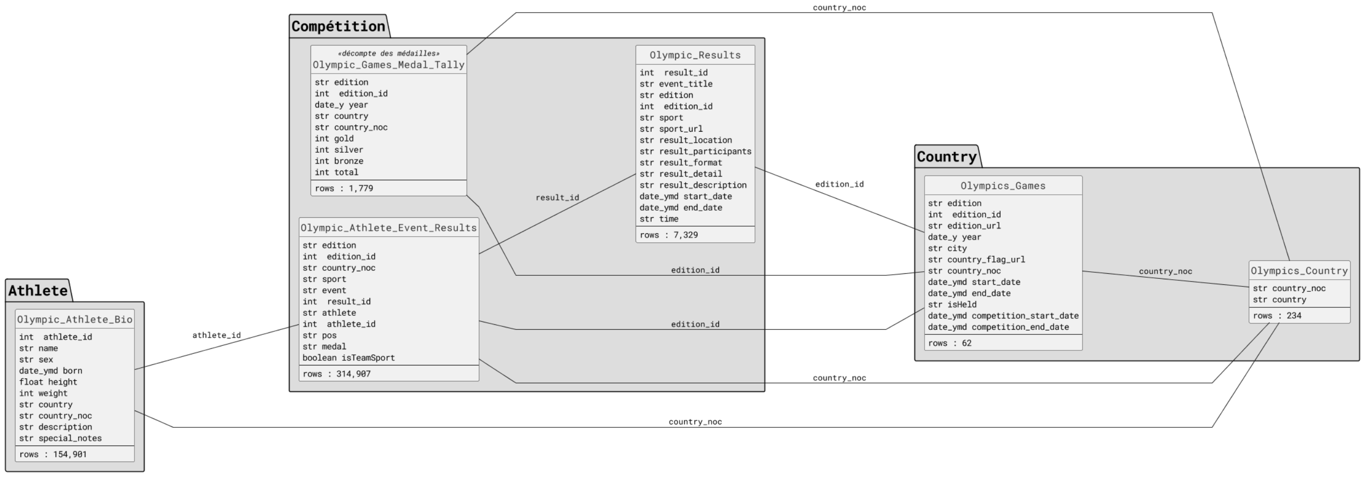
[CS209A-24Fall] Assignment 1 (100 points)

This summer, we've enjoyed the Olympic Games Paris 2024. Many of us are still reliving the exciting moments of the summer Olympics, and many of us may be interested in the event of past Olympics and the past performance of our favorite athletes.

In this assignment, we provide a large dataset of Olympic games from 1896 to 2022. The dataset contains athlete bio and athlete-event data, including both team and individual sports, from all Winter/Summer Olympics (1896-2022). The dataset contains 6 csv, with the following relation:

Update : **2022-08-18**

Olympic historical dataset from olympedia.org



Compétition

```
«décompte des médailles»
Olympic_Games_Medal_Tally
str edition
int  edition_id
date_y year
str country
str country_noc
int gold
int silver
int bronze
int total
rows : 1,779
```

```
Olympic_Results
int  result_id
str event_title
str edition
int  edition_id
str sport
str sport_url
str result_location
str result_participants
str result_format
str result_detail
str result_description
date_ymd start_date
date_ymd end_date
str time
rows : 7,329
```

```
Olympic_Athlete_Event_Results
str edition
int  edition_id
str country_noc
str sport
str event
int  result_id
str athlete
int  athlete_id
str pos
str medal
boolean isTeamSport
rows : 314,907
```

Athlete

```
Olympic_Athlete_Bio
int  athlete_id
str name
str sex
date_ymd born
float height
int weight
str country
str country_noc
str description
str special_notes
rows : 154,901
```

Country

```
Olympics_Games
str edition
int  edition_id
str edition_url
date_y year
str city
str country_flag_url
str country_noc
date_ymd start_date
date_ymd end_date
str isHeld
date_ymd competition_start_date
date_ymd competition_end_date
rows : 62
```

```
Olympics_Country
str country_noc
str country
rows : 234
```

Source : https://bit.ly/3Cifauj
(on Kaggle, thanks **Joseph Cheng**)

The data should be self-explanatory. Please check the column name and the above diagram carefully to understand the data.

Analysis of the Olympic Historical Dataset

In this assignment, you will design an `OlympicsAnalyzer` class, which could read the dataset and perform various useful analyses. You are encouraged to use the Java features we recently introduced, such as `Collections`, `Lambda`, and `Streams`, whenever possible.

The `OlympicsAnalyzer` class has one constructor that reads a dataset file from a given path.

```
public OlympicsAnalyzer(String datasetPath)
```

This class also has 5 other methods that perform data analyses. You'll implement these methods in the `OlympicsAnalyzer` class. Method details are described below.

1. Top 10 Performant Female Athletes in Individual Sport

```
public Map<String, Integer> topPerformantFemale()
```

This method returns a `<name, count>` map of size 10, where:

- the key is the name of the female athlete.
- the value is the total gold medals acquired by that female athlete in individual sport (i.e., not team sport).

The map should be sorted by the **descending order of `count`** (i.e., from most to the least). If two athletes have the same gold medal count, then they should be sorted by the alphabetical order of the name.

2. BMI by Sports

```
public Map<String, Float> bmiBySports()
```

This method returns a `<sport, BMI>` map, where:

- the key is the sport name (e.g., Athletics, Boxing, Diving, etc.)
- the value is the average BMI of the athletes of that sport. BMI is computed as `weight/(height x height)` where the unit of height is the meter. The average BMI should be rounded to one decimal place.

You should consider all athletes with valid height and weight in our data. The athlete should also have participated in any Olympic events so that we know the `sport` type of that athlete.

The map should be sorted by the **descending order of average `BMI`** (i.e., from highest to lowest). If two sports have the same average BMI, then they should be sorted by the alphabetical order of the sport name.

3. Top 10 Least Appeared Summer Sports

```
public Map<String, Set<Integer>> leastAppearedSport()
```

This method returns a `<sport, set of years>` map with a size of 10, where:

- the key is the sport name (e.g., Athletics, Boxing, Diving, etc.)
- the value is set of summer Olympics years in which the sport appears.

The map should be sorted by the **ascending order of the size of the set** (i.e., from least appeared to most appeared). If two sports have the same occurrences, then they should be sorted by the alphabetical order of the sport name.

4. Top 10 Countries by Medals in Winter Olympics since the year 2000

```
public Map<String, Integer> winterMedalsByCountry()
```

This method returns a `<country, count>` map with a size of 10, where:

- the key is the country code.

- the value is the sum of all medals, including gold, silver and bronze, acquired by that country in all Winter Olympics since the year 2000 (including 2000).

The map should be sorted by the **descending order of** `count` (i.e., from most to the least). If two countries have the same medal count, then they should be sorted by the alphabetical order of the country code.

5. Top 10 Countries with Young Athletes in 2020 Summer Olympic

```
public Map<String, Integer> topCountryWithYoungAthletes()
```

This method returns a `<country, age>` map with a size of 10, where:

- the key is the country name (not country code).
- the value is the average age of the athletes of that country that participate in 2020 Summer Olympic. For simplicity, you may ignore the exact month and day and only use the year of birth when computing the age. For example, an athlete born in Dec 31 2000 can be considered as of age 20 in 2020. The average age should be rounding to the nearest integer.
- Athletes with no `born` information should be ignored.

The map should be sorted by the **ascending order of average** `age` (i.e., from youngest to oldest). If two countries have the same average age, then they should be sorted by the alphabetical order of the country name.