

# KV Software Engineering

## Übungsaufgaben SS 2005

Martin Glinz, Silvio Meier,  
Nancy Merlo-Schett, Katja Gräfenhain

### Übung 1

#### Aufgabe 1 (10 Punkte)

Lesen Sie das Originalpapier von Dijkstra "Go To Statement Considered Harmful", *Communications of the ACM* **11**, 3 (March 1968). 147-148.

- Formulieren Sie die Kernthese des Papiers mit Ihren Worten. (2 Punkte)
- Betrachten Sie folgendes Basic-Programm zur Berechnung des ggT's zweier Ganzzahlen und beurteilen Sie es im Hinblick auf Dijkstra's Artikel. (2 Punkte)

```
5 print "ENTER U: "  
10 input U  
15 print "ENTER V: "  
20 input V  
30 if U<V then goto 100  
40 U=U-V  
50 if U=0 then goto 120  
60 goto 30  
100 T=U  
105 U=V  
110 V=T  
115 goto 40  
120 print "GCD: ";V
```

- c) Untersuchen Sie nachfolgendes Java-Programm. Beim Aufruf der Methode `getInventoryValue()` wird Dijkstra's Kernthese widersprochen. Inwiefern? (2 Punkte)

```
public abstract class PriceCalculator {
    float a = 0.1f;
    float m = 12f;

    public float calculatePrice(float quantity) {
        return a * quantity + m;
    }
}

public class OtherPriceCalculator extends PriceCalculator {
    float v = 1.25f;

    public float calculatePrice(float quantity) {
        return a * v * quantity + m;
    }
}

public class InventoryControl {

    PriceCalculator pc;

    float inventoryQuantity = 500;

    public void setPriceCalculator(OtherPriceCalculator pc) {
        this.pc = pc;
    }

    public float getInventoryValue() {
        return pc.calculatePrice(inventoryQuantity);
    }
}
```

- d) Betrachten Sie folgendes Codefragment (Modula-2) und vergleichen Sie es mit Dijkstra's Position. (2 Punkte)

```
MODULE Sort;

FROM InOut
  IMPORT WriteLn, WriteInt;

CONST
  sortelements = 5;

TYPE
  listsize = [0..sortelements];
  sortARRAY = ARRAY listsize OF INTEGER;
VAR
  sortlist: sortARRAY;
  i : INTEGER;
  len : INTEGER;

PROCEDURE Swap(VAR a,b : INTEGER);
  VAR t : INTEGER;
  BEGIN
    t := a; a := b; b := t;
  END Swap;

PROCEDURE Qsort(VAR a: sortARRAY; l,r: INTEGER);
  (* quicksort the ARRAY A from start to finish *)
  VAR i,j: INTEGER;
      x,w: INTEGER;
  BEGIN
    i:=l; j:=r;
    x:=a[(l+r) DIV 2];
    REPEAT
      WHILE a[i]<x DO i := i+1; END;
      WHILE x<a[j] DO j := j-1; END;
      IF i<=j THEN
        Swap(a[i], a[j]);
        i := i+1; j:= j-1;
      END;

      UNTIL i>j;
      IF l <j THEN Qsort(a,l,j); END;
      IF i<r THEN Qsort(a,i,r); END;
    END Qsort;

PROCEDURE Quicksort(VAR s : sortARRAY; VAR maxSize : INTEGER);
  BEGIN
    Qsort(sortlist,0,maxSize);
  END Quicksort;

END Sort.
```

- e) Welches der 3 Beispiele (b-d) lässt sich am besten mit Dijkstra's These vereinbaren? Wie lässt sich objektorientierte Programmierung dennoch rechtfertigen? (2 Punkte)

## Aufgabe 2 (4 Punkte) - Kontrollstrukturen

- a) Betrachten Sie folgendes Java-Beispiel zur Berechnung des ggT's. Weshalb lässt sich der Einsatz des break-Statements, das ja eigentlich einem goto entspricht, trotzdem mit geschlossenen Ablaufkonstrukten vereinbaren? (2 Punkte)

```
public static int ggTWert (int a, int b) {
    int divisionsRest;
    while (true) {
        divisionsRest = a % b;
        a = b;
        b = divisionsRest;
        if (divisionsRest == 0) {
            break;
        }
    }
    return a;
}
```

- b) Annehmende vs. abweisende Schleife:  
Was fällt Ihnen in folgendem Beispiel auf? Wie würden Sie es verbessern? (2 Punkte)

```
/**
 * berechnet zu einer Zahl in Dezimaldarstellung den entsprechenden
 * Wert in Binaerdarstellung
 */
public static String umwandelnInBinaerdarstellung(int zahl) {
    String binaer = "";
    int rest;
    do {
        rest = zahl % 2;
        zahl = zahl / 2;
        binaer = rest + binaer;
    } while (zahl > 0);
    return binaer;
}
```

## Aufgabe 3 (6 Punkte) - Namenwahl, Datenstrukturen, Sichtbarkeit

- a) Auf was ist bei der Namenwahl zu achten? Zählen Sie die wichtigsten Punkte auf! (2 Punkte)

b) Sichtbarkeit: Wie lautet die Ausgabe des folgenden Programmes? Nehmen Sie dazu Stellung. (2 Punkte)

```
public class Point {
    private int x;

    public static void main (String[] args) {
        Point point = new Point();
        point.getX();
    }

    void getX() {
        int x = 7;
        System.out.println (x);
        System.out.println (this.x);
    }
}
```

c) Definition von Datenstrukturen: Wo stecken die Probleme in folgendem Codefragment ? (2 Punkte)

```
/**
 * Die Klasse PhysValue bietet 2 Funktionen an, welche den Zahlenwert
 * der Lichtgeschwindigkeit in Meter pro Sekunde bzw. in Kilometer pro
 * Stunde zurueckgeben.
 */
public class PhysValue {
    public double m = 1;
    public double s = 1;
    public double km = 1000 * m;
    public double h = 60 * 60 * s;
    public double c = 299792458 * m / s;

    public double getCinMS() {
        return c / ( m / s );
    }

    public double getCinKmH() {
        return c / ( km / h );
    }

    public static void main (String[] args) {
        PhysValue phys = new PhysValue();
        System.out.println("Die Lichtgeschwindigkeit betraegt " +
            phys.getCinMS() + " m/s oder " + phys.getCinKmH() + " km/h.");
    }
}
```