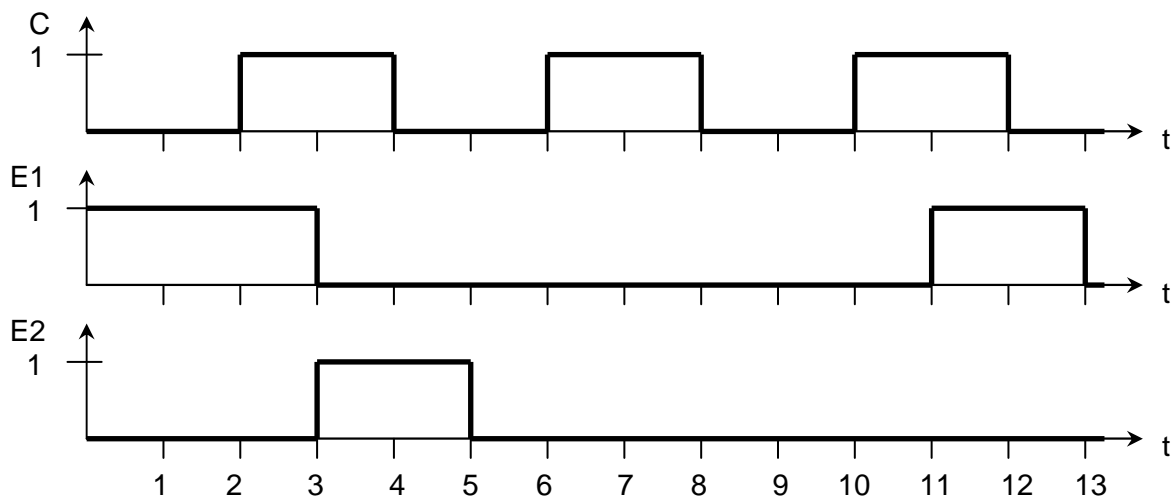
**Fachhochschule Ingolstadt**  
**Studiengang Elektro- und Informationstechnik**  
**Sommersemester 1999**

Aufgabe 4: (ca. 15%)

Gegeben sei folgendes Schaltwerk:



An den Eingängen liegt folgender Signalverlauf an:



Stellen Sie den Signalverlauf an den Stellen Q\*, Q, D und A graphisch dar. Zum Zeitpunkt  $t = 0$  seien  $Q = Q^* = 0$  und  $A = 1$ .