

Hinweise für den Prüfling

Auswahlzeit: 30 Minuten

Bearbeitungszeit (insgesamt): 240 Minuten

Auswahlverfahren

Es gibt drei Aufgabengruppen A, B und C, aus denen jeweils ein Vorschlag zu bearbeiten ist. Die vorliegenden Vorschläge aus den Gruppen A (Objektorientierte Modellierung) und B (Konzepte und Anwendungen der theoretischen Informatik) sind Pflichtvorschläge.

Wählen Sie von den zwei vorliegenden Vorschlägen der Gruppe C (Datenbanken) einen zur Bearbeitung aus. Der nicht ausgewählte Vorschlag muss am Ende der Auswahlzeit der Aufsicht führenden Lehrkraft zurückgegeben werden.

Erlaubte Hilfsmittel

1. ein Wörterbuch der deutschen Rechtschreibung
2. eine Liste der fachspezifischen Operatoren

Sonstige Hinweise

ohne PC-Nutzung

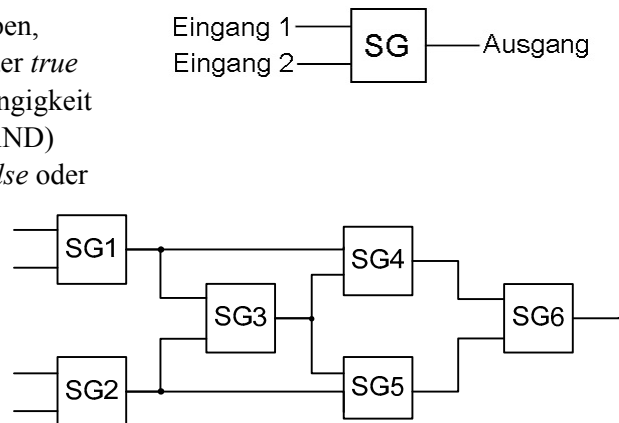
In jedem Fall vom Prüfling auszufüllen

Name: _____	Vorname: _____
Prüferin/Prüfer: _____	Datum: _____

Digitalsimulator – Delphi

Es soll ein Simulator für digitale Schaltungen mit Schaltgliedern, die zwei Eingänge und einen Ausgang haben, entwickelt werden. Jeder Eingang kann mit *false* oder *true* belegt werden. Ein Schaltglied (SG) liefert in Abhängigkeit von seiner Bauart (z.B. AND, OR, NOT, XOR, NAND) und der Belegung der beiden Eingänge den Wert *false* oder *true* am Ausgang.

Die nebenstehende Abbildung zeigt einen Ausschnitt einer digitalen Schaltung.



Aufgaben

1. In Material 1 ist die Implementierung der Klasse *TSchaltglied* gegeben. Mit der nachstehenden Anweisungsfolge werden die Objekte der gegebenen Schaltung erzeugt.

```
SG1:= TSchaltglied.create('G1');
SG2:= TSchaltglied.create('G2');
SG3:= TSchaltglied.create('G3');
SG4:= TSchaltglied.create('G4');
SG5:= TSchaltglied.create('G5');
SG6:= TSchaltglied.create('G6');
```

Geben Sie eine Anweisungsfolge an, mit der alle Verbindungen der obigen Schaltglieder SG1 bis SG6 hergestellt werden.

(7 BE)

2. Stellen Sie die in Material 1 und 2 gegebenen Klassen und ihre Beziehungen in einem UML-Klassendiagramm dar.

Erläutern Sie die Beziehungen.

(8 BE)

3. Digitale Schaltungen können alleine mit NAND-Gliedern realisiert werden. Ein NAND-Glied hat am Ausgang den Wert *false*, wenn beide Eingänge den Wert *true* haben. Andernfalls hat der Ausgang den Wert *true*. Ist ein Eingang unbeschaltet, so hat dieser den Wert *false*.
- 3.1 Stellen Sie die Klasse *TNANDGlie*d in Ihrem UML-Diagramm dar und erläutern Sie Ihre Überlegungen.
Implementieren Sie die Klasse *TNANDGlie*d mit allen notwendigen Methoden. **(7 BE)**
- 3.2 Die gegebene Schaltung bestehe nur aus NAND-Gliedern.
Bestimmen Sie nachvollziehbar den Wert des Ausgangs von SG6, wenn die Ausgänge von SG1 und SG2 den Wert *true* haben. **(3 BE)**
4. Analysieren Sie die Konzeption und Implementierung der Klasse *TSchaltgliederliste* in Material 2. **(5 BE)**
5. Implementieren Sie die Methode *aktualisiereVerbundeneSchaltglieder* der Klasse *TSchaltglied* in Material 1, welche nach einer Änderung des Ausgangs den Zustand aller betroffenen Schaltglieder aktualisiert. **(5 BE)**

Material

Hinweise

Wird *self* innerhalb einer Methode benutzt, so ist *self* der Verweis auf das Objekt, das diese Methode gerade ausführt.

Mit *inherited create(...)* kann in einem Konstruktor einer Unterklasse ein Konstruktor der Oberklasse aufgerufen werden.

Material 1

```
01 TSchaltglied = class
02 private
03   Name: string;
04   verbundenMit: TSchaltgliederListe; // Implementierung in Material 2
05 protected
06   Eingang1: TSchaltglied;
07   Eingang2: TSchaltglied;
08   Ausgang: boolean;
09   procedure aktualisiereAusgang; virtual;
10   procedure aktualisiereVerbundeneSchaltglieder;
11 public
12   constructor create(Name: string);
13   function getName: string;
14   function getAusgang: boolean;
15   procedure verbindeAusgangMitEingang(Nr: integer; einSchaltglied: TSchaltglied);
16 end;
17
18 ...
19
20 constructor TSchaltglied.create(Name: string);
21 begin
22   self.Name := Name;
23   Ausgang := false;
24   Eingang1 := nil;
25   Eingang2 := nil;
26   verbundenMit := TSchaltgliederListe.create;
27 end;
28
29 procedure TSchaltglied.aktualisiereAusgang;
30 begin
31   // wird durch Spezialisierung implementiert
32 end;
33
34 procedure TSchaltglied.aktualisiereVerbundeneSchaltglieder;
35 begin
36   // noch nicht implementiert, vergleiche Aufgabe 5
37 end;
38
39 procedure TSchaltglied.verbindeAusgangMitEingang
40   (Nr: integer; einSchaltglied: TSchaltglied);
41 begin
42   verbundenMit := verbundenMit.neuesSchaltglied(einSchaltglied);
43   if Nr = 1 then
44     einSchaltglied.Eingang1 := self
45   else
46     einSchaltglied.Eingang2 := self;
47 end;
```

**Informatik
Leistungskurs****Thema und Aufgabenstellung
Vorschlag AD**

```
48
49 function TSchaltglied.getName: string;
50 begin
51   result:= Name;
52 end;
53
54 function TSchaltglied.getAusgang: boolean;
55 begin
56   result:= Ausgang;
57 end;
```

Material 2

```
01 TSchaltgliederListe = class
02 private
03   erstesSchaltglied: TSchaltglied;
04   weitereSchaltglieder: TSchaltgliederListe;
05 public
06   constructor create;
07   function neuesSchaltglied(einSchaltglied: TSchaltglied): TSchaltgliederListe;
08   function getErstesSchaltglied: TSchaltglied;
09   function getWeitereSchaltglieder: TSchaltgliederListe;
10 end;
11
12 ...
13
14 constructor TSchaltgliederListe.create;
15 begin
16   erstesSchaltglied:= nil;
17   weitereSchaltglieder:= nil;
18 end;
19
20 function TSchaltgliederListe.neuesSchaltglied(einSchaltglied: TSchaltglied):
21                                             TSchaltgliederListe;
22   var neueListe: TSchaltgliederListe;
23 begin
24   neueListe:= TSchaltgliederListe.create;
25   neueListe.erstesSchaltglied:= einSchaltglied;
26   neueListe.weitereSchaltglieder:= self;
27   result:= neueListe;
28 end;
29
30 function TSchaltgliederListe.getErstesSchaltglied: TSchaltglied;
31 begin
32   result:= erstesSchaltglied;
33 end;
34
35 function TSchaltgliederListe.getWeitereSchaltglieder: TSchaltgliederListe;
36 begin
37   result:= weitereSchaltglieder;
38 end;
```