



**Aufgabe 31 (Nutzung eines Beweissystems (1/2) [klausurähnlich], 3+5 Punkte)**

Beweisen oder widerlegen Sie die partielle Korrektheit folgender Hoare-Tripel.

a)  $\{z=x\} \ y := x + 5; \ x := y + 5; \ \{z=x-10\}$

b)  $\{x>0 \wedge x<16\}$   
**if**  $(x>8)$  {  
     $x := x - 8;$   
} **else** {  
     $x := x + 3;$   
}  
 $\{x>0 \wedge x<12\}$

$$\frac{}{\{p[x:=y]\} \ x := y; \ \{p\}}$$

$$\frac{\{p\} \text{ Prog1 } \{q\}, \{q\} \text{ Prog2 } \{r\}}{\{p\} \text{ Prog1 Prog2 } \{r\}}$$

$$\frac{}{\{p\} \text{ Prog1 Prog2 } \{r\}}$$

$$\frac{\{p \wedge B\} \text{ Prog1 } \{q\}, \{p \wedge \neg B\} \text{ Prog2 } \{q\}}{\{p\} \text{ if } (B) \{ \text{Prog1} \} \text{ else } \{ \text{Prog2} \} \{q\}}$$

$$\frac{}{\{p\} \text{ if } (B) \{ \text{Prog1} \} \text{ else } \{ \text{Prog2} \} \{q\}}$$

$$\frac{}{\{p \wedge B\} \text{ Prog } \{p\}}$$

$$\frac{}{\{p\} \text{ while } (B) \{ \text{Prog} \} \{p \wedge \neg B\}}$$

$$\frac{p \rightarrow p1, \{p1\} \text{ Prog } \{q1\}, q1 \rightarrow q}{\{p\} \text{ Prog } \{q\}}$$

$$\frac{}{\{p\} \text{ Prog } \{q\}}$$

**Aufgabe 32 (Nutzung eines Beweissystems (1/2) [klausurähnlich], 3+10 Punkte)**

Beweisen oder widerlegen Sie die partielle Korrektheit folgender Hoare-Tripel.

a)  $\{z=x\} \ y := x + 1; \ x := y - 1; \ \{z=x\}$

b)  $\{z=x \wedge x>0\}$   
 $u := 0;$   
 $y := x;$   
**while**  $(\text{not } (y==0))$  {  
     $y := y - 1;$   
     $u := u + 3;$   
}  
 $\{u = 3z\}$

$$\frac{}{\{p[x:=y]\} \ x := y; \ \{p\}}$$

$$\frac{\{p\} \text{ Prog1 } \{q\}, \{q\} \text{ Prog2 } \{r\}}{\{p\} \text{ Prog1 Prog2 } \{r\}}$$

$$\frac{}{\{p\} \text{ Prog1 Prog2 } \{r\}}$$

$$\frac{\{p \wedge B\} \text{ Prog1 } \{q\}, \{p \wedge \neg B\} \text{ Prog2 } \{q\}}{\{p\} \text{ if } (B) \{ \text{Prog1} \} \text{ else } \{ \text{Prog2} \} \{q\}}$$

$$\frac{}{\{p\} \text{ if } (B) \{ \text{Prog1} \} \text{ else } \{ \text{Prog2} \} \{q\}}$$

$$\frac{}{\{p \wedge B\} \text{ Prog } \{p\}}$$

$$\frac{}{\{p\} \text{ while } (B) \{ \text{Prog} \} \{p \wedge \neg B\}}$$

$$\frac{p \rightarrow p1, \{p1\} \text{ Prog } \{q1\}, q1 \rightarrow q}{\{p\} \text{ Prog } \{q\}}$$

$$\frac{}{\{p\} \text{ Prog } \{q\}}$$

**Aufgabe 33 (Nutzung eines Beweissystems)**

Beweisen oder widerlegen Sie die partielle Korrektheit folgender Hoare-Tripel.

a)  $\{x>0 \wedge x<16\}$   
**if**  $(x>8)$  {  
     $x := x - 8;$   
} **else** {  
     $x := x + 3;$   
}  
 $\{x>0 \wedge x<11\}$



- b)  $\{z=3\}$   
   $y := x + 5;$   
   $x := y + 5;$   
   $\{z=3\}$
- c)  $\{y=0 \wedge u>0\}$   
   $x := 0;$   
  **while**(**not**( $x == u$ )) {  
     $y := y + 2;$   
     $x := x + 1;$   
  }  
   $\{y = 2*u\}$
- d)  $\{y=0\}$   
   $x := 0;$   
  **while**(**not**( $x == u$ )) {  
     $y := y + 2;$   
     $x := x + 1;$   
  }  
   $\{y = 2*u\}$
- e)  $\{true\}$   
  **if** ( $x>3$ ) {  
     $y := 1;$   
  } **else** {  
    **if**( $x<5$ ) {  
       $y := 0;$   
    } **else** {  
       $y := 42;$   
    }  
  }  
   $\{y=0 \vee y=1\}$

**Aufgabe 34 (👉 eigenes Verifikationswerkzeug; interessante Idee, kurzer Aufwand)**

Eine Möglichkeit Korrektheit automatisch zu zeigen ist es alle Möglichkeiten für alle Werte auszuprobieren, was nur für endliche, genauer recht kleine Wertebereiche möglich beziehungsweise sinnvoll ist. Schreiben Sie so ein Prüfprogramm, das nur Programme prüfen kann, die maximal die 4 int Variablen v,x,y,z als Klassenvariablen haben, deren Wertebereich nur 0 bis 9 jeweils einschließlich umfassen. Die Idee ist dann eine Vorbedingung zu prüfen, wenn diese erfüllt ist, dann die zu testende Methode auszuführen, um danach eine Nachbedingung zu prüfen. Sollte diese Prüfung scheitern, wird das Programm beendet und auf den Fehler hingewiesen. Erweitern Sie dazu die Klasse main.Checker um die benötigten Prüfungen und prüfen Sie die dort angegebene Methode.

**package** main;

```
public class Checker {  
  private static int v;  
  private static int x;  
  private static int y;  
  private static int z;
```



```
public static boolean vorbedingung() { // ein Beispiel
    return true;
}

public static boolean nachbedingung() { // ein Beispiel
    return z >= v && z >= x && z >= y;
}

public static void zuTesten() { // ein Beispiel hier: zu
                                // verifizierendes Programm
    if(v >= x) {
        z = v;
    }
    if(x >= y) {
        z = x;
    }
    if(y >= v) {
        z = y;
    }
}

public static boolean checken() {
    // TODO
    return true;
}

public static void main(String... strings) {
    if(checken()) {
        System.out.println("ok");
    }
}
}
```