

## Musterlösung

### Aufgabe 1

Gegeben sei das folgende Programm:

```
program WasPassiert(input,output);  
{ Berechnet das arithmetische Mittel  
  zweier natürlicher Zahlen a und b. }  
var  
  a:integer;  
  b:integer;  
begin  
  writeln('Geben Sie zwei natürliche Zahlen ein: ');  
  readln(a);  
  readln(b);  
  if a>b then  
    begin  
      b:=a+b;  
      a:=b-a;  
      b:=b-a  
    end;  
  while a<b do  
    begin  
      a:=a+1;  
      b:=b-1  
    end;  
  if a=b then  
    writeln('Ergebnis: ',b)  
  else  
    writeln('Ergebnis: ',b, '.5')  
end.
```

Überlegen Sie sich was das Programm leistet und wie es dabei vorgeht.

- a) Was gibt das Programm für die Eingaben  $a=7$  und  $b=2$  aus? 'Ergebnis: 4.5'  
Was gibt das Programm für die Eingaben  $a=3$  und  $b=5$  aus? 'Ergebnis: 4'
- b) Ergänzen Sie im Programm einen erklärenden Kommentar an der grau eingefärbten Stelle und schreiben Sie eine passende Problemspezifikation:

Eingabe:  $a$  und  $b \in \mathbb{N}$

Ausgabe:  $c \in \mathbb{N}/2$

Nachbedingung:  $c = (a+b) / 2$

## Musterlösung

### Aufgabe 2

```
program Dreieck(input,output);  
{Liest eine natürliche Zahl n ein und gibt ein Dreieck aus 1+...+n X-  
Symbolen aus.}  
  var  
    n:integer;  
    i:integer;  
    j:integer;  
begin  
  writeln('Geben Sie eine natürliche Zahl ein: ');  
  readln(n);  
  for i:=1 to n do  
    begin  
      for j:=1 to i do  
        write('X');  
      writeln()  
    end  
end.  
end.
```

## Musterlösung

### Aufgabe 3

```
function FeldSAbweichung(inFeld:tFeld):real;  
{Berechnet die Standardabweichung der Werte im Feld inFeld.}  
  var  
    Mittelwert:real;  
    SAbweichung:real;  
    i:tIndex;  
begin  
  {Mittelwert berechnen}  
  Mittelwert:=0;  
  for i:=1 to FELDGROESSE do  
    Mittelwert:=Mittelwert+inFeld[i];  
  Mittelwert:=Mittelwert/FELDGROESSE;  
  {Standardabweichung berechnen}  
  SAbweichung:=0;  
  for i:=1 to FELDGROESSE do  
    SAbweichung:=SAbweichung+sqr(inFeld[i]-Mittelwert);  
  SAbweichung:=SAbweichung/FELDGROESSE;  
  FeldSAbweichung:=sqrt(SAbweichung)  
end;
```

## Musterlösung

### Aufgabe 4

```
procedure DVListenElementEntfernen (inWert:integer;  
                                     var ioListe:tRefDVElement);  
{Löscht das erste Element mit Wert inWert aus der DVListe ioListe.}  
var  
  Zeiger:tRefDVElement;  
  gefunden:boolean;  
begin  
  if ioListe<>nil then  
    {Die Liste ist nicht leer.}  
    if ioListe^.Wert=inWert then  
      {Der Suchwert ist das erste Element.}  
      begin  
        Zeiger:=ioListe;  
        ioListe:=ioListe^.next;  
        dispose(Zeiger);  
        if ioListe <> nil  
          ioListe^.prev:=nil  
      end  
    else  
      {Der Suchwert ist nicht das erste Element.}  
      begin  
        Zeiger:=ioListe^.next;  
        gefunden:=false;  
        while (Zeiger<>nil) and not gefunden do  
          if Zeiger^.Wert=inWert then  
            begin  
              Zeiger^.prev^.next:=Zeiger^.next;  
              if Zeiger^.next<>nil then  
                {Es wird nicht das letzte Element gelöscht.}  
                Zeiger^.next^.prev:=Zeiger^.prev;  
                gefunden:=true;  
                dispose(Zeiger)  
            end  
          else  
            Zeiger:=Zeiger^.next  
          end  
      end  
end;  
end;
```

## Musterlösung

### Aufgabe 5

```
function AddiereBinBaum(inBaum:tRefBinBaum):integer;  
{Addiert die Werte aller Knoten des BBAumes inBaum.}  
begin  
  if inBaum=nil then  
    {Abbruchbedingung: Der Baum ist leer.}  
    AddiereBinBaum:=0  
  else  
    {Rekursionsfall: Addition des Wertes mit den zwei Summen der  
    Unterbäume.}  
    AddiereBinBaum:=inBaum^.Wert+AddiereBinBaum(inBaum^.links)  
    +AddiereBinBaum(inBaum^.rechts)  
end;
```

## Musterlösung

### Aufgabe 6

Eine für einen Black-Box-Test sinnvolle Zerlegung der Menge der zulässigen Eingabedaten ist zum Beispiel:

Klasse 1: Alle Paare  $(a,b)$  mit  $a < b$ .

Klasse 2: Alle Paare  $(a,b)$  mit  $a = b$ .

Klasse 3.1: Alle Paare  $(a,b)$  mit  $a > b$  und  $b|a$ .

Klasse 3.2: Alle Paare  $(a,b)$  mit  $a > b$  und  $\neg(b|a)$ .

Testdatum für Klasse1:  $(2,5,2)$

Testdatum für Klasse2:  $(5,5,0)$

Testdatum für Klasse3.1:  $(15,5,0)$

Testdatum für Klasse3.2:  $(17,5,2)$