Programmierpraktikum

Wintersemester 2021/22

Prof. Dr. rer. nat. habil. Petra Hofstedt
Ilja Becker M. Sc., Sven Löffler M. Sc.
Sonja Breuß B. Sc., Carlo Bückert B. Sc., Endris Wenzke

Brandenburgische
Technische Universität
Cottbus - Senftenberg

Übungsblatt 12

Abgabedatum: 23.01.2022

Die Abgabe Ihrer Lösungen erfolgt vor Ablauf der Abgabefrist digital über die Moodle-Plattform. Erstellen Sie dazu ein PDF-Dokument, das die Lösungen Ihrer schriftlichen Aufgaben enthält. Laden Sie dieses PDF-Dokument und den erarbeiteten Java-Code (.java-Dateien) mit den in den Aufgaben vorgegebenen Namen bei Moodle hoch. Bitte laden Sie die Dateien einzeln hoch, Dateiarchive (z.B. .zip-Dateien) werden nicht akzeptiert.

Sie können maximal **(9 Punkte)** mit diesem Übungsblatt erreichen.

Aufgabe 1 (Doppelt verkettete Liste)

5 Punkte

Wir betrachten die folgende Interface-Definition für einen Listen-Datentyp:

```
/**
* Ein Interface zur Beschreibung einer Liste, die mit Index 0 beginnt.
*/
public interface List<T> {
  public void add(T obj);
  public void insert(T obj, int index);
  public T get(int index);
  public void delete(int index);
  public int indexOf(T obj);
  public int length();
}
```

Die aufgeführten Methoden haben die folgenden Bedeutungen:

- Die Methode `add(Object obj)` fügt das Objekt obj an das Ende der Liste an.
- Die Methode `insert(Object obj,int index)` fügt das Objekt `obj` vor das an `index`-ter Stelle stehende Objekt in die Liste ein. Falls `index` gleich 0 ist, wird das Objekt vor dem ersten Objekt der Liste (dem mit Index 0), d. h. an den Anfang der Liste eingefügt. Entsprechend wird, falls `index` gleich der Länge der Liste ist, das Objekt an das Ende der Liste angefügt.
- Die Methode `get(int index)` liefert das an Indexposition `index` stehende Objekt der Liste zurück.
- Die Methode `delete(int index)` löscht das an Indexposition `index` stehende Objekt der Liste.
- Die Methode `indexOf(T obj)` liefert die Indexposition des Objekts `obj` in der Liste zurück. Dazu werden die Objekte mit der `equals`-Methode der Listenobjekte auf Gleicheit getestet.
- Die Methode `length()` liefert die Länge der Liste zurück.

Beachten Sie, dass die Länge der Liste im Unterschied zu Arrays nicht bereits bei der Erzeugung der Liste feststehen muss, sondern jederzeit variabel ist.

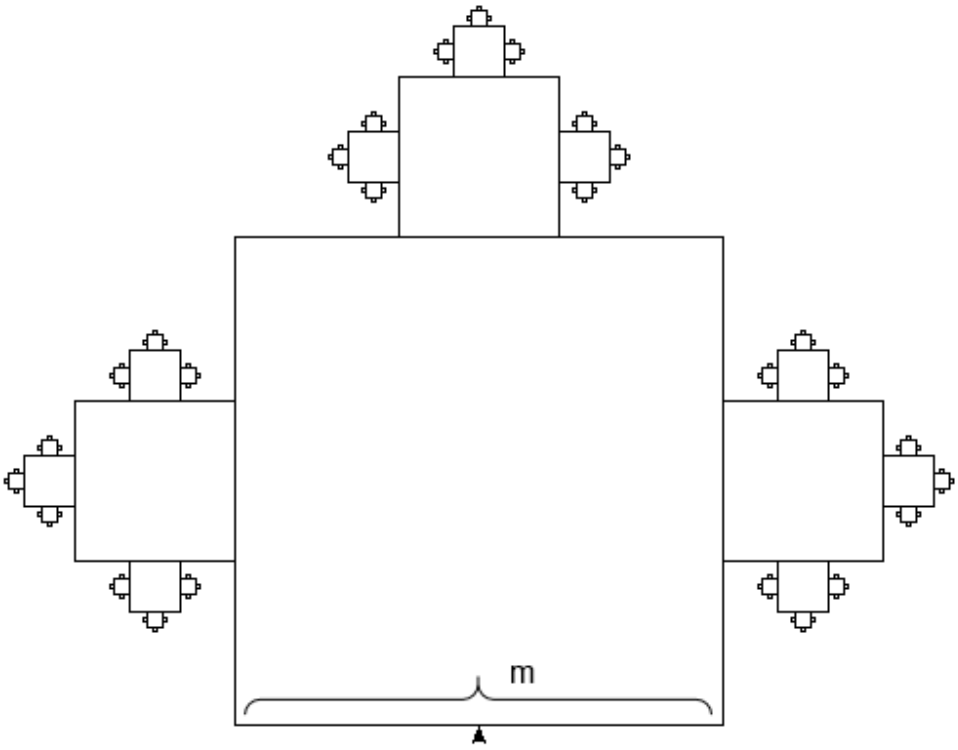
1. Programmieren Sie eine Java-Klasse `DoubleLinkedList<T>`, die dieses Interface als doppelt verkettete Liste implementiert. Schreiben Sie dazu eine innere Klasse `Node<T>` die über ein Element T, einen Vorgänger und einen Nachfolger verfügt.
   Machen Sie sich vor der Übung darüber gedanken, wie die innere Klasse genutzt werden kann und welche Attribute die Klasse `DoubleLinkedList<T>` mindestens enthalten sollte.
2. Überschreiben Sie in der Klasse `DoubleLinkedList<T>` desweiteren die beiden Methoden:

   ```
   public String toString()
   public boolean equals(Object obj)
   ```

3. Schreiben Sie eine `Main.java`-Klasse, die alle Methoden von `DoubleLinkedList<T>` testet. Erzeugen Sie dazu zunächst eine Liste und führen Sie einige Methodenaufrufe auf dieser Liste aus. Durch Ausgabe der Liste vor und nach dem Methodenaufruf kann dann die korrekte Arbeitsweise manuell überprüft werden.
4. Denken Sie bei Ihren Implementierungen an den DAU (dümmsten anzunehmenden User). Das Bedeutet, verhindern Sie, dass das Programm abstürzt, wenn fehlerhafte Eingaben erfolgen.
5. Halten Sie sich an die Code Conventions!

Aufgabe 2 (Quadratpflanze) 4 Punkte

Betrachten Sie die in der Abbildung gezeigte Quadratpflanze.



Im Jahr 0 besteht die Quadratpflanze nur aus dem großen Basisquadrat. Jedes Jahr wachsen dann neue Quadrate mit $m_{neu} = 1/3 * m_{alt}$ an drei Quadratseiten. Die Grundseite des Basisquadrates sei m

Einheiten (z. B. cm) lang.

Legen Sie eine Klasse *Quadratpflanze.java* an, in der Sie die folgenden Methoden implementieren.

1. Die Funktion $flaecheninhalt(n, m)$ beschreibe den Flächeninhalt der gesamten Quadratpflanze nach n Jahren bei einer Seitenlänge m. Entwicklen Sie einen rekursiven Algorithmus zur Berechnung der Methode.

   Für $flaecheninhalt(1, 1)$ ergibt sich z.B. der Wert 1,3333 und für $flaecheninhalt(4, 1)$ der Wert 1.4938.

   (1 Punkt)

2. Die Funktion $umfang(n, m)$ beschreibe den Umfang der gesamten Quadratpflanze nach n Jahren bei einer Seitenlänge m. Entwicklen Sie einen rekursiven Algorithmus zur Berechnung der Methode. Beachten Sie, dass nur die äußeren Linien zum Umfang gehören, nicht aber die inneren.

   Für $umfang(1, 1)$ ergibt sich z.B. der Wert 6 und für $flaecheninhalt(4, 1)$ der Wert 12.

   (1 Punkt)

3. Die Funktion $zeichne(n, m)$ zeichnet die Quadratpflanze nach n Jahren bei einer Seitenlänge m. Entwicklen Sie einen rekursiven Algorithmus zur Berechnung der Methode. Nutzen Sie zum Zeichnen zum Beispiel die *Turtle*-Grafik aus vorherigen Aufgaben oder eine beliebige andere grafische Umsetzung.

   Schreiben Sie eine *main*-Methode, die von der Console die Werte n und m einliest und die 3 Methoden aufruft.

   Welche Beobachtung kann man für die Entwicklung des Umfangs und der Fläche für große n machen?

   (2 Punkt)