

Fortgeschrittene Programmierungsmethoden

Praktische Prüfung

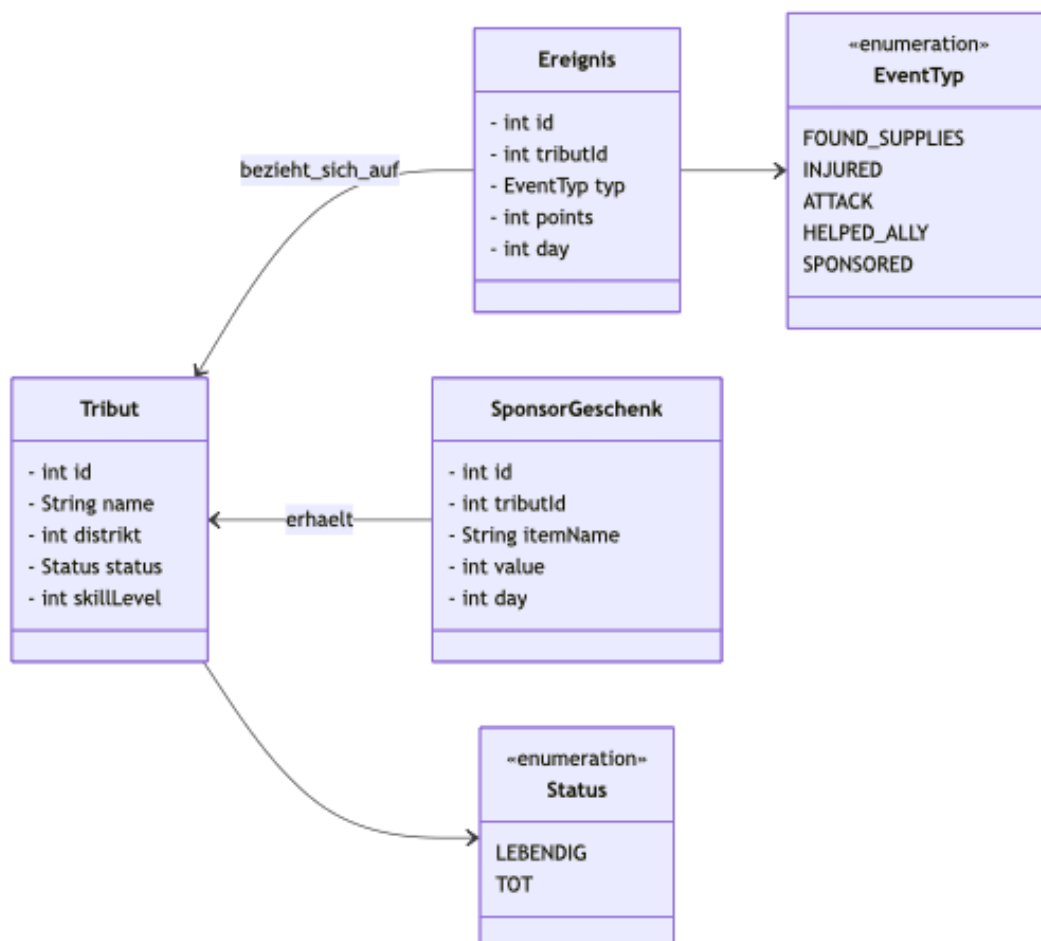
V1

Hunger Games - Arenaregister

Vor Beginn der Hunger Games verwaltet das Kapitol ein **Arenaregister**, in dem alle Tribute, deren Aktionen in der Arena sowie die von Sponsoren erhaltenen Geschenke erfasst werden.

Auf Grundlage dieser Informationen erstellt das Kapitol verschiedene Listen und Berichte, welche den Verlauf der Spiele beeinflussen.

Modellieren Sie in Java das folgende Klassendiagramm und verwenden Sie es in Ihrer Implementierung:



1. (1 Punkt) Lesen Sie die Daten aus den JSON-Dateien `tributes.json`, `events.json` und `gifts.json` und speichern Sie diese in Java-Listen. Geben Sie anschließend auf der Konsole aus:

- Anzahl der Tribute
- Anzahl der Ereignisse
- Anzahl der Sponsorengeschenke
- alle Tribute (jeweils eine Zeile)

Ausgabeformat (Tribut):

`id | name | D<district> | status | skill=<skillLevel>`

Ausgabe:

Tributes loaded: 15

Events loaded: 11

Gifts loaded: 8

```
1 | Katniss Everdeen | D12 | ALIVE | skill=9
2 | Johanna Mason | D7 | ALIVE | skill=8
3 | Foxface | D5 | DEAD | skill=7
4 | Clove | D2 | DEAD | skill=6
5 | Thresh | D11 | ALIVE | skill=8
6 | Rue | D11 | DEAD | skill=5
7 | Glimmer | D1 | DEAD | skill=6
8 | Marvel | D1 | DEAD | skill=7
9 | Cato | D2 | ALIVE | skill=8
10 | Peeta Mellark | D12 | ALIVE | skill=7
11 | Finnick Odair | D4 | ALIVE | skill=9
12 | Annie Cresta | D4 | ALIVE | skill=6
13 | Brutus | D2 | DEAD | skill=8
14 | Wiress | D3 | DEAD | skill=6
15 | Beetee | D3 | ALIVE | skill=7
```

2. (0.5 Punkte) Filtern nach Distrikt und Status

Lesen Sie von der Tastatur eine ganze Zahl **D**, die ein Distrikt repräsentiert. Geben Sie anschließend **nur die Tribute aus**, die **gleichzeitig** folgende Bedingungen erfüllen:

- `district == D`
- `status == ALIVE`

Die Ausgabe erfolgt im selben Format wie in Aufgabe 1.

Ausgabe:

```
Input district: 12
```

```
1 | Katniss Everdeen | D12 | ALIVE | skill=9
10 | Peeta Mellark | D12 | ALIVE | skill=7
```

3. (0.5 Punkte) Sortierung der Tribute

Sortieren Sie die Liste der Tribute wie folgt:

- absteigend nach `skillLevel`
- bei gleichem `skillLevel` aufsteigend nach `name`

Geben Sie die sortierte Liste auf der Konsole aus.

Ausgabe:

```
11 | Finnick Odair | D4 | ALIVE | skill=9
1 | Katniss Everdeen | D12 | ALIVE | skill=9
13 | Brutus | D2 | DEAD | skill=8
9 | Cato | D2 | ALIVE | skill=8
2 | Johanna Mason | D7 | ALIVE | skill=8
5 | Thresh | D11 | ALIVE | skill=8
15 | Beetee | D3 | ALIVE | skill=7
3 | Foxface | D5 | DEAD | skill=7
8 | Marvel | D1 | DEAD | skill=7
10 | Peeta Mellark | D12 | ALIVE | skill=7
12 | Annie Cresta | D4 | ALIVE | skill=6
4 | Clove | D2 | DEAD | skill=6
7 | Glimmer | D1 | DEAD | skill=6
14 | Wiress | D3 | DEAD | skill=6
6 | Rue | D11 | DEAD | skill=5
```

4. (1 Punkt) Schreiben in eine Datei

Schreiben Sie die in Aufgabe 3 sortierte Liste der Tribute in die Datei `tributes_sorted.txt`. Jeder Tribut soll in einer eigenen Zeile gespeichert werden, im selben Format wie bei der Konsolenausgabe.

5. (1.5 Punkte) Punktberechnung

Das Kapitel definiert feste Regeln zur Berechnung der Punkte für jedes Ereignis. Implementieren Sie für alle folgenden `EventTyp`-Werte je eine Regel:

- `FOUND_SUPPLIES` → `computedPoints = points + 2 * day`
- `INJURED` → `computedPoints = points - day`
- `ATTACK` → `computedPoints = points * 2 + day`
- `HELPED_ALLY` → `computedPoints = points + 5`
- `SPONSORED` → `computedPoints = points + 10`

Berechnen Sie für die ersten 5 Ereignisse aus `events.json` die `computedPoints` und geben Sie pro Ereignis eine Zeile aus:

Ausgabeformat:

```
Event      <id>      ->      rawPoints=<points>      ->
computedPoints=<computedPoints>
```

Ausgabe:

```
Event 1 -> rawPoints=10 -> computedPoints=12
Event 2 -> rawPoints=-5 -> computedPoints=-7
Event 3 -> rawPoints=15 -> computedPoints=31
Event 4 -> rawPoints=8 -> computedPoints=13
Event 5 -> rawPoints=12 -> computedPoints=14
```

6. (1.5 Punkte) Ranking

Berechnen Sie für jeden Tribut den Gesamtscore:

```
totalScore = Summe(computedPoints aus den Events des Tributs)
             + Summe(value aus den SponsorGeschenken des Tributs)
```

Geben Sie die Top 5 Tribute aus, sortiert nach

- `totalScore` absteigend
- bei Gleichstand `name` aufsteigend

Ausgabe:

```
Top 5 Tributes:
1. Finnick Odair -> 80
2. Johanna Mason -> 59
3. Cato -> 51
4. Katniss Everdeen -> 35
5. Beetee -> 21
```

7. (1 Punkt) Abschlussbericht

Erstellen Sie die Datei `arena_report.txt`.

Der Bericht soll eine Übersicht über die Ereignisse in der Arena enthalten. Berechnen Sie auf Basis der Datei `events.json` die Anzahl der Ereignisse pro EventType und schreiben Sie das Ergebnis in die Datei.

```
FOUND_SUPPLIES -> 3
INJURED -> 2
ATTACK -> 3
HELPED_ALLY -> 2
```

SPONSORED -> 1

- **2 Punkte** für **Codequalität**
(Einhaltung der Java-Konventionen, klare Trennung in Schichten / MVC, strukturierter und gut lesbarer Code)
- **1 Punkt** wird **von Amts wegen** vergeben