

Aufzählungstypen und Ausnahmen

Dieser Aufgabenzettel muss spätestens in Ihrer Praktikumsgruppe in **KW 26** vorgestellt werden.

Alle für diesen Aufgabenzettel entwickelten Programme sollen sich in einem Paket **de.hsruhrwest.oop.ss2025.assignment8** oder entsprechenden **Unterpaketen** befinden.

Hinweis: Alle Klassen müssen vollständig mit JavaDoc kommentiert sein.

Aufgabe 1: Aufzählungstypen

Implementieren Sie die folgenden Aufzählungstypen (Enums) in Java. Jeder Aufzählungstyp sollte mindestens fünf Ausprägungen besitzen. Hinweis: Die Verwendung von `instanceof` ist in dieser Aufgabe nicht erlaubt.

- a) Ein Aufzählungstyp für Musikrichtungen.
- b) Ein Aufzählungstyp für Städte. Für jedes Element sollen ein lesbarer Name und die Anzahl der Einwohner zurückgegeben werden können.
- c) Ein Aufzählungstyp für Lehrmodule. Für jedes Element sollen eine ID und der Name eines Moduls in lesbarer Form zurückgegeben werden können.

Aufgabe 2: Ausnahmen

- a) Erklären Sie in eigenen Worten den Unterschied zwischen geprüften und ungeprüften Ausnahmen. Wie werden die Ausnahmen jeweils erstellt? Welche Besonderheiten gibt es beim Fangen (catch) der jeweiligen Ausnahmen?
- b) Implementieren Sie einen Datentyp, der eine **Person** mit Namen, Alter und Gender repräsentiert. Gender sollte als einfacher Aufzählungstyp realisiert werden (männlich, weiblich, divers). Die Klasse Person sollte mindestens einen Konstruktor besitzen, der alle Werte setzt, Getter für jedes Attribut und eine `toString`-Methode. Wird versucht, im Konstruktor einen

ungültigen Wert zu setzen, soll eine `IllegalArgumentException` (mit sinnvollem Parameter) geworfen werden.

c) Schreiben Sie eine Klasse **PersonStatistics**. Sie speichert über den Konstruktor eine Liste von Personen (siehe a). Die Klasse soll die folgenden Methoden bieten:

- Eine Methode soll die jüngste Person in der Liste zurückzugeben. Ist die Liste leer, soll eine von Ihnen definierte geprüfte **EmptyListException** geworfen werden. Gibt es mehr als eine mögliche jüngste Person, soll eine von Ihnen definiert geprüfte **NonUniquePersonFoundException** geworfen werden. Diese speichert das jüngste Alter der Liste.
- Schreiben Sie analog eine Methode, die die älteste Person zurückgibt.
- Schreiben Sie eine Methode `printYoungestAndOldestPerson`. Sie soll die bereits entwickelten Methoden benutzen, um die jüngste und älteste Person zu bestimmen und diese jeweils ausgeben. Tritt dabei eine Ausnahme auf, soll diese gefangen werden. Die Methode wirft dann eine von Ihnen definierte geprüfte Ausnahme **CouldNotCreateStatisticException**, die die Ursprungsausnahme (cause) als Parameter besitzt.

d) Testen Sie das korrekte Auftreten aller oben genannten Ausnahmen in einem Hauptprogramm.