Lösungen

Aufgabe 1:

```
public class Mitarbeiter {

   private static int MAX_ID = 1;

   private int id;
   private String name;


   public Mitarbeiter(String name) {
      id = MAX_ID++;
      this.name = name;
   }

   public int getId() {
      return id;
   }

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String toString() {
      return id + ", " + name;
   }
}
```

Aufgabe 2:

```
public class PersonalVerwaltung {

   private ArrayList<Mitarbeiter> mitarbeiterListe;

   public PersonalVerwaltung() {
      mitarbeiterListe =
         new ArrayList<Mitarbeiter>();
   }
```

```
    public void addMitarbeiter(Mitarbeiter m) {
       mitarbeiterListe.add(m);
    }

    public void removeMitarbeiter(Mitarbeiter m) {
       mitarbeiterListe.remove(m);
    }

    public void listMitarbeiter() {
       System.out.println("\nMitarbeiter");
       for (Mitarbeiter m : mitarbeiterListe) {
          System.out.println(m);
       }
    }
}
```

Aufgabe 3:

Neue Methode der Klasse `Mitarbeiter`*:*

```
public boolean istKleiner(Mitarbeiter m) {
   return name.compareTo(m.name) < 0;
}
```

Neue Methode der Klasse `PersonalVerwaltung`*:*

```
public void sortMitarbeiter() {

   for (int i = 1;i < mitarbeiterListe.size();i++) {
      for (int j = mitarbeiterListe.size() - 1;
           j >= i; j--) {

         if (mitarbeiterListe.get(j)
            .istKleiner(mitarbeiterListe
            .get(j - 1))) {

            Mitarbeiter m =
               mitarbeiterListe.remove(j - 1);
            mitarbeiterListe.add(j, m);
         }
      }
   }
}
```

Aufgabe 4:

```
public abstract class Abrechnung {

   private int periode;
   private Mitarbeiter mitarbeiter;

   public Abrechnung(int periode, Mitarbeiter m) {
      this.periode = periode;
      mitarbeiter = m;
   }

   public int getPeriode() {
      return periode;
   }

   public Mitarbeiter getMitarbeiter() {
      return mitarbeiter;
   }

   public abstract double getVerdienst();

   public String toString() {
      return periode + ", " +
             mitarbeiter.getName() + ", " +
             getVerdienst();
   }
}

public class GehaltsAbrechnung extends Abrechnung {

   private double gehalt;

   public GehaltsAbrechnung(int periode,
                    Mitarbeiter m, double gehalt) {
      super(periode, m);
      this.gehalt = gehalt;
   }

   public double getVerdienst() {
      return gehalt;
   }
}
```

```
public class LohnAbrechnung extends Abrechnung {

   private double stundenLohn;
   private double anzahlStunden;

   public LohnAbrechnung(int periode, Mitarbeiter m,
                  double stundenlohn, int stunden) {
      super(periode, m);
      stundenLohn = stundenlohn;
      anzahlStunden = stunden;
   }

   public double getVerdienst() {
      return stundenLohn * anzahlStunden;
   }
}
```

Aufgabe 5:

Neue Methode der Klasse `PersonalVerwaltung`*:*

```
public void listAbrechnungen(int periode) {
   System.out.println("\nAbrechnungen");
   for (Abrechnung a : abrechnungsListe) {
      if (a.getPeriode() != periode) continue;
      System.out.println(a);
   }
}
```

Aufgabe 6:

Änderungen in der Klasse `Mitarbeiter`:

```
public class Mitarbeiter
   implements Comparable<Mitarbeiter> {

   ...

   public int compareTo(Mitarbeiter m) {
      return name.compareTo(m.name);
   }
}
```

Außerdem wird in `PersonalVeraltung` *durch* `compareTo()` *jetzt die Methode* `istKleiner()` *nicht mehr benötigt und kann daher gelöscht werden.*

Geänderte `sortMitarbeiter()` *Methode in der Klasse* `Personal Verwaltung`:

```
public void sortMitarbeiter() {
   Collections.sort(mitarbeiterListe);
}
```