

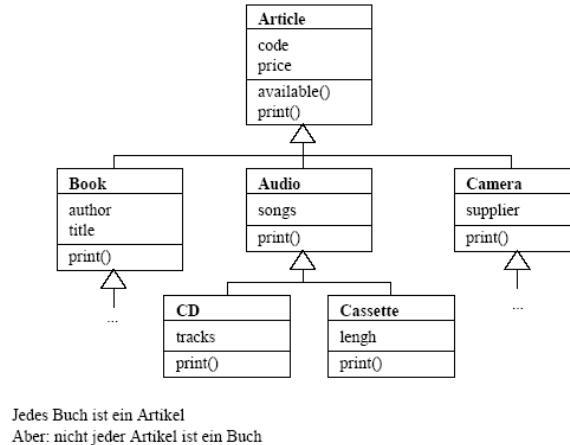
Programmieren und Software-Engineering II

Übung 17

Name: _____ Klasse: _____ Datum: _____

Aufgabe 1: Artikel

Gegeben sei die folgende Klassenhierarchie:



Ergänze die einzelnen Klassen um einige sinnvolle Felder.

Der Ablauf sollte folgendermaßen funktionieren:

- Im HP wird ein Array von Artikeln angelegt.
- Dann wird eine Artikelliste ausgegeben.
- Anschließend werden alle Artikel aufsteigend nach dem Kaufpreis sortiert und wieder ausgegeben.
- In einem weiteren Schritt werden alle Artikel sortiert nach Kategorien ausgegeben.
- Zuletzt wird folgendermaßen sortiert: Die Kategorie mit dem größten Umsatz wird an erster Stelle ausgegeben (hier wiederum der teuerste Artikel zuerst), usw.

Das Hauptprogramm besitzt folgenden Aufbau:

```
public static void main(String[] args){
    ...
    Article[] ar=new Article[10];
    ar[0]=new Book("London","Wolfsblut",1,12);
    ar[1]=new CD(new String[]{"stille nacht","ihr kinderlein kommet","oh du fröhliche"},2,18);
    ar[2]=new CD(new String[]{"Angel","Ghost","All about that bass","Blau blüht der Enzian"},3,28);
    ar[3]=new Book("Zweig","Schachnovelle",4,17);
    ar[4]=new Book("Goethe","Faust",5,23);
    ar[5]=new Camera("Canon",6,255);
    ar[6]=new Camera("Sony",7,499);
    ar[7]=new Book("Brown","Inferno",8,25);
    ar[8]=new Book("Krüger","Java",9,55);
    ar[9]=new Camera("Kodak",10,199);

    // Ausgabe
    System.out.println("Normale Ausgabe");
    printList(ar);
    System.out.println();

    // Sortieren nach Kaufpreis und Ausgabe
    System.out.println("Sortiert nach Kaufpreis");
    sortByPrice(ar);
    printList (ar);
    System.out.println();

    // Sortieren nach Artikelkategorie und Ausgabe
    System.out.println("Sortiert nach Artikelkategorie");
    sortByCategory(ar);
    printList (ar);
    System.out.println();

    // Sortieren nach nach Kategorie mit dem größten Umsatz (hier wiederum sortieren),
    // dann Kategorie mit dem zweitgrößten Umsatz, usw...
    ...
}
```

Programmieren und Software-Engineering II

Übung 17

Aufgabe 2: Erweiterung HÜ 16

Verändere deine Lösung von Beispiel 16 dahingehend, dass im Hauptprogramm nur ein Array von Personen angelegt wird. Berechne wieder, welche Gruppe mehr Geld benötigt.

Erweitere das Beispiel dahingehend, dass folgende Funktionalitäten noch möglich sind:

- Sortieren eines Personenarrays: verwende dazu eine compareTo-Methode, die beide Kindklassen implementieren. Es sollen dabei immer Angestellte vor Arbeitern sortiert werden. Innerhalb der einzelnen Gruppen wird nach Name sortiert.

Beispiel compareTo:

Klasse Worker:

```
public int compareTo(Person other){
    if (other instanceof Worker){
        return (this.name.compareTo(other.name));
    }
    return 1;
}
```