

A08 MergeSort

```
public void merge(Person[] pers1, Person[] pers2, Person[] result, int start)
```

```
public void sort(Person[] personen, int start, int end)
```

Zuerst wird die Variable mitte mit Null initialisiert. Wenn der Start kleiner als das Ende ist, dann wird die Mitte per Berechnung $(\text{Start} + \text{Ende}) / 2$ ermittelt.

Das Array wird nun von

- dem Start bis zur errechneten Mitte
- und dann von der Stelle Mitte +1 bis zum Ende sortiert.

Nun wird aus den zwei sortierten Teilen jeweils ein Array erstellt und ein merge durchgeführt.

In der Klasse Person:

```
public int compareTo(Person p)
```

```
* Vergleicht zwei Personen miteinander  
* @return <0, wenn a<b || =0, wenn a=b || >0, wenn a>b  
*/  
public int compareTo(Person p) {  
  
    int cmp = this.nachname.compareTo(p.getNachname());  
  
    if (cmp == 0){  
        return this.vorname.compareTo(p.getVorname());  
    }  
  
    return cmp;  
}
```

Ersteres wird pro Personenarray ein counter erstellt. Solange der jeweilige counter kleiner als die Länge des entsprechenden Arrays ist, wird geprüft in welchen Array die früher vorkommenden Namen enthalten sind (compareTo wird hier genutzt, da < 0 bedeutet, dass dies vorher kommt). Dann wird das Resultarray an der Stelle start mit den Daten von dem zutreffenden Personenarray an der Stelle counter befüllt. Folglich wird der start erhöht.

Es gibt noch zwei while-Schleifen, die gegebenenfalls die übrigen Elemente in einfügt. Da der counter bei 0 startet und die Länge um min. 1 versetzt ist, wird hier ergänzt

