

Übungsblatt 12

Abgabe bis Montag, 18.07.2016, 23:59 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

Das pascalsche Dreieck ist eine Form der grafischen Darstellung der Binomialkoeffizienten. Sie sind im Dreieck derart angeordnet, dass jeder Eintrag die Summe der zwei darüberstehenden Einträge ist:

$n = 0:$					1				
$n = 1:$				1		1			
$n = 2:$			1		2		1		
$n = 3:$		1		3		3		1	
$n = 4:$	1		4		6		4		1

Schreiben Sie eine rekursive Java-Methode, die eine Zeile des pascalschen Dreiecks berechnet. Eine Zeile des Dreiecks soll hierbei als Array von int Werten zurückgegeben werden.

```
public class Pascal {  
    Pascal () {}  
    public int[] pascal(int row) {  
        \\....  
    }  
}
```

Aufgabe 12.2

1. Implementieren folgende Klassen und testen sie diese.

```
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 String[] toppings;
}

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

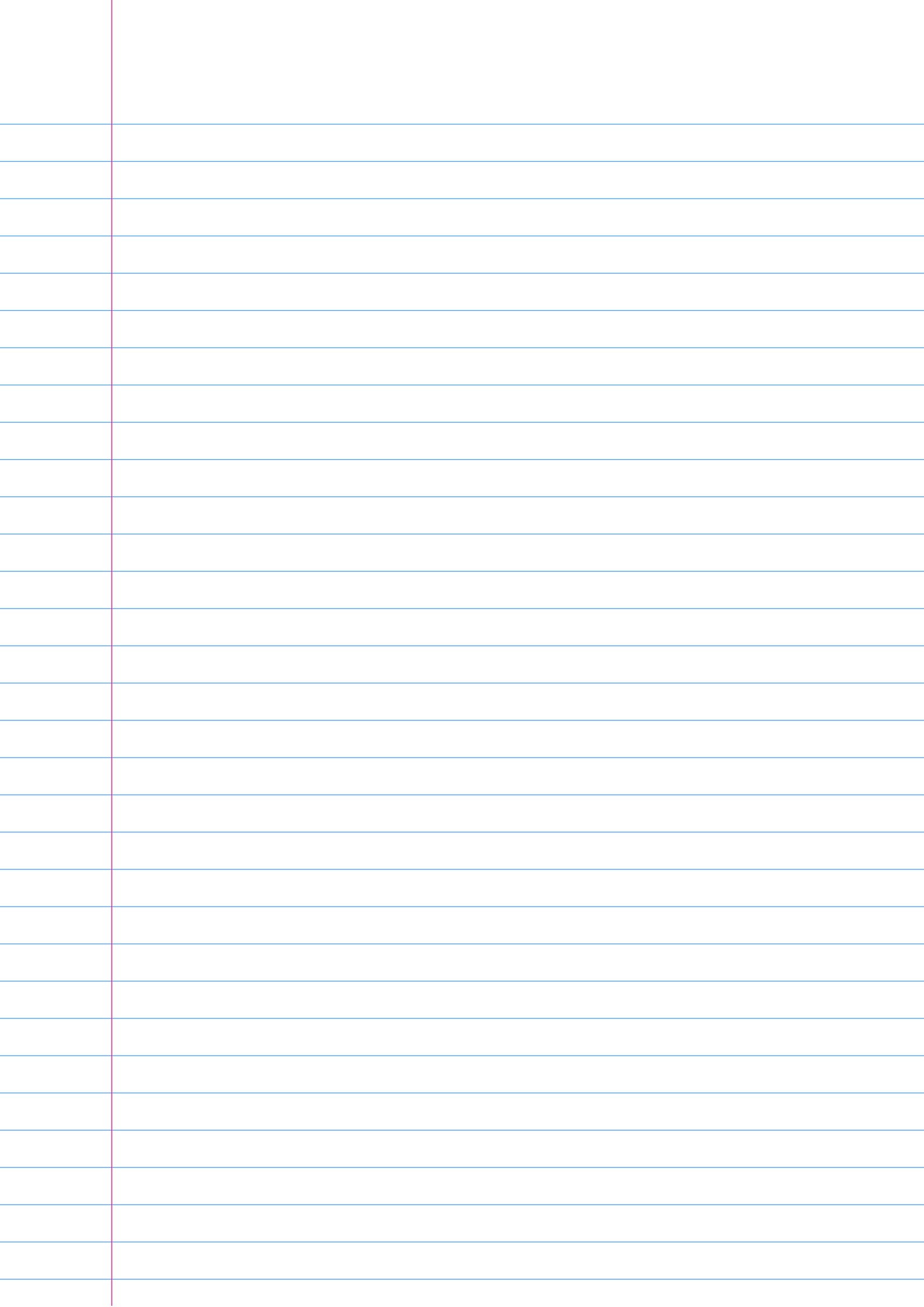
class Veggy extends Margherita{
    public Veggy() { ... }
}

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, die die Größe und den Preis um 1 Euro erhöht. Die Kosten Funktion gibt die derzeitigen Kosten der Pizza zurück und addTopping erweitert die Pizza um eine Zutat. 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 (Preis inbegriffen, jede weiter 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. Macht das Speichern der Beläge in einem Array Sinn oder gibt es bessere Datenstrukturen für

diese Aufgabe? Wäre eine andere Vererbungsreihenfolge sinnvoller als die Vorgeschlagene?



```

public static int[] pascal(int zeile) {
    int[] erg = new int[zeile + 1];
    for (int spalte = 0; spalte <= zeile; spalte++) {
        erg[spalte] = help(zeile, spalte);
    }
    return erg;
}

public static int help(int zeile, int spalte) {
    if (spalte == 0 || spalte == zeile) {
        return 1;
    }
    return help(zeile - 1, spalte) + help(zeile - 1, spalte - 1);
}

```

