

Erstelle ein neues Java-Projekt "MarathonResults2023", in dem die Daten eines Marathon-Rennens (2023) von einer Datei eingelesen und nach bestimmten Kriterien gefiltert werden. Zudem sollen die Ergebnisse in eine neue CSV-Datei geschrieben werden.

Setze dazu folgende Aufgaben um:

1.) Klasse Participant:

- Erstelle eine Klasse Participant mit den Eigenschaften name, gender, category, finishTime und country.
- Implementiere einen Konstruktor, der alle Eigenschaften übernimmt.
- Erstelle Getter- und Setter-Methoden für alle Eigenschaften.
- Überschreibe die toString-Methode so, dass ein Participant-Objekt beispielhaft wie folgt dargestellt wird:

```
John Doe;Male;USA;M30;03:15:42;
```

3.) Factory-Methode in der Klasse Participant:

- Implementiere eine statische Methode `getParticipantFromCSV(String line)` in der Klasse Participant, die eine CSV-Zeile verarbeitet und ein Participant-Objekt zurückgibt.

4.) Klasse FileProcessor:

- Erstelle eine Klasse FileProcessor die auch die main-Methode beinhaltet.

5.) Datei einlesen:

- Entwickle in der Klasse FileProcessor eine Methode `readCSVFile(String filename)`, die eine Liste von Participant-Objekten zurückgibt.
- Verwende einen BufferedReader zum Einlesen der Datei mit Berücksichtigung der Pfadangabe.
- Überspringe die Kopfzeile der Datei.

6.) Daten in Datei schreiben:

- Implementiere in der Klasse FileProcessor eine Methode `writeFile(String filename, List<Participant> participants)`, die die Participant List in eine neue TXT-Datei schreibt.
- Jedes write erweitert den Inhalt der TXT Daten (>> beim FileWriter `append=true` setzen!).
- Trenne jede Ergänzung in der TXT Datei mit einem Datum/Zeit-Stempel als Trennzeile. Die Zeit soll dabei bis inkl. Mikrosekunden genau angeschrieben werden.

7.) Filtern nach einer angegebenen category:

- Implementiere in einer eigenen Klasse **ParticipantStreamer** die static Methode, die über Streams, eine Liste mit Participant Objekten erstellt, die nach dem Übergabeparameter „category“ aus der Liste aller Participant-Objekte die relevanten filtert.
`List<Participant> filterParticipantsByCategory(participants, category);`

8.) Mapping der Daten:

- Implementiere in der Klasse `FileProcessor` eine Methode `Map<String, List<Participant>> buildParticipantMap(List<Participant> participants)`, die eine Liste von Teilnehmern gruppiert nach category zurückgibt:

Key (String)	Value (List<Participant>)
M20	Liste mit Teilnehmern in dieser Kategorie

8.) ParticipantStreamer, Participants filtern und ausgeben:

- Implementiere in der Klasse **ParticipantStreamer** eine static Methode `List<Participant> printParticipantsByCategoryAndCountry(Map<String, List<Participant>> map, String category, String countryCode)`, die eine Kategorie und die Map übergeben bekommt und folgende Liste als Rückgabewert erzeugt – diese Liste der Rückgabe Ausgabe soll dann zusätzlich mit der `writeFile` Methode in die TXT Datei geschrieben werden:

- Die ersten drei Teilnehmer aus der angegebenen Kategorie M20 mit der Nation AUT, sortiert nach finishTime und dann nach name.

Beispielhafte Ausgabe für category „M20“ und countryCode „AUT“:

```
Name;Gender;Country;Category;FinishTime;
Austin Rosario;Male;AUT;M20;02:32:41;
Jeffrey Franco;Male;AUT;M20;02:39:18;
Shawn Hudson;Male;AUT;M20;03:14:12;
```

- Überlege selbst zwei weitere Streaming Methoden in `ParticipantStreamer`:
 - eine einfachere und
 - eine komplexere.

9.) Test der Funktionen:

- Teste alle Methoden in der main-Methode der Klasse `FileProcessor`.
- Stelle sicher, dass die erwartete Ausgabe erzeugt wird und die neue TXT-Datei korrekt erstellt wird.

Am Ende sollte die main-Methode in etwa so aussehen:

```
public static void main(String[] args) throws IOException {
    String category = "M20";
    String country = "AUT";
    List<Participant> participants = readCSVFile(PATH + inputfile);
    List<Participant> participantsByCategory =
        ParticipantStreamer.filterParticipantsByCategory(participants, category);
    Map<String, List<Participant>> participantMap = buildParticipantMap(participants);
    writeFile(PATH + outputfile, participantMap.getOrDefault(category, new ArrayList<>()));
    writeFile(PATH + outputfile, ParticipantStreamer.printParticipantsByCategoryAndCountry(participantMap,
        category, country));
}
```