technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 1
Dr. Lars Hildebrand
Iman Kamehkhosh, Marcel Preuß, Henning Timm

Praktikum zu

Einführung in die Informatik für LogWings und WiMas

Wintersemester 2013/14

Übungsblatt 7

Bearbeitungszeit:
16.–20.12.2013

Mit Lösungen

Ergänzende Aufgaben

Aufgabe 7.7 – Negative Exponenten

Verbessern Sie das Programm aus der Aufgabe 7.3 so, dass auch negative Exponenten zulässig sind.

```
public static double hoch(int b, int e){
   if(e<0) return (1.0 / hoch(b,-e));
   ... // Rest bleibt so
}
```

Aufgabe 7.8 – Eingabeverarbeitung

In dieser Aufgabe soll eine Eingabe unbekannter Länge verarbeitet werden. Die Eingabe besteht aus einer Folge ganzer Zahlen, die durch die Folge $-1, -1, -1$ abgeschlossen wird, welche sonst nicht vorkommt. Ein Beispiel:

$1, 2, 4, 0, -1, 0, -4, 0, 0, 3, 4, -1, -1, -1$

Diese Werte sollen wie folgt verarbeitet werden:

a) Negative Zahlen werden nicht wieder ausgegeben, also aus der Folge „gelöscht“.

b) In der *verbleibenden* Zahlenfolge wird für jede Gruppe von zwei oder mehr aufeinanderfolgenden Nullen eine einzige Null ausgegeben.

c) Die übrigen Zahlen werden ausgegeben.

Für das Beispiel wäre die korrekte Ausgabe damit:

$1, 2, 4, 0, 3, 4$

Die Eingabe soll Zahl für Zahl durch den Benutzer erfolgen, Sie müssen sich also in ihrem Programm jeweils merken, in welchem Zustand bzgl. der obigen Fälle sie sich jeweils zum Eingabezeitpunkt befinden. Um zu erkennen, ob Sie sich gerade in einer Nullfolge befinden, und auch für das Zählen aufeinanderfolgender Eingaben von -1 können Sie zum Beispiel jeweils eine eigene Variable benutzen.

Lösung:
Wir benötigen ein Integer-Array `AusgabeFeld`, das die wesentlichen Zahlen enthalten soll ; und eine Hilfsmethode `aktualisiereFeld(int[] feld, int eingabe)`, die das Feld um eine Stelle erweitert und am Ende des Arrays die neue Zahl der Eingabe einfügt und dieses als aktualisiertes Feld zurück gibt.
Wir verwenden die Variable `int anzahl_minus_einsen` , um die Anzahl der -1en zu speichern. Es wird eine weitere Variable `zuletzt_geschrieben` benötigt, die dazu verwendet wird, um eine aktuelle 0 zu überspringen, falls die letzte Zahl, die gespeichert wurde eine 0 ist (in Sequenzen wie [0,0,0,..] und [0,-1,0,..]). Das Programm soll eine `do{ ... }while(anzahl_minus_einsen<3);` Schleife verwenden, um die Verarbeitung richtig zu beenden. Die Verarbeitung der `eingabe` hängt von sich selbst, der `letzten` und zuletzt geschriebenen Zahl ab:

zuletzt_geschrieben	letzte	eingabe	Aktion
egal	egal	> 0	aktualisiereFeld
egal	> 0	== 0	aktualisiereFeld
egal	egal	==-1	anzahl_minus_einsen++
egal	egal	<-1	
==0	egal	== 0	
egal	==-1	> 0	aktualisiereFeld anzahl_minus_einsen=0
!=0	==-1	== 0	aktualisiereFeld anzahl_minus_einsen=0
egal	==-1	<-1	anzahl_minus_einsen=0
!=0	<-1	== 0	aktualisiereFeld

Daraus lassen sich Bedingungen für die jeweiligen Aktionen ableiten:

```
if(eingabe>0 || (eingabe==0 && zuletzt_geschrieben!=0){ aktualisiereFeld(feld,eingabe);}
if(letzte==-1 && eingabe!=-1) { anzahl_minus_einsen = 0; }
if(eingabe==-1) { anzahl_minus_einsen++; }
```

```
import java.util.Scanner;

class Eingabeverarbeitung {

   public static int[] aktualisiereFeld(int[] feld, int eingabe) {
      int[] neuesFeld = new int[feld.length+1];
      // alte Zahlen umkopieren
      for(int i=0; i<feld.length; i++) {
         neuesFeld[i] = feld[i];
      }
      // Eingabe am Ende einfügen
      neuesFeld[neuesFeld.length-1] = eingabe;

      return neuesFeld;
   }


   public static void main (String[] args ) {
      Scanner scanner = new Scanner(System.in);

      int[] AusgabeFeld = new int[0]; // Anfangs ist es leer
      int anzahl_minus_einsen = 0;

      // "letzte" muss eine negative Zahl,
      // damit es beim 1. Durchlauf ignoriert wird
      int letzte = -5;
      int eingabe;
      int zuletzt_geschrieben = -2;

      System.out.println("Eingabe :");
      do {
         eingabe = scanner.nextInt();

         if(eingabe>0 || (eingabe==0 && zuletzt_geschrieben!=0)){
            AusgabeFeld = aktualisiereFeld(AusgabeFeld, eingabe);
            zuletzt_geschrieben = eingabe;
         }

         if(letzte==-1 && eingabe!=-1)
            anzahl_minus_einsen = 0;
```

```
            if(eingabe==-1)
               anzahl_minus_einsen++;

            // letzte Zahl auf aktuelle Eingabe
            // für nächsten Durchlauf setzen
            letzte = eingabe;

         } while (anzahl_minus_einsen < 3);

         System.out.println("Ausgabe :");
            for(int i=0; i<AusgabeFeld.length; i++)
               System.out.print(AusgabeFeld[i] + ", ");
      }
   }
```