Pseudocode, Programmablaufplan, Nassi-Schneidermann-Diagramm (Wiederholung)

Wenn man ein Programm entwickelt, ist es oft hilfreich, den Algorithmus erst einmal in einer umgangssprachlichen Form aufzuschreiben und ihn gegebenenfalls grafisch darzustellen. Die Formulierung in Pseudocode macht aus meiner Sicht auch wirklich Sinn, da ein Algorithmus auf diese Art programmiersprachenunabhängig beschrieben werden kann und jeder Programmierer ihn dann in seiner Sprache implementieren kann.

Die grafische Darstellung als **Programmablaufplan** (PaP) bzw. als **Nassi-Schneidermann-Diagramm** wird heute kaum noch verwendet, wird aber von den Vorgaben des Lehrplans gefordert.

Pseudocode

Beim Pseudocode formuliert man den Algorithmus wie oben beschrieben in deutsch, ohne auf irgendeine Programmiersprache Bezug zu nehmen. Alle Formulierungen im Pseudocode sollten sich aber später leicht programmieren lassen.

Wir werden das Ganze an einem kleinen Programm zur Berechnung der Lösungen einer quadratischen Gleichung durchspielen.

Wie ihr wisst, löst man eine quadratische Gleichung der Form $x^2+px+q=0$ mittels der pq-Formel $x_{1,2}= \frac{-p}{2} \mp \sqrt{\left(\frac{p}{2}\right)^2 - q}$. Ein solches Programm werden wir nun schrittweise entwickeln.

Strategie

Zuerst müssen wir p und q einlesen. Sollte die Diskriminante (das ist der Teil unter der Wurzel) kleiner als 0 sein, lässt sich die Wurzel nicht ausrechnen und es gibt keine Lösung. Ist er gleich Null, gibt es eine Lösung. Ist er größer Null, gibt es zwei Lösungen.

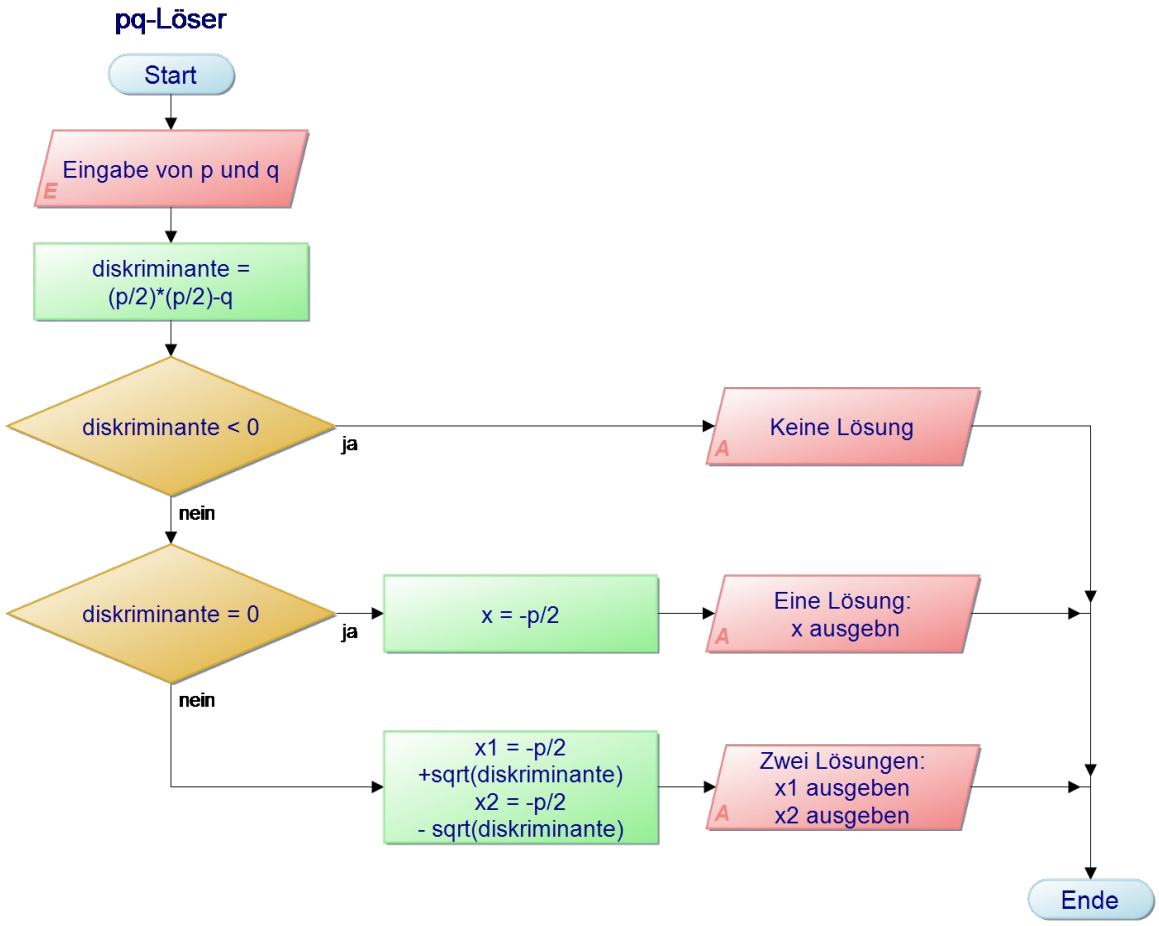
Pseudocode

```
lese p ein
lese q ein

berechne diskriminante = p²/4 - q

falls diskriminamte < 0
  gib aus 'keine Lösung'
sonst
  falls diskriminante = 0
    berechne x = - p/2
    gib aus 'es gibt eine Lösung, sie lautet' x
  sonst
    berechne x1 = - p/2 + sqrt(diskriminante)
    berechne x2 = - p/2 - sqrt(diskriminante)
    gib aus 'es gibt zwei Lösungen' x1 und x2
```

Hier ist bereits alles direkt in z.B. Java programmierbar. Der Pseudocode hat aber mit Java nichts zu tun und könnte so auch in C oder Pascal programmiert werden.

Programmablaufplan (PaP)

Wenn man möchte, kann man einen solchen Algorithmus auch in einen Programmablaufplan überführen. Hier sieht man das Beispiel, welches die Lösungen der quadratischen Gleichung berechnet.



In einem solchen Programmablaufplan werden:

- der Start und das Ende in **ovalen Symbolen** dargestellt,
- Anweisungen in **Rechtecken** gezeichnet,
- Fallunterscheidungen werden als **Raute** dargestellt,
- Ein- und Ausgabe sind **Parallelogramme**.
- Bei Schleifen zeigt ein Pfeil wieder an einen Punkt weiter oben.

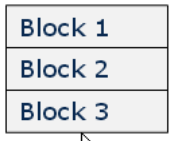
Eine weitere bekannte Darstellungsform von Algorithmen ist das Nassi-Schneidermann-Diagramm.

Nassi-Schneidermann-Diagramme

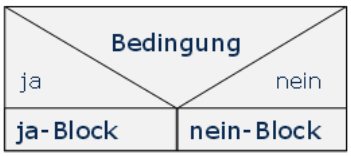
Nassi-Schneidermann-Diagramme wurden früher häufiger zur Visualisierung von kleineren Algorithmen eingesetzt. Die üblichen Bestandteile sind auf der nächsten Seite dargestellt.

Einzelblock

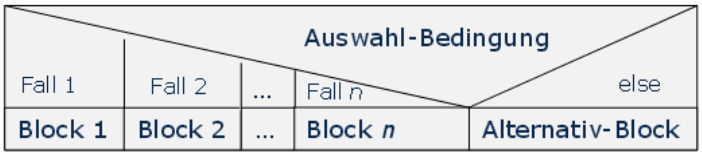
Sequenz: Block enthält eine Folge von Unterblöcken



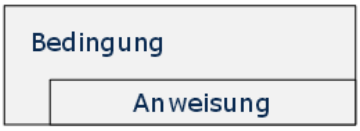
Alternative umfasst zwei Blöcke



Auswahlanweisung umfasst mehrere Blöcke

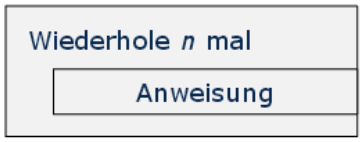


Kopfgesteuerte Schleife (while)



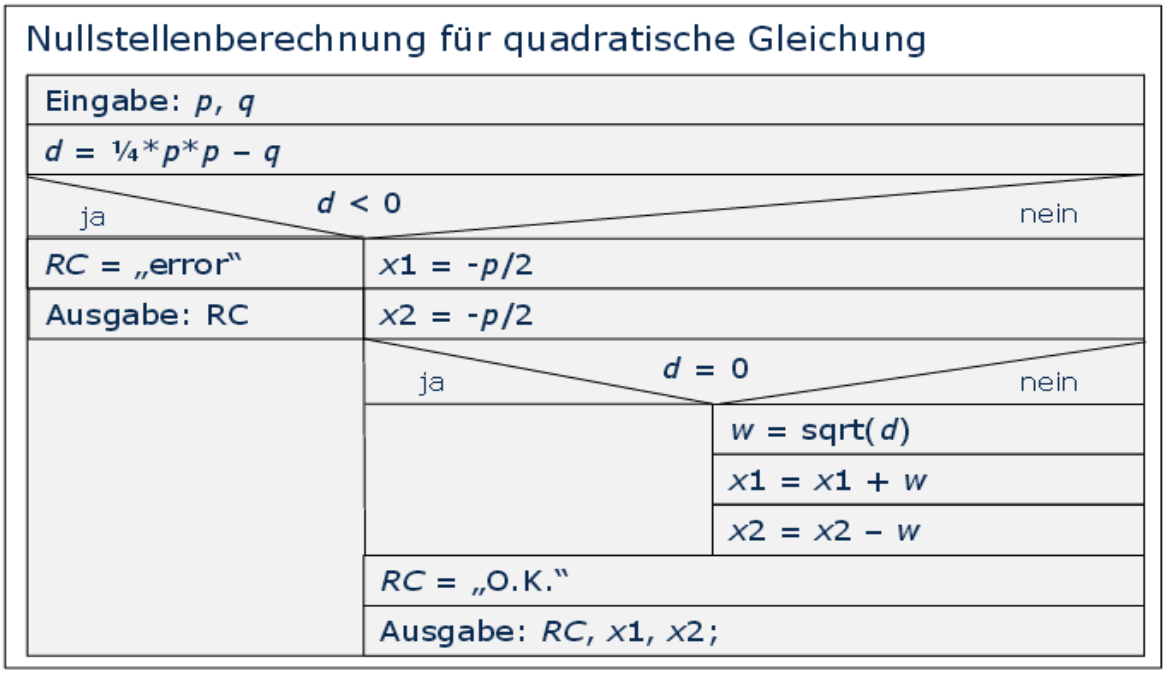
Fußgesteuerte Schleife (do while)

Zählschleife (for)



vorzeitiges Schleifenende

Hier sehen wir nun unseren Algorithmus zur Bestimmung der Lösung der quadratischen Gleichung in einem Nassi-Schneidermann-Diagramm.



Und hier das fertige Java-Programm:

```
 /**
  * Loesungen einer quadratischen Gleichung
  */

public class pqLoeser {

  public static void main(String[] args) {

    System.out.println("Diese Programm löst eine quadratische Gleichung der Form:");
    System.out.println("\n x^2+px+q=0");
    System.out.println();

    double p = IOTools.readDouble("p: ");
    double q = IOTools.readDouble("q: ");

    double diskriminante = (p/2)*(p/2)-q;

    if (diskriminante<0) {
      System.out.println("\nDiese quadratische Gleichung hat keine Lösungen!");
    }else{

      if (diskriminante==0) {
        double x = -p/2;

        System.out.println("\nDiese quadratische Gleichung hat eine Lösungen: "+x);
      }else{
        double x1 = -p/2+Math.sqrt(diskriminante);
        double x2 = -p/2-Math.sqrt(diskriminante);

        System.out.println("\nDiese quadratische Gleichung hat zwei Lösungen:\n");
        System.out.println("Lösung 1: "+x1);
        System.out.println("Lösung 2: "+x2);
      }
    }
  }
}
```

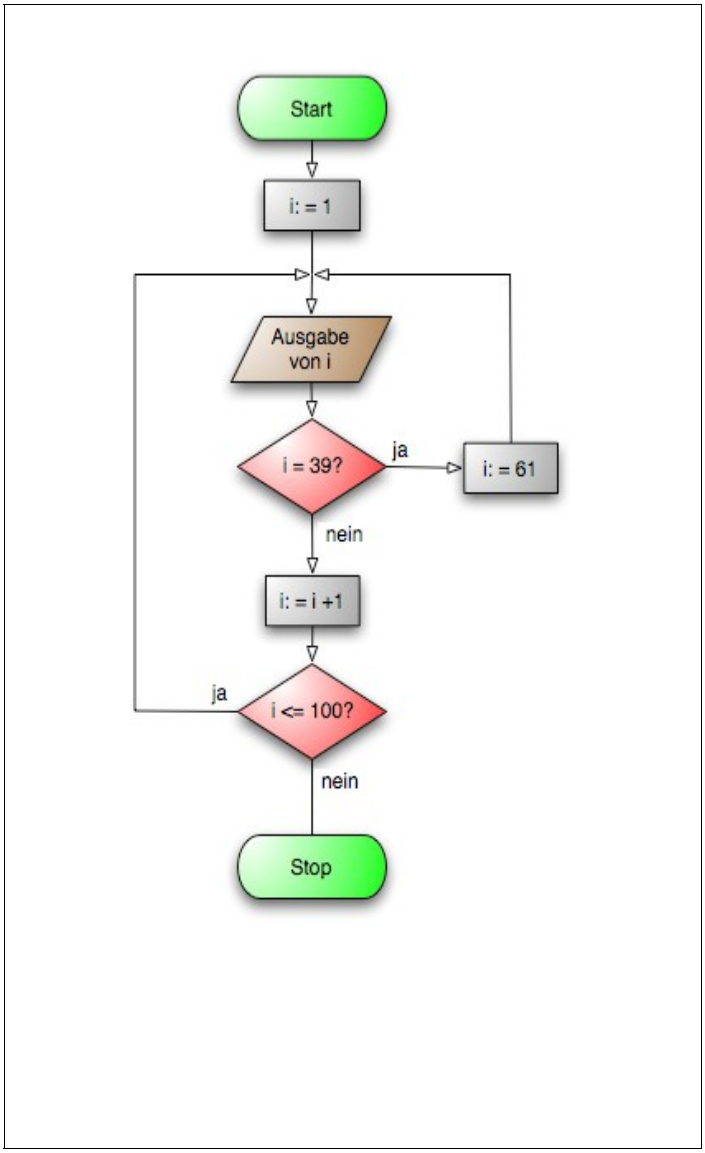
Übungen

Aufgabe 1

a) Überführe den PaP in Pseudocode.

b) Was berechnet der Algorithmus, der in dem nebenstehenden PaP dargestellt ist?

c) Überführe den PaP in der Abbildung in ein Java-Programm und teste deine Annahme aus b).



Aufgabe 2

a) Schreibe den folgenden Pseudocode als PaP.

```
Eingabe a
Eingabe b

falls (a>b)
  solange (a>b)
     a = a-b
   Ausgabe a
sonst
   Ausgabe 0
```

Was wird hier berechnet?

b) Schreibe den folgenden Pseudocode als Nassi-Schneidermann-Diagramm.

```
Eingabe a
summe = 0

für i=1 bis a
  summe = summe + i;

Ausgabe summe
```