Informatik 12 mit Java: Dynamische Datenstrukturen (Übung)

Gierhardt

1. Das folgende Programm benutzt Zeiger und macht diverse Modifikationen. Stelle das Geschehen graphisch dar und bestimme damit, welche Ausgaben erfolgen.

```
class Person
{ String vorname, nachname;
  Person naechste;
  Person (String vorn, String nachn, Person next) // Konstruktor
  { this.vorname  = vorn; this.nachname = nachn; this.naechste = next; }
} // Person

class Zeigeruebung
{ static Person anker, alpha, beta, gamma, delta, lauf;

  static void init()
  { alpha = new Person("Jan", "Bangert", null);
    beta  = new Person("Alexandru", "Ienasiga", null);
    gamma = new Person("Lukas", "Janssen", null);
    delta = new Person("Kai-Philipp", "Junke", null);
  } // init

  static void modifikation1()
  { beta = alpha;
    delta = beta;
    delta.nachname = "Konieczka";
    gamma.nachname = alpha.nachname;
  } // modifikation1

  static void modifikation2()
  { anker = beta;
    alpha.naechste = gamma;
    delta.naechste = alpha;
    beta.naechste  = delta;
  } // modifikation2

  static void modifikation3()
  { gamma.naechste = alpha.naechste;
    alpha.naechste = gamma.naechste;
    delta.naechste = anker;
    anker.naechste = alpha;
    anker          = delta;
  } // modifikation3

  static void ausgabe1()
  { Out.println("Folgende Namen sind gespeichert:");
    Out.println(alpha.vorname + " " + alpha.nachname);
    Out.println(beta.vorname  + " " + beta.nachname);
    Out.println(gamma.vorname + " " + gamma.nachname);
    Out.println(delta.vorname + " " + delta.nachname);
  } // ausgabe1

  static void ausgabe2()
  { Out.println("Folgende Namen sind in der Liste gespeichert:");
```

```
    lauf = anker;
    while (lauf != null)
    { Out.println(lauf.vorname + " " + lauf.nachname);
      lauf = lauf.naechste;
    } // while
  } // ausgabe2

  public static void main(String args[]) // Hauptprogramm
  { init();           ausgabe1();
    modifikation1();  ausgabe1();
    init();
    modifikation2();  ausgabe2();
    modifikation3();  ausgabe2();
  } // Ende von main
} // Ende von class Zeigeruebung
```

2. Sehr häufig kommt es vor, dass ein Programm eine Textdatei einlesen und intern verarbeiten muss. Da die Größe der Textdatei unbekannt ist, muss ein Programm je nach Größe entsprechend Speicher anfordern.

   Vereinfacht dargestellt kann eine in ein Programm (z.B. in einen Editor) eingelesene Textdatei folgendermaßen betrachtet werden:

   - Eine Textdatei besteht aus *Zeilen*.
   - Jede Zeile besteht aus ihrem Inhalt und einem Verweis auf die nächste und die vorhergehende Zeile.
   - Es gibt eine erste und eine letzte Zeile.
   - Es gibt eine aktuelle Zeile.

   Entwirf eine geeignete Datenstruktur und schreibe ein Java-Programm, das sich selbst im Quelltext auf dem Bildschirm darstellt (eine Art Mini-Editor).

   Version 1: Die Datei soll eingelesen, in eine entsprechende Datenstruktur eingebracht und auf dem Bildschirm ausgegeben werden.

   Version 2: Zu Beginn sei die erste Zeile die aktuelle Zeile und soll ausgegeben werden. Nun soll durch Eingabe von Zahlen eine bestimmte Zeile relativ zur aktuellen Zeile ausgegeben werden: Eingabe von 2 liefert die übernächste Zeile zur aktuellen Zeile, -1 die vorherige Zeile.