

Hinweise für den Prüfling

Einlese- und Auswahlzeit: 30 Minuten

Bearbeitungszeit: 180 Minuten

Auswahlverfahren

Es gibt zwei Aufgabengruppen A und C, aus denen jeweils ein Vorschlag zu bearbeiten ist. Aus der **Gruppe A (Objektorientierte Modellierung)** hat Ihre Prüferin / Ihr Prüfer einen Vorschlag für Sie ausgewählt.

Wählen Sie von den zwei vorliegenden Vorschlägen der **Gruppe C (Konzepte und Anwendungen der theoretischen Informatik)** einen zur Bearbeitung aus. Der nicht ausgewählte Vorschlag muss am Ende der Einlesezeit der Aufsicht führenden Lehrkraft zurückgegeben werden.

Erlaubte Hilfsmittel

ein Wörterbuch zur deutschen Rechtschreibung

Sonstige Hinweise

ohne PC-Nutzung
Java-Variante

In jedem Fall vom Prüfling auszufüllen

Name: _____ Vorname: _____

Prüferin / Prüfer: _____ Datum: _____

Pfannkuchen sortieren

Pfannkuchen haben alle einen unterschiedlichen Durchmesser. Ein Pfannkuchenstapel soll so sortiert werden, dass die Pfannkuchen der Größe nach geordnet übereinander liegen, wobei der Größte unten liegen soll. Als Hilfsmittel steht lediglich ein Pfannkuchenwender zur Verfügung, mit dem man einen einzelnen Pfannkuchen oder einen Teilstapel von Pfannkuchen wenden kann.

Das Sortieren der Pfannkuchen soll nach folgendem Algorithmus erfolgen: Man sucht im Stapel zunächst den größten Pfannkuchen, schiebt den Wender unter ihn und wendet anschließend den kompletten Teilstapel über dem Wender. Dadurch kommt der größte Pfannkuchen an die oberste Position. Wendet man nun den gesamten Stapel, so gelangt der größte Pfannkuchen ganz nach unten an die gewünschte Position und wird nicht mehr bewegt. Jetzt sucht man den zweitgrößten Pfannkuchen und bringt ihn nach dem gleichen Verfahren an die zweitunterste Position. So fährt man fort, bis der gesamte Stapel sortiert ist.¹

Zur Modellierung dieses Problems wird der Pfannkuchenstapel im Feld *pfannkuchen* dargestellt, wobei in den Feldelementen die Größen der Pfannkuchen gespeichert werden. Das erste Feldelement *pfannkuchen[0]* entspricht dabei dem obersten Pfannkuchen.

Aufgaben

- 1.1 In der Materialvorlage wird das Feld *pfannkuchen* mit Werten vorbelegt. Es soll nach dem beschriebenen Algorithmus sortiert werden. Geben Sie die Reihenfolge der Feldelemente nach jedem Wendevorgang an.
(4 BE)
- 1.2 Analysieren und beschreiben Sie die Methoden *suche_maxIndex(int obergrenze)* und *wenden2(int m, int n)* aus der Materialvorlage.
(17 BE)
- 1.3 Implementieren Sie die Methode *wenden(int n)*, mit der ein Teilstapel der Höhe *n* mit $n \leq \text{Anzahl}$ gewendet wird, d.h. die Reihenfolge der ersten *n* Elemente des Feldes *pfannkuchen* wird umgedreht.
(8 BE)
- 2.1 Stellen Sie den Sortieralgorithmus in einem Struktogramm dar. Verwenden Sie dabei in der Materialvorlage vorkommende Methoden ohne diese selbst detailliert im Struktogramm darzustellen.
(8 BE)
- 2.2 Implementieren Sie die Methode *sortieren()*, welche das Feld *pfannkuchen* gemäß obigem Algorithmus sortiert.
(8 BE)

¹ Geändert nach: Brian Hayes, Pfannkuchen und Gene, Spektrum der Wissenschaft, Oktober 2008, S. 80–86.

3. Eine andere Art das Feld *pfannkuchen* zu sortieren ist die folgende, die allerdings beim Pfannkuchenstapel nicht mehr anwendbar ist: Zunächst wird der kleinste Feldinhalt durch Wenden eines Teilfeldes an die Position 0 des Feldes *pfannkuchen* gebracht. Diese Position wird nicht mehr verändert. Nun kann man wiederum durch Wenden eines Teilfeldes die zweitkleinste Zahl an die zweite Position *pfannkuchen*[1] bringen. So fährt man fort, bis das Feld sortiert ist. Dazu benötigt man eine Methode *wenden2*, die nicht die Positionen 0 bis n von *pfannkuchen* wendet, sondern die Positionen m bis n ($m \leq n$).

Implementieren Sie die Methode *sortieren2()*, mittels derer das Feld *pfannkuchen* gemäß dem 2. Algorithmus sortiert wird.

(9 BE)

4. Beim Backen von Pfannkuchen gelingen selten beide Seiten gleich gut. Jeder Pfannkuchen soll daher eine verbrannte Seite haben, die zufällig nach oben oder unten zeigt. Entwerfen und erläutern Sie einen Algorithmus, der einen Stapel mit einseitig verbrannten Pfannkuchen so sortiert, dass am Ende alle verbrannten Seiten nach unten zeigen.

(6 BE)

Material 1

```
public class PfannkuchenSortieren {

    private int anzahl = 6;
    private int [] pfannkuchen = {12, 18, 22, 15, 19, 13};

    private void wenden(int n) {
        // wendet einen Teilstapel der Höhe n
        // vgl. Aufgabe 1.3
    }

    private int suche_maxIndex(int obergrenze) {
        int maxIndex = 0;
        int max = pfannkuchen[0];
        for (int i = 1; i <= obergrenze; i++) {
            if (pfannkuchen[i] > max) {
                maxIndex = i;
                max = pfannkuchen[i];
            }
        }
        return maxIndex;
    }

    private void sortieren() {
        // vgl. Aufgabe 2.2
    }

    private void wenden2(int m, int n) {
        if (m < n) {
            int hilf = pfannkuchen[m];
            pfannkuchen[m] = pfannkuchen[n];
            pfannkuchen[n] = hilf;
            m++;
            n--;
            wenden2(m, n);
        }
    }

    private int suche_minIndex(int untergrenze) {
        // analog zu suche_maxIndex
    }

    private void sortieren2() {
        // vgl. Aufgabe 3.
    }
}
```