

Prof. Dr. Claudia Müller-Birn, Barry Linnert

# Konzepte der imperativen und objektorientierten Programmierung SoSe 2017 Übungsblatt 05

Abgabe: 31.05.2017

Geben Sie immer die Namen aller Mitglieder Ihrer Gruppe an!

---

## 1 Programmverifikation mittels Hoare-Kalkül (3 Punkte)

```
1 if (x%2==0)
2   y = y+y
3   x = x // 2
4 else
5   z = z+y
6   x = x-1
```

Nutzen Sie als Vorbedingung  $\{P\}=\{x \geq 0 \wedge c=xy+z\}$  und als Nachbedingung  $\{Q\}=\{x \geq 0 \wedge c=xy+z\}$ .

## 2 Programmverifikation mittels Hoare-Kalkül (3 Punkte)

```
1 if x >= 0 :
2   y = x
3 else :
4   y = -x
```

Nutzen Sie als Vorbedingung  $\{P\}=\{\text{True}\}$  und als Nachbedingung  $\{Q\}=\{y \geq 0\}$ .

### 3 Programmverifikation mittels Hoare-Kalkül

(6 Punkte)

```
1 def gauss(n) :  
2     assert(n>0)  
3     i = 1  
4     sum = 0  
5     while i<=n :  
6         sum = sum+i  
7         i = i+1  
8     return sum
```

Nutzen Sie als Nachbedingung  $\{Q\} = \{\text{sum} = n(n+1)/2\}$  und als Invariante  $\{INV\} = \{\text{sum} = i(i-1)/2 \wedge i \leq n+1\}$ . Wählen Sie eine geeignete Vorbedingung selbst.

### 4 Türme von Hanoi in Python

(18 Punkte)

Überführen Sie Ihr Programm zur Lösung des Problems der Türme von Hanoi (Übungsblatt 04, Aufgabe 2) in einer iterativen Version ohne Nutzung von Rekursion. Als Ausgang Ihrer Betrachtungen können Sie Ihre eigene Lösung oder eine im Tutorium vorgestellt nutzen. Sollten Sie das Problem der Türme von Hanoi bereits iterativ gelöst haben, schätzen Sie den Speicherverbrauch (Stack und Datenbereich) Ihrer Lösung im Vergleich zu einer rekursiven Lösung begründet ab.