
Aufgaben zur Übergangsprüfung **Grundlagen der Programmierung** (Wiederholung) im SS 97 (WI)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten

Aufgabe 2:

Gegeben seien die folgenden elementaren Prädikate: $dreieckig(s)$, $rot(s)$ und $klein(s)$ über der Grundmenge aller *Schilder*.

Geben Sie für die folgenden umgangssprachlichen Aussagen gleichwertige prädikatenlogische Formeln an. Alle Schilder, die nicht klein sind, sind groß.

1. Alle dreieckigen Schilder sind rot

.....

2. Kein rotes Schild ist klein

.....

3. Alle kleinen Schilder sind rot

.....

4. Alle dreieckigen Schilder sind groß

.....

5. Einige kleine Schilder sind nicht rot

.....

6. Einige große Schilder sind rot

.....

Welche der Aussagen 3. bis 6. folgen aus den Aussagen 1. und 2. ?

.....

Aufgabe 3:

Gegeben seien die folgenden Programmvariablen:

```
var f : array [0..n - 1] of Z;  
var i : N0;  
var s : Z;
```

Spezifizieren Sie die Aussage, daß das Feld f abwechselnd gerade und ungerade Werte gespeichert hat, mit einer Formel P aus der Prädikatenlogik. Der Wert des 1. Elements soll dabei keinen Einfluß auf das Prädikat haben.

Die Aussage P soll mit Hilfe eines All-Quantors formuliert werden. Hinweis: nur ein weiterer logischer Operator ist hier notwendig.

P :

.....

Formen Sie diesen Ausdruck P um in einen gleichwertigen Ausdruck P_1 ohne All-Quantor.

P_1 :

.....

Entwickeln Sie eine Spezifikation für ein Programm $Prog$, das in der Variablen s die Summe der Produkte von benachbarten Feldelementen des Feldes f berechnet.

$$\{ V \} Prog \{ P \}$$

V :

P :

.....

Formen Sie P so um, daß hieraus eine Invariante I und eine Schleifenabbruchbedingung B abgeleitet werden können. Diese Umformung soll durch Einführung einer neuen Variablen i für die *obere* Grenze gemacht werden.

$I \wedge \neg B$:

.....

Vervollständigen Sie mit Hilfe von I und B das folgende Programmstück.

```
{ V }
i, s := E0, E1; { I }
while B do
    i, s := E2, E3 { I }
end while
{ P }
```

E_0 :

E_1 :

B :

E_2 :

E_3 :

Aufgabe 4:

Lineare Suche: Entwickeln Sie eine Funktion *suche*, die für eine natürliche Zahl $n > 1$ den kleinsten Teiler von n berechnet, der größer als 1 ist.

Das Prädikat für die Suche ist also

$$P(i) : n \bmod i = 0$$

.....

.....

.....

.....

.....

.....

.....

Warum ist hier die Terminierung gesichert?

.....

.....

