

Mitarbeiterverwaltung

Einführung in die Objektorientierung

1 Aufgabe

Erstelle eine Klasse *Department*, diese Klasse repräsentiert eine Abteilung einer Firma.

Folgende Eigenschaften muss eine Abteilung aufweisen:

- name
- id
- country
- city

Es soll möglich sein eine "default"-Abteilung erstellen zu können, diese hat die Attribute:

- name = "Test"
- id = 123
- country = "Neverland"
- city = "Gotham"

2 Aufgabe

Erstelle eine Klasse *Employee*, diese Klasse repräsentiert einen Mitarbeiter in einer Firma.

Folgende Eigenschaften muss ein Mitarbeiter aufweisen:

- firstName
- lastName
- id
- department

Des weiteren soll es möglich sein, einen "default" Mitarbeiter anlegen zu können. Dieser soll standardmäßig wie folgt definiert sein:

- firstName = "Max"
- lastName = "Mustermann"
- id = 123
- Test Abteilung von der *Department* - Klasse

Die Methode *print()* soll folgende Konsolenausgabe, dynamisch für jeden Mitarbeiter liefern:

Der Mitarbeiter 123 Max Mustermann arbeitet in der Abteilung Test in Gotham

Beide Klassen **müssen** die *isEqual*-Methode implementieren. Diese soll auf Basis der ID die Gleichheit zweier Objekte zurückgeben.

3 Aufgabe

Die Klasse *App* ist die Hauptapplikation.

3.1 Grundfunktionen

Implementiere eine *main*-Methode, in der die Funktionen der vorher erstellten Klassen getestet werden können.

Bonus: Die IDs sollen zufällig mit Hilfe einer Methode generiert werden.

3.2 Konsolenanwendung

Programmiere eine Konsolenanwendung, mit der beliebige Mitarbeiter und Departments erzeugt werden können.

Ein Dialog könnte wie folgt aussehen:

Bitte legen Sie eine Abteilung an:

Name **Development**

Land **Deutschland**

Stadt **Berlin**

Bitte legen Sie einen Mitarbeiter an:

Vorname **Hans**

Nachname **Müller**

Sie haben folgenden Mitarbeiter angelegt:

Der Mitarbeiter 668 Hans Müller arbeitet in der Abteilung Development in Berlin

Möchten Sie noch einen Mitarbeiter anlegen?

j für ja und n für beenden

n

Auf Wiedersehen!

Für die Aufgabe steht folgendes Skeleton zur Verfügung:

```
public class App {  
  
    public static void main(String[] args) {  
  
        // your code comes here  
  
    }  
  
    /*  
     * for the bonus task  
     */  
    private static int generateID() {}  
    /*  
     */  
  
    /*  
     * for the console application  
     */  
    private static void application() {}  
    /*  
     */  
  
}
```

Listing 1: App.java

```
public class Employee {  
  
    final Department dept;  
  
    /*  
    *  
    * Your code comes here.  
    * Remember you can overload constructors.  
    *  
    */  
  
    public void print() {  
  
        System.out.println();  
  
    }  
  
    public boolean isEqual(Employee other) {  
  
        return false;  
  
    }  
}
```

Listing 2: Employee.java

```
public class Department {  
  
    /*  
    *  
    * Your code comes here.  
    * Remember you can overload constructors.  
    *  
    */  
  
    public boolean isEqual(Department other) {  
        return false;  
    }  
}
```

Listing 3: Department.java

4 Zusatz

- Alle Mitarbeiter sollen in ein Array gespeichert werden.
- Wurde der Mitarbeiter erfolgreich angelegt, soll er ausgegeben werden.
- Nach beenden des Anlegevorgangs sollen alle angelegten Mitarbeiter ausgegeben werden.
- Ist das Array voll, also kann kein Mitarbeiter mehr eingefügt werden soll das Array verdoppelt werden.
- Während der Erstellung neuer Mitarbeiter soll es die Möglichkeit geben aus den jeweiligen Departments zu wählen.

Hinweise

- Achtung auf *NullPointerExceptions* bei Arrays!
- Mit *.equalsIgnoreCase("")* können Strings verglichen werden. Hier wird nicht auf die Groß-/Kleinschreibung geachtet beim vergleich *A.equals("a")*.
- Eine foreach-Schleife könnte hilfreich sein. Ist eventuell lesbarer.
- Nicht auf das Schlüsselwort *static* vergessen.
- Die IDs sollen bei jedem Mitarbeiter eindeutig sein.

Ein möglicher Bildschirmdialog könnte wie folgt aussehen:

Bitte legen Sie einen Mitarbeiter an:

Vorname Alf

Nachname Tenner

In Welcher Abteilung soll Alf Tenner arbeiten?

Für Development geben Sie d oder m für Management ein, oder eine beliebige andere Taste für eine Test-Abteilung:

m

Sie haben folgenden Mitarbeiter angelegt:

Der Mitarbeiter 508 Alf Tenner arbeitet in der Abteilung Management in Berlin

Möchten Sie noch einen Mitarbeiter anlegen?

x für beenden. Beliebige Taste für weiter

f

Bitte legen Sie einen Mitarbeiter an:

Vorname Karl

Nachname Mayer

In Welcher Abteilung soll Karl Mayer arbeiten?

Für Development geben Sie d oder m für Management ein, oder eine beliebige andere Taste für eine Test-Abteilung:

d

Sie haben folgenden Mitarbeiter angelegt:

Der Mitarbeiter 483 Karl Mayer arbeitet in der Abteilung Development in Bangladesh

Möchten Sie noch einen Mitarbeiter anlegen?

x für beenden. Beliebige Taste für weiter

l

Bitte legen Sie einen Mitarbeiter an:

Vorname Hans

Nachname Mayer

In Welcher Abteilung soll Hans Mayer arbeiten?

Für Development geben Sie d oder m für Management ein, oder eine beliebige andere Taste für eine Test-Abteilung:

Eine Test-Abteilung wurde angelegt

Sie haben folgenden Mitarbeiter angelegt:

Der Mitarbeiter 568 Hans Mayer arbeitet in der Abteilung Test in Gotham

Möchten Sie noch einen Mitarbeiter anlegen?

x für beenden. Beliebige Taste für weiter

x

Folgende Mitarbeiter wurden angelegt

508 Alf Tenner

483 Karl Mayer

568 Hans Mayer

Auf Wiedersehen!