

## 1 Schüler

Schreibe eine Klasse `Schueler`:

```
public class Schueler implements Comparable<Schueler> {  
    private String klasse;  
    private String name;  
    private String vorname;  
    private char geschlecht;  
    private LocalDate geboren;  
    private String religion;  
}
```

Zwei `Schueler` gelten als gleich, wenn sie gleiche Klasse, Namen, Vornamen und Geburtsdatum besitzen.

- Implementiere entsprechende Methoden `equals` und `hashCode`
- Implementiere in der Klasse eine natürliche Sortierreihenfolge so, dass aufsteigend nach Klasse, dann Name, Vorname und danach nach Geburtsdatum sortiert wird.

## 2 Schülerverwaltung

Schreibe eine Klasse `SchuelerVerwaltung` welche als Instanzvariable eine `Collection<Schueler>` verwaltet. Bei der Erstellung eines solchen Verwaltungsobjektes wird dem Konstruktor der Namen einer Datei übergeben. Diese Datei wird geöffnet und zeilenweise gelesen. Wenn möglich wird jede Zeile in ein gültiges `Schueler`-Objekt verwandelt und in die interne Collection **sortiert** gespeichert. Etwaige Fehlzeilen werden einfach ignoriert. Außerdem besitzt die Klasse `SchuelerVerwaltung` noch folgende Instanzmethoden:

- `public Set<Schueler> getSchuelerFromKlasse(String klasse)`  
Liefert alle Schüler, welche in die entsprechende Klasse gehen.
- `public Set<Schueler> containsName(String name, boolean komplett)`  
Liefert alle Schüler, in welchen der übergebene `String` enthalten ist oder übereinstimmt (abhängig vom übergebenen `boolean`-Wert).
- `public Set<Schueler> getAllWith(char geschlecht)`  
Liefert alle Schüler, welche das entsprechende Geschlecht haben gereiht nach Klasse, Name, Vorname.
- `public Set<Schueler> getGeborenBis(LocalDate datum, boolean vorNach)`  
Liefert je nach übergebenem `boolean` alle Schüler zurück welche vor dem angegebenen Datum (inklusive) geboren wurden bzw. nach dem angegebenen Datum zur Welt gekommen sind gereiht nach Klasse, Name, Vorname.
- `public Map<String, Integer> getKlassenAnzahl()`  
Liefert eine Map in welcher als Schlüssel die Klassenbezeichnungen und als zugehöriger Wert die Anzahl der Schüler in dieser Klasse gespeichert sind.

- `public Map<String, Map<String, List<String>>> getReligionsZugehoerigkeit()`  
Liefert eine Map in welcher als Schlüssel das Religionsbekenntnis gespeichert ist. Der zugehörige Wert ist selbst wiederum eine Map, welche als Schlüssel die Klassenbezeichnung und als zugehörigen Wert eine Liste aller Namen aus dieser Klasse mit entsprechender Religionszugehörigkeit besitzt.
- `public Map<LocalDate, Set<String>> getGeburtstagsListe(int jahr)`  
Liefert eine Map in welcher die Geburtstage und die zugehörigen Geburtstagskinder nach Datum gereiht sind. Schlüssel ist das Geburtsstagsdatum eines Schülers im übergebenen Jahr. Wert ist eine Liste aller Geburtstagskinder. In dieser Liste ist jeder enthaltene Schüler folgendermaßen textmäßig und entsprechend sortiert erfasst: Name gefolgt von einem Leerzeichen, Vorname(n) gefolgt von einem Leerzeichen, Klasse gefolgt von einem Leerzeichen und um den wievielten Geburtstag es sich handelt.
- `public Map<LocalDate, Set<String>> getGeburtstagsListe()`  
Liefert eine Map in welcher die Geburtstage und die zugehörigen Geburtstagskinder nach Datum gereiht sind. Schlüssel ist das Geburtsstagsdatum eines Schülers im aktuellen Jahr. Wie überladene Methode für das aktuelle Jahr.

Testen Sie ihre Klasse mit den beiliegenden Files.