Aufgabe 1

(a) Definieren Sie die Begriffe *„partielle Korrektheit"* und *„totale Korrektheit"* und grenzen Sie sie voneinander ab.

partielle Korrektheit Ein Programmcode wird bezüglich einer Vorbedingung P und einer Nachbedingung Q partiell korrekt genannt, wenn bei einer Eingabe, die die Vorbedingung P erfüllt, jedes Ergebnis die Nachbedingung Q erfüllt. Dabei ist es noch möglich, dass das Programm nicht für jede Eingabe ein Ergebnis liefert, also nicht für jede Eingabe terminiert.

totale Korrektheit Ein Code wird total korrekt genannt, wenn er partiell korrekt ist und zusätzlich für jede Eingabe, die die Vorbedingung P erfüllt, terminiert. Aus der Definition folgt sofort, dass total korrekte Programme auch immer partiell korrekt sind.

(b) Geben Sie die Verifikationsregel für die abweisende Schleife `while(b) { A }` an.

Anstattdessen verwendet man eine Art Vollständige Induktion um die Funktion der Schleife nachzuweisen.

Um die schwächste Vorbedingung eines Ausdrucks der Form „`while(b) { A }`" zu finden, verwendet man eine *Schleifeninvariante*. Sie ist ein Prädikat für das

$$\{I \wedge b\} A \{I\}$$

gilt. Die Schleifeninvariante gilt also sowohl vor, während und nach der Schleife.

(c) Erläutern Sie kurz und prägnant die Schritte zur Verifikation einer abweisenden Schleife mit Vorbedingung P und Nachbedingung Q.

Schritt 0: Schleifeninvariante I finden

Schritt 1: I gilt vor Schleifenbeginn,
d.h: $P \rightarrow \text{wp}($`"Code vor Schleife"`$, I)$

Schritt 2: I gilt nach jedem Schleifendurchlauf
d.h. $I \wedge b \Rightarrow \text{wp}($`"Code in der Schleife"`$, I)$

Schritt 3: Bei Terminierung der Schleife liefert Methode das gewünschte Ergebnis,
d.h. $I \wedge \neq b \Rightarrow \text{wp}($`"Code nach der Schleife"`$, !)$

(d) Wie kann man die Terminierung einer Schleife beweisen?

Zum Beweis der Terminierung einer Schleife muss eine Terminierungsfunktion T angeben werden:

$$T : V \rightarrow \mathbb{N}$$

V ist eine Teilmenge der Ausdrücke über die Variablenwerte der Schleife

Die Terminierungsfunktion muss folgende Eigenschaften besitzen:

- Ihre Werte sind natürliche Zahlen (einschließlich 0).
- Jede Ausführung des Schleifenrumpfs verringert ihren Wert (streng monoton fallend).
- Die Schleifenbedingung ist false, wenn $T = 0$.

T ist die obere Schranke für die noch ausstehende Anzahl von Schleifendurchläufen.

Beweise für Terminierung sind nicht immer möglich!

(e) Geben Sie für das folgende Suchprogramm die nummerierten Zusicherungen an. Lassen Sie dabei jeweils die invariante Vorbedingung P des Suchprogramms weg. Schreiben Sie nicht auf dem Aufgabenblatt!

$P \equiv n > 0 \wedge a_0 \dots a_{n-1} \in \mathbb{Z}^n \wedge m \in \mathbb{Z}$

```
i = -1;
// (1)
j = 0;
// (2)
while (i == -1 && j < n) // (3)
{ // (4)
  if (a[j] == m) {
    // (5)
    i = j;
    // (6)
  }
  else {
    // (7)
    j = j + 1;
    // (8)
  }
  // (9)
}
```

```
public class Verfikation {

  public static void methode(int n, int m) {
    int[] a = new int[n];

    a[4] = 10;
    int i = -1;
    // (1)
    int j = 0;
    // (2)
    while (i == -1 && j < n) // (3)
    { // (4)
      if (a[j] == m) {
        // (5)
        i = j;
        // (6)
      } else {
        // (7)
        j = j + 1;
        // (8)
```

```
      }
      // (9)
      System.out.println(j);
    }
  }

  public static void main(String[] args) {
    methode(10, 10);
  }
}
```

$$Q \equiv P \wedge (i = -1 \wedge \forall\, 0 \leq k < n \colon a_k \neq m) \vee (i \geq 0 \wedge a_i = m)$$