

---

**Aufgaben zur Klausur in  
*Grundlagen der Programmierung (WS 2004/2005)*  
Studiengänge WI, MI, II**

Zeit: 60 Minuten,  
erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den freien Stellen nach den jeweiligen Aufgaben ein (ggf. auf der jeweiligen Rückseite weiterschreiben).

Diese Klausur besteht einschließlich dieses Deckblattes aus 5 Seiten.

Für die Klausur werden insgesamt 28 Bewertungseinheiten (BE) vergeben. Zum Bestehen benötigen Sie mindestens 14 BE.

Viel Erfolg !

## 1. Aufgabe (6 BE)

- 1.1 Formulieren Sie die Aussagen a) und b), indem Sie aus M, P und S Prädikate bilden, die Sie ausschließlich mit Operationszeichen der Prädikatenlogik verbinden (ergänzende Worte sind verboten)! Ihre Formulierung soll zu den gegebenen Aussagen äquivalent sein.
- 1.2 Geben Sie für jede Aussage an, ob sie wahr oder falsch ist. Begründen Sie Ihre Antwort jeweils durch logische Argumente oder die Angabe von Mengendiagrammen. Geben Sie gegebenenfalls ein Gegenbeispiel an (auch hierfür ist ein Mengendiagramm erlaubt)!
- a) Aus der Tatsache, dass alle M auch P sind und einige M auch S sind, folgt: Einige S sind auch P.

- b) Aus der Tatsache, dass kein M auch P ist und kein S auch M ist, folgt: Kein S ist auch P.

## 2. Aufgabe (6 BE)

Finden Sie zum folgendem Programmausschnitt und der gegebenen Nachbedingung die schwächste Vorbedingung! Geben Sie alle Zwischenschritte Ihrer Beweiskette an!

```
if  $x \geq 0$ 
```

```
  then
```

```
     $x := x * x$ 
```

```
  else
```

```
     $x := -x * x;$ 
```

```
 $y := -x;$  {gehört nicht mehr zur Verzweigung}
```

```
{ $y \geq 0$ }
```

### 3. Aufgabe (8 BE)

Gegeben sei die folgende Prozedur:

```
{n sei eine Zahl vom Typ integer}
procedure aufgabe3 (n): integer
begin
  if (n = 0)
    then return 0
    else return (n + aufgabe3 (n-1))
end {aufgabe3}
```

3.1 Was berechnet diese Prozedur? Nennen Sie gegebenenfalls Vorbedingungen!

3.2 Beweisen Sie 3.1 mit Hilfe einer Ihnen geläufigen Verifikationstechnik.

Hinweis: Falls Ihnen das mehr liegt, dürfen Sie Ihren Beweis auch an einer äquivalenten iterativen Version durchführen (ich rate dazu aber nicht).

3.3 Wenn Sie die Originaldarstellung in Pascal implementieren, sollte die Parameterübergabe dann call-by-value oder call-by-reference erfolgen? Begründen Sie Ihre Entscheidung!

## 4. Aufgabe (8 BE)

4.1 Beschreiben Sie die Algorithmen Selectionsort und Mergesort! Es reicht eine verbale Beschreibung, die Angabe von Pseudocode ist nicht erforderlich.

4.2 Geben Sie die Komplexität der beiden Algorithmen an! Begründen Sie, welche Vorteile das gewählte Komplexitätsmaß zur Bewertung von Algorithmen hat!