

## Übungsblatt 12

Abgabe bis Montag, 24.07.2017, 12:00 Uhr

### Hinweis:

Aufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Bei Programmieraufgaben Java Quellcode und evtl. benötigte Datendateien).

### Aufgabe 12.1

- Was sind verkettete Listen?
- In welcher Komplexitätsklasse liegt der Zugriff auf ein Element in einer verketteten Liste?
- Was sind die Unterschiede zu einer ArrayList?
- In welcher Reihenfolge wird ein Pre-order Binärbaum durchlaufen?
- Was bedeutet statische Finitheit und dynamische Finitheit?

### Aufgabe 12.2

- Implementieren Sie folgende Klassen:

```
public class Pizza {
    public static void main(String[] args) {
        Margherita p1 = new Salami();
        Margherita p2 = new Veggy();

        String[] toppings = new String[3];
        toppings[0] = "Salami";
        toppings[1] = "Pepper";
        toppings[2] = "Hot Pepper";
        Margherita p3 = new GenericPizza(toppings);

        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);

        p1.makeLarger();
        p2.addTopping("Mushroom");

        System.out.println(p1);
        System.out.println(p2);
    }
}

class Margherita {
```

```

public Margherita() {
    name = "Margherita";
    price = 5.0;
    toppings = new String[2];
    toppings[0] = "Tomato";
    toppings[1] = "Cheese";
}

public void makeLarger() { ... }
public double costs() { ... }
public void addTopping(String topping) { ... }
public String toString() { ... }

protected String name;
protected double price;
protected double size;
protected String[] toppings;
}

class Salami extends Margherita{
    public Salami() { ... }
}

class Veggy extends Margherita{
    public Veggy() { ... }
}

class GenericPizza extends Margherita {
    public GenericPizza(String[] toppings) { ... }
}

```

2. Eine Margherita besteht nur aus den Belägen Tomate und Käse. Desweiteren besitzt die Klasse eine Funktion `makeLarger()`, die die Größe erhöht und den Preis um 1 Euro steigert. Die Kostenfunktion gibt die derzeitigen Kosten der Pizza zurück und `addTopping` erweitert die Pizza um eine Zutat (+50 Cent). `toString` soll sowohl die Beläge als auch die Kosten der Pizza zurückgeben.
3. Die Klassen `Salami` und `Veggy` erweitern die Klasse `Margherita` um die entsprechenden Zutaten (jede weitere Zutat kostet 50 Cent zusätzlich zu den Grundkosten der Margherita).
4. Die Klasse `GenericPizza` ermöglicht Pizzen mit jeder denkbaren Kombination von Belägen (zusätzlich zu den Grundzutaten der Margherita).
5. Testen Sie Ihr Program mit der bereitgestellten `main` Methode. Ergibt das Speichern der Beläge in einem Array Sinn oder gibt es bessere Datenstrukturen für diese Aufgabe? Wäre eine andere Vererbungsreihenfolge sinnvoller als die Vorge-schlagene?

### Aufgabe 12.3

Implementieren Sie einen Binärbaum wie dieser in der Vorlesung vorgestellt wurde. Folgende Funktionalitäten soll der Baum besitzen:

```

class Node {
    public Node(double content) { ... }

    public void insertElement(double content) { ... }
}

```

```
public bool isInTree(double content) { ... }  
  
private double content;  
private Node left;  
private Node right;  
}
```

insertElement soll ein Node mit dem Inhalt content einfügen, sofern das Element noch nicht vorhanden ist. Rekursive Lösungen bieten sich hierfür an.