
**Aufgaben zur Klausur in
Grundlagen der Programmierung (SS 2006)
Studiengänge II, WI**

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 Deckblatts aus 6 Seiten.

Für die Klausur werden insgesamt 30 Bewertungseinheiten (BE) vergeben. Zum Bestehen benötigen Sie mindestens 15 BE.

Viel Erfolg !

2. Aufgabe (4 BE)

Gegeben seien die folgenden Prädikate mit den zugehörigen Bedeutungen:

$B(x,y)$ Studierende(r) x hat das Fach y bestanden.

$T(x,y)$ Studierende(r) x hat an der Prüfung im Fach y teilgenommen.

$K(y)$ y ist ein Klausurfach (Negation: y hat eine mündliche Prüfung)

Außerdem beschreibe die Menge S alle Studierenden und die Menge F alle Fächer.

Beschreiben Sie die folgenden Aussagen mit jeweils einem prädikatenlogischen Ausdruck, der *ausschließlich* die Prädikate B , T , K oder M verwendet! Hierbei seien Alex und Linda Studierende aus S und GTI, DM und CG Fächer aus F .

a) In GTI und DM werden Klausuren geschrieben, während CG eine mündliche Prüfung hat.

b) Alex hat alle Fächer bestanden, die Klausurfach sind.

c) Linda hat jedes Fach bestanden, an dessen Prüfung sie teilgenommen hat.

d) Wenn ein Studierender ein Fach bestanden hat, dann hat er an der Prüfung teilgenommen.

3. Aufgabe (6 BE)

- a) Finden Sie zum folgenden Programmausschnitt und der gegebenen Nachbedingung die schwächste Vorbedingung!
Achtung: Vereinfachen Sie nur, was wirklich zulässig ist!
Geben Sie alle Zwischenschritte Ihrer Beweiskette an! (4 BE)
- b) Geben Sie eine Belegung für x und y an, welche die gegebene Nachbedingung erfüllt und die then-Anweisung durchläuft und geben Sie eine Belegung an, welche die gegebene Nachbedingung erfüllt und die else-Anweisung durchläuft! (2 BE)

```
if x > y
```

```
  then
```

```
    x := - x
```

```
  else
```

```
    y := - y;
```

```
{x > y}
```

4. Aufgabe (7 BE)

Gegeben sei die folgende Prozedur:

```
procedure f (n: integer): integer
begin
  result := 1;
  k := 0;
  while (k < n) do
    begin
      result := result * k;
      k := k+1;
    end;
  return result;
end {f}
```

a) Was berechnet diese Prozedur?

Geben Sie die Lösung für genau die Vorbedingung an, die hier spezifiziert ist! (2 BE)

b) Beweisen Sie Ihre Lösung von a) für $n > 0$ mit folgendem Verfahren:

Seien \mathbf{result}_i und \mathbf{k}_i die Werte der Variablen nach dem i -ten Schleifendurchlauf.

Bestimmen Sie diese Werte und beweisen Sie diese durch vollständige Induktion über i . Folgern Sie daraus das Resultat aus a)!

(5 BE)

5. Aufgabe (4 BE)

Gegeben sei die folgende Prozedur:

```
procedure mysterious (data, left, right, k): integer
begin
  if left > right then return 0;
  mid := (left + right) div 2;
  if k = data[mid] then return mid;
  pos := mysterious (data, left, mid-1, k);
  if pos ≠ 0 then return pos;
  return mysterious (data, mid+1, right, k);
end {mysterious}
```

a) Was berechnet diese Prozedur?

Geben Sie die schwächstmöglichen Vorbedingungen für die Parameter data, left, right und k an! (2 BE)

b) Wenn Sie die Vorbedingung in a) verschärfen, kann die Prozedur mysterious zu einem Algorithmus modifiziert werden, der in der Algorithmienlehre unter einem bestimmten Namen bekannt ist:

Worin besteht die Verschärfung und wie heißt der Algorithmus? (2 BE)
(Der Code des Algorithmus muss nicht angegeben werden)