

Praktikum zu
**Einführung in die Informatik für
LogWings und WiMas**
Wintersemester 2013/14

Übungsblatt 11

Bearbeitungszeit:

27.–31.01.2014

Mit Lösungen

Ergänzende Aufgaben

Aufgabe 11.4 – Adressbuch – Objektorientierung all-in-one

Firmen benötigen für Testzwecke häufig große Datensätze für ihre Datenbanken. Ihre Aufgabe ist es, ein Programm zu schreiben, welches ein Adressbuch generiert, in dem eine konstante Anzahl an zufällig generierten Personendaten mit Vorname, Nachname, sowie Adresse (Straße, Hausnummer, Postleitzahl, Ort) erfasst werden. Die Personen werden in zwei Gruppen (Mitarbeiter und Kunde) eingeteilt. Ein Mitarbeiter hat ein Gehalt, wohingegen ein Kunde eine Kundennummer.

Erstellen Sie 4 Dateien, in denen Sie jeweils einen Klassenrumpf mit entsprechenden Namen vorbereiten:

- Anwendung.java
- Person.java
- Adresse.java
- Adressbuch.java

Modellieren Sie die Klassen entsprechend folgender Beschreibung:

a) **Person:**

Die Klasse `Person` enthält folgende *private* Attribute:

- Zeichenkette Vorname
- Zeichenkette Nachname
- Markierung, die angibt, ob es sich dabei um einen Kunden handelt
- Ganzzahligen Wert für eine Kundennummer
- Fließkommazahl für Gehalt, falls es sich dabei um einen Mitarbeiter handelt
- Adresse vom Typ `Adresse`

Die Klasse `Person` soll folgende Methoden unterstützen:

- Erzeugen einer Person als Mitarbeiter, der ein übergebenes Gehalt zugewiesen wird; oder als Kunde, der eine übergebene Kundennummer zugewiesen wird. Dabei wird eine vorhandene Adresse mit übergeben.
- Ausgabe der Personendaten inklusive der Adressdaten. Handelt es sich dabei um einen Mitarbeiter, so soll sein Gehalt mit ausgegeben werden; entsprechend die Kundennummer eines Kunden.
- Abfrage, ob es sich bei der Person um einen Mitarbeiter oder Kunden handelt
- Abfrage des Vornamen
- Abfrage des Nachnamen

b) **Adresse:**

Die Klasse *Adresse* enthält folgende *private* Attribute:

- Zeichenkette Strasse
- Ganzzahligen Hausnummer
- Ganzzahligen Postleitzahl
- Zeichenkette Ort

Die Klasse *Adresse* soll folgende Methoden unterstützen:

- Erzeugen einer Adresse mit übergebenen Werten.
- Ausgabe der Adressdaten.

c) **Adressbuch:**

Die Klasse *Adressbuch* enthält folgende *private* Attribute:

- Ganzzahlige Anzahl an Datensätzen
- Array, in dem Kontakte vom Typ *Person* gespeichert werden
- Ganzzahliger Wert für die Größe des Arrays

Die Klasse *Adressbuch* soll folgende Methoden unterstützen:

- Erzeugen eines leeren Adressbuches mit fester Größe.
- Einfügen einer neuen Person in das Adressbuch. Die Anzahl an Datensätzen wird dabei um 1 erhöht. Wurde vorher ein Kontakt aus dem Adressbuch entfernt, sodass eine Lücke entstanden ist, so soll die neue Person in diese Lücke eingefügt werden. Wird die Größe des Adressbuches dabei überschritten, so soll die neue Person nicht eingefügt werden.
- Entfernen einer Person aus dem Adressbuch anhand des Vor- und Nachnamen. Die Anzahl der Kontakte soll dabei um 1 verringert werden.
- Ausgabe aller Personen, die Mitarbeiter sind.
- Ausgabe aller Personen, die Kunden sind.
- Abfrage einer Person anhand der Vor- und Nachnamen. Dabei soll ein Objekt vom Typ *Person* zurückgegeben werden, oder *null*, falls kein passender Kontakt gefunden wurde.

Nachdem unsere Datenstrukturen modelliert wurden, können wir diese in der Klasse *Anwendung* testen. Fügen Sie der Klasse *Anwendung* folgende Methode zur Generierung der Personen hinzu:

```
public static Person generierePerson(){
    String[] Vornamen =
        {"Alexander", "Anna", "Alina", "Christian", "Nicole",
         "Florian", "Mia", "Lisa", "Sebastian", "Vanessa",
         "Daniel", "Oliver", "Max", "Katharina"};
    String[] Nachnamen =
        {"Müller", "Schmidt", "Mustermann", "Becker",
         "Fischer", "Schneider", "Böhm", "Hartmann"};
    String[] Strassen =
        {"Feldweg", "Bürgerweg", "Klammstraße", "Am Feld",
         "Nordstraße", "Bahnhofsallee", "Beim Brandt"};
    String[] Orte =
        {"Dortmund", "Bochum", "Essen", "Hagen", "Düsseldorf",
         "Köln", "Berlin", "Hamburg", "München", "Aachen"};

    Person neuePerson = null;
    java.util.Random rd = new java.util.Random();
    int vn=rd.nextInt(14);
    int nn=rd.nextInt(8);
    int st=rd.nextInt(7);
```

```

int or=rd.nextInt(10);
int hn=rd.nextInt(99);
int plz=0;
while(plz<10101) plz=rd.nextInt(99999);
Adresse adr = new Adresse(Strassen[st],hn,plz,Orte[or]);
int x=0;
while(x<1200) x=rd.nextInt(5000);
if(rd.nextInt(10)%2==0){
    neuePerson = new Person(Vornamen[vn],Nachnamen[nn],adr,x*1.02);
}else{
    neuePerson = new Person(Vornamen[vn],Nachnamen[nn],adr,x);
}
return neuePerson;
}

```

Fügen Sie der Klasse Anwendung eine statische main-Methode hinzu, die Ihr Adressbuch testet. Erstellen Sie hierzu ein neues Adressbuch fester Größe. Generieren Sie mithilfe der obigen Methode einige Personendaten und fügen Sie diese in das Adressbuch ein. Geben Sie alle Kunden aus. Suchen Sie sich einen davon heraus und entfernen Sie diesen aus dem Adressbuch. Suchen Sie im Adressbuch nach dem entfernten Kontakt.

Anwendung.java

```

class Anwendung{

    public static Person generierePerson(){
        // siehe oben
    }

    public static void main(String[] args){
        Adressbuch meinAdressbuch = new Adressbuch(15);

        for(int i=0; i<25; i++){
            Person p = generierePerson();
            meinAdressbuch.einfügen(p);
        }

        System.out.println("*** Kunden: ***");
        meinAdressbuch.gibKundenAus();

        Person p = meinAdressbuch.suche("Max","Mustermann");
        if(p!=null){
            System.out.println("*** Entferne: Max Mustermann ***");
            meinAdressbuch.entfernen("Max","Mustermann");
        }else{
            System.out.println("Max Mustermann ist nicht vorhanden.");
        }
    }
}

```

Adresse.java

```

class Adresse{
    private String Strasse;
    private int Hausnummer;
    private int PLZ; private String Ort;

    public Adresse(String str, int hausnr, int plz, String ort){
        Strasse = str;
        Hausnummer = hausnr;
        PLZ = plz; Ort = ort;
    }
}

```

```

public void gibAus(){
    System.out.println(Strasse+" "+Hausnummer);
    System.out.println(PLZ+" "+Ort);
}
}

```

Person.java

```

class Person{
    private String Vorname;
    private String Nachname;
    private boolean Kunde;
    private double Gehalt;
    private int Kundennr;
    private Adresse adresse;

    public Person(String Vn, String Nn, Adresse adr, double gehalt){
        // Mitarbeiter
        this.Vorname = Vn;
        this.Nachname = Nn;
        this.adresse = adr;
        Kunde = false;
        Gehalt = gehalt;
        Kundennr = 0;
    }

    public Person(String Vn, String Nn, Adresse adr, int kundennr){
        // Kunde
        this.Vorname = Vn;
        this.Nachname = Nn;
        this.adresse = adr;
        Kunde = true;
        Gehalt = 0;
        Kundennr = kundennr;
    }

    public void gibAus(){
        System.out.println("Vorname: "+Vorname);
        System.out.println("Nachname: "+Nachname);
        this.adresse.gibAus();
        if(Kunde){
            System.out.println("Kundennummer: "+Kundennr);
        }else{
            System.out.println("Gehalt: "+Gehalt);
        }
        System.out.println();
    }

    public boolean istKunde(){
        return Kunde;
    }

    public String gibVorname(){
        return this.Vorname;
    }

    public String gibNachname(){
        return this.Nachname;
    }
}

```

```
class Adressbuch{
    private int Anzahl; private int Größe;
    private Person[] Kontakte;

    public Adressbuch(int größe){
        this.Größe = größe;
        this.Anzahl = 0;
        this.Kontakte = new Person[größe];
    }

    public void einfügen(Person p){
        for(int i=0; i<Größe; i++){
            if(Kontakte[i]==null){
                Kontakte[i] = p;
                Anzahl++;
                break;
            }
        }
    }

    public void entfernen(String vorname, String nachname){
        Person p = suche(vorname, nachname);
        if(p != null){
            p=null;
            Anzahl--;
        }
    }

    public Person suche(String vorname, String nachname){
        Person p=null;
        for(int i=0; i<Größe;i++){
            if(Kontakte[i]!=null){
                if(Kontakte[i].gibVorname()==vorname &&
                    Kontakte[i].gibNachname()==nachname){
                    p = Kontakte[i];
                    break;
                }
            }
        }
        return p;
    }

    public void gibMitarbeiterAus(){
        for(int i=0; i<Größe; i++){
            if(Kontakte[i].istKunde()){
                continue;
            }
            Kontakte[i].gibAus();
        }
    }

    public void gibKundenAus(){
        for(int i=0; i<Größe; i++){
            if(Kontakte[i].istKunde()){
                Kontakte[i].gibAus();
            }
        }
    }
}
```