

# Prüfungsaufgabe

In dieser Abschlusssaufgabe untersuchen Sie Daten zu Inzidenzen und Todesfällen im Zusammenhang mit HIV/AIDS von der Global Health Data Exchange (perma-Link zum Download-Tool mit den Einstellungen, mit denen die csv-Datei für diese Aufgabe generiert wurde) zusammen mit UN-Bevölkerungsdaten (wir verwenden die csv-Datei “Medium variant” unter “Period indicators”).

Kurz zusammengefasst lesen Sie aus der GHDx-csv-Datei Informationen zu HIV-Inzidenz sowie HIV-bezogener Sterblichkeit und aus der UN-csv-Datei Informationen zu Gesamtbevölkerung und Gesamtsterblichkeit über die Zeit hinweg aus. Diese stellen Sie dann im Zusammenhang dar, und Sie berechnen den Anteil der Sterblichkeit, der auf HIV zurückzuführen ist.

## PopulationEntry

Implementieren Sie zunächst zur Darstellung eines Eintrags aus der Bevölkerungsstatistik-Tabelle eine Klasse `PopulationEntry` mit den folgenden Methoden:

- `public PopulationEntry(String country, int fromYear, int toYear, double deathRate):` Constructor
- `public boolean isValidForYear(int year):` Gibt `true` zurück, falls die `PopulationEntry` das Jahr `year` beschreibt (es also zwischen `fromYear` (inklusive) und `toYear` (exklusive) liegt - Vorsicht, die Zahlen in der Datei überlappen sich, `toYear` zählt also nicht dazu, sonst könnten zwei Zeilen das selbe Jahr beschreiben), sonst `false`
- Die folgenden getter:
  - `public String getCountry()`
  - `public int getFromYear()`
  - `public int getToYear()`
  - `public double getDeathRate()`

## HIVEntry

Implementieren Sie dann zur Darstellung eines Eintrags aus der HIV-Fallstatistik eine Klasse `HIVEntry` mit den folgenden Methoden:

- `public HIVEntry(String measure, int year, double value):` Constructor
- Die folgenden getter:
  - `public String getMeasure()`
  - `public int getYear()`
  - `public double getValue()`

## EntryManager

Implementieren Sie anschließend zum Einlesen und Verwalten von Einträgen aus den zwei csv-Dateien eine Klasse `EntryManager` mit den folgenden Methoden:

- `public EntryManager(String populationFilename, String HIVfilename, String country):` Constructor, der:
  - Aus der Datei `populationFilename` für alle Zeilen, die zum Land `country` gehören, `PopulationEntry`-Objekte generiert und speichert.
  - Aus der Datei `HIVfilename` für alle Zeilen, die zum Land `country` gehören und deren `measure` “Deaths” ist sowie deren Metrik “Rate” ist, entsprechende `HIVEntry`-Objekte generiert und speichert.
  - Aus der Datei `HIVfilename` für alle Zeilen, die zum Land `country` gehören und deren `measure` “Prevalence” ist sowie deren Metrik “Rate” ist, entsprechende `HIVEntry`-Objekte generiert und speichert.
- `public PopulationEntry getPopulationEntryForYear(int year):` Gibt das `PopulationEntry` zurück, welches sich auf das Jahr `year` bezieht, zurück. Falls kein solches eingelesen wurde, soll stattdessen `null` zurückgegeben werden.

- `public HIVEntry getHIVDeathrateForYear(int year)`: Gibt das `HIVEntry` für das measure “Deaths” zurück, welches sich auf das Jahr `year` bezieht, zurück. Falls kein solches eingelesen wurde, soll stattdessen `null` zurückgegeben werden.
- `public HIVEntry getHIVPrevalenceForYear(int year)`: Gibt das `HIVEntry` für das measure “Prevalence” zurück, welches sich auf das Jahr `year` bezieht, zurück. Falls kein solches eingelesen wurde, soll stattdessen `null` zurückgegeben werden.
- `public String getCountry()`: Gibt das Land zurück, für welches die Daten eingelesen wurde.

Schauen Sie sich zum Implementieren des Einlesens die csv-Dateien an: Die erste Zeile ist in beiden Fällen ein header, in dem die Spaltennamen stehen.

In der UN-Datei finden Sie den Namen des Landes in der Spalte `Location`, die Jahre auf die sich die Zeile bezieht in der Spalte `Time`, und die death rate in der Spalte `CDR` (“Crude death rate”, das sind die Sterbefälle pro 1000 Personen).

In der GDBx-Datei finden Sie die measure in der Spalte `measure_name`, die Metrik in der Spalte `metric_name`, den Wert in der Spalte `val` und das Jahr, auf das sich die Zeile bezieht, in der Spalte `year`.

Sie können zum Testen - da die vollständigen Dateien sehr groß sind - gerne die Dateien `GDBx_HIVdata_four.csv` sowie `UN_popIndicators_four.csv` verwenden. Diese Dateien enthalten jeweils nur Einträge für 4 Länder/Regionen.

## Analysis

Implementieren Sie nun zum Analysieren der Daten eine Klasse `Analysis` mit den folgenden Methoden:

- `public Analysis(String populationFilename, String HIVfilename, String country)`: Constructor, erstellt und speichert einen `EntryManager` mit den übergebenen Parametern.
- `public double[] getCrudeDeathrates(int fromYear, int toYear)`: Gibt die crude death rates (also das, was `getDeathRate()` von `PopulationEntry` zurückgibt) **in Todesfällen pro 100'000 Personen** von den Jahren zwischen `fromYear` und `toYear` (jeweils inklusive) als `double`-Array zurück. Dabei soll im ersten Element des Arrays die crude death rate vom Jahr `fromYear` stehen, im zweiten Element die vom Jahr `fromYear+1`, und so weiter, bis im letzten Element die vom Jahr `toYear` steht. Sollte es für ein Jahr keine passende `PopulationEntry` geben, soll für dieses Jahr im Array eine 0 eingetragen werden. **Vorsicht:** In der UN-csv-Datei steht die death rate in Todesfällen pro 1'000 Personen, also müssen Sie hier umrechnen!
- `public double[] getHIVDeathrates(int fromYear, int toYear)`: Gibt die HIV-bezogene Todesrate (also das, was in der GDBx-csv-Datei als measure “Deaths” steht) in Todesfällen pro 100'000 Personen von den Jahren zwischen `fromYear` und `toYear` (jeweils inklusive) als `double`-Array zurück. Dabei soll im ersten Element des Arrays die Todesrate vom Jahr `fromYear` stehen, im zweiten Element die vom Jahr `fromYear+1`, und so weiter, bis im letzten Element die vom Jahr `toYear` steht. Sollte es für ein Jahr keine passende `HIVEntry` geben, soll für dieses Jahr im Array eine 0 eingetragen werden. **Vorsicht:** In der GDBx-csv-Datei steht die Raten bereits in Fällen pro 100'000 Personen, Sie brauchen hier also nichts mehr umrechnen!
- `public double[] getHIVPrevalences(int fromYear, int toYear)`: Gibt die HIV-Prävalenz (also das, was in der GDBx-csv-Datei als measure “Prevalence” steht) in Erkrankten pro 100'000 Personen von den Jahren zwischen `fromYear` und `toYear` (jeweils inklusive) als `double`-Array zurück. Dabei soll im ersten Element des Arrays die Prävalenz vom Jahr `fromYear` stehen, im zweiten Element die vom Jahr `fromYear+1`, und so weiter, bis im letzten Element die vom Jahr `toYear` steht. Sollte es für ein Jahr keine passende `HIVEntry` geben, soll für dieses Jahr im Array eine 0 eingetragen werden. **Vorsicht:** In der GDBx-csv-Datei steht die Prävalenzen bereits in Fällen pro 100'000 Personen, Sie brauchen hier also nichts mehr umrechnen!
- `public double[] getPercentageHIVDeathrates(int fromYear, int toYear)`: Gibt den Anteil der Todesrate von den Jahren zwischen `fromYear` und `toYear` (jeweils inklusive), der auf HIV zurückzuführen ist (also die HIV-Todesrate geteilt durch die crude death rate), in Prozent als `double`-Array zurück. Dabei soll im ersten Element des Arrays der Anteil vom Jahr `fromYear` stehen, im zweiten Element der vom Jahr `fromYear+1`, und so weiter, bis im letzten Element der vom Jahr `toYear` steht. Sollte für ein

Jahr die crude death rate 0 sein (oder es keinen passenden `PopulationEntry`-Eintrag geben), soll für dieses Jahr im Array eine 0 eingetragen werden. **Hinweis:** Nutzen Sie die Methoden von `Analysis`, die Sie bereits implementiert haben!

Implementieren Sie dann zum plotten Ihrer Analyseergebnisse ebenfalls in `Analysis` eine Methode `public void plotPercentageAnalysis(int fromYear, int toYear, boolean plotPrevalence, boolean plotCrudeDeathrate, String filename)`, die den prozentualen Anteil von HIV-Todesfällen an der Todesrate (also `getPercentageHIVDeathrates`) sowie optional - je nachdem, ob die jeweiligen `boolean` `true` oder `false` sind - die HIV-Prävalenz sowie die crude death rate plottet und das Ergebnis in der Datei `filename` speichert.

**Hinweis:** Verwenden Sie für die Plots, damit die unterschiedlich großen Werte alle auf einem Plot sichtbar sind, für jede Datenreihe (also Anteil von HIV-Todesrate und Gesamttodesrate, HIV-Prävalenz, Gesamttodesrate) eine eigene Y-Achse. Sie können das tun, indem Sie zunächst mittels `setYAxisGroup` die verfügbaren Y-Achsen definieren. Dabei ist das erste Argument die Nummer der Y-Achse und das 2. Argument ihr Titel:

```
XYChart chart = new XYChart(800, 300);
chart.setYAxisGroupTitle(1, "Titel der ersten Y-Achse");
chart.setYAxisGroupTitle(2, "Titel der zweiten Y-Achse");
```

Wenn Sie dann eine Serie hinzufügen, können Sie ihr mit `setYAxisGroup(int achsenNummer)` eine dieser Y-Achsen zuweisen. Der folgende Beispielcode:

```
XYChart chart = new XYChart(800, 300);
chart.setXAxisTitle("X");
chart.setYAxisGroupTitle(1, "Y-Achse 1");
chart.setYAxisGroupTitle(2, "Y-Achse 2");
double[] x = new double[] {1, 2, 3, 4, 5};
double[] y1 = new double[] {100, 250, 20, 80, 190};
double[] y2 = new double[] {0.01, 0.025, 0.05, 0.08, 0.02};
XYSeries series1 = chart.addSeries("Große Zahlen", x, y1);
series1.setYAxisGroup(1);
XYSeries series2 = chart.addSeries("Kleine Zahlen", x, y2);
series2.setYAxisGroup(2);
new SwingWrapper(chart).displayChart();
```

generiert folgendes Bild:

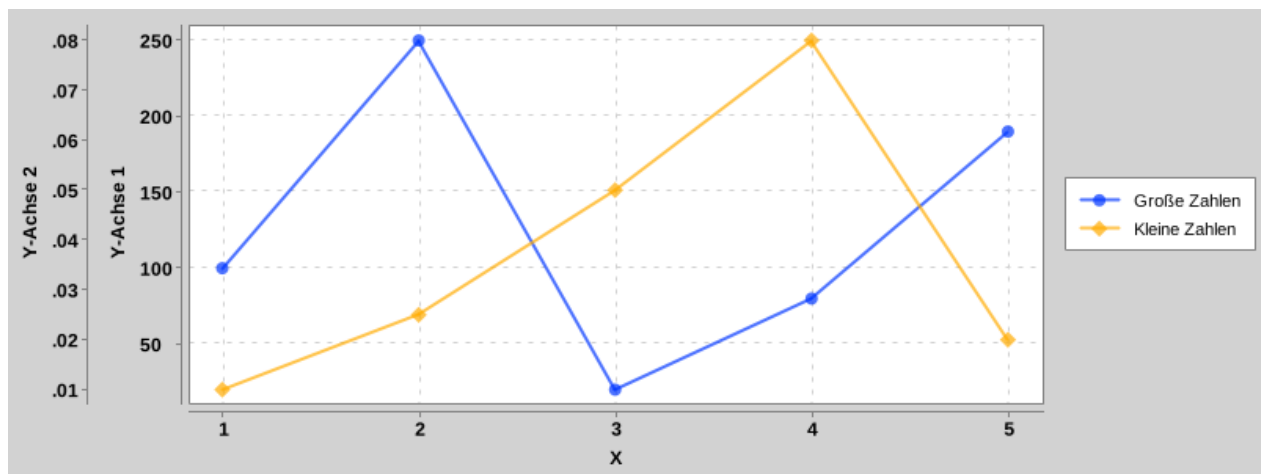


Figure 1: Beispiel Achsengruppen

Generieren Sie mit Ihrer Methode zwei Bilder, die Sie mit Ihrer Lösung hochladen (es gibt bis zu 4 Punkte/Bild - laden Sie die Bilder also auch hoch, wenn sie nicht perfekt sind, es kann trotzdem Punkte dafür geben!).

## Bild 1

Das erste Bild soll den Anteil von HIV an der Todesrate sowie die HIV-Prävalenz für Deutschland in den Jahren 1990-2018 darstellen. Der entsprechende Plot für Indien würde beispielsweise so aussehen:

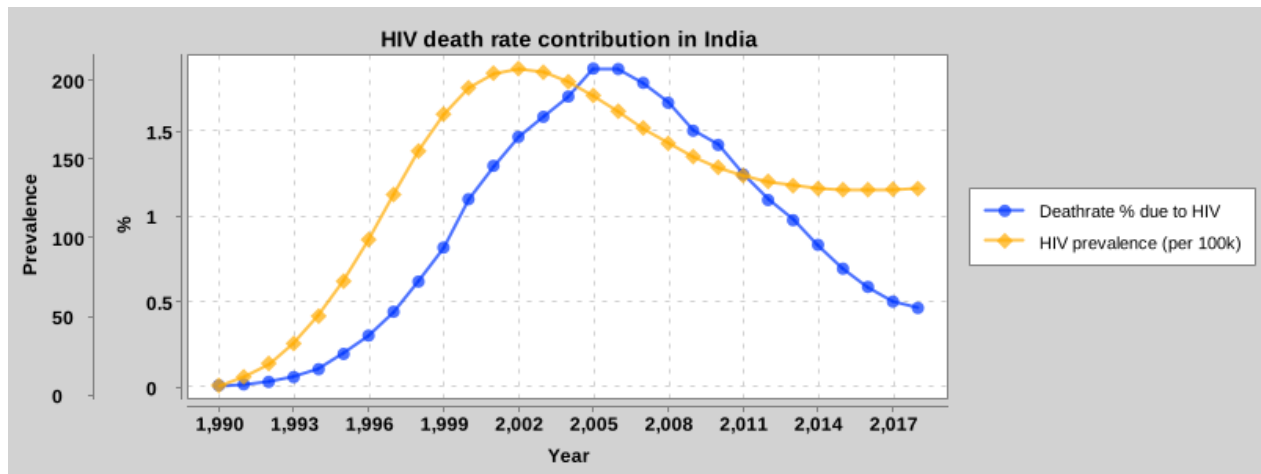


Figure 2: Beispiel Indien

## Bild 2

Erstellen Sie ein zweites Bild, bei dem Sie von irgendeinem anderen Land in irgendeinem beliebigen Zeitraum die Daten (mindestens Anteil von HIV an der Todesrate und HIV-Prävalenz, optional auch die crude death rate) plotten - irgendetwas, was Sie interessant finden.

## Interpretation

Sehen Sie einen Unterschied? Haben Sie eine Idee, woran das liegen könnte (oder woran der Unterschied zwischen Indien in dem Beispiel oben und Deutschland liegen könnte)? Sie können (müssen aber nicht) für 3 Bonuspunkte eine zusätzliche Textdatei mit Ihren Ideen hochladen. Das ist nämlich der Punkt, an dem Gesundheitsinformatik anfängt, Spaß zu machen: Man sieht irgendwelche Effekte und versucht, sie zu erklären.