

Java-Grundlagentraining

Tag 8

Agenda

- Lambda Expressions
- Aufgabe 1 – Umgang mit Lambdas
- Default Methoden in Interfaces
- Aufgabe 2 – Arbeiten mit Default Methoden in Interfaces
- Stream API
- Aufgabe 3 – Umgang mit Streams
- Abschlussprüfung
- Bonusday Ideen

Lambda Expressions

- Griechischer Buchstabe
- Beschreibt eine Methode, welche nicht über den Namen, sondern über eine Referenz ausgeführt wird.
- Ermöglicht Programmlogik(Methode) als Parameter zu übergeben
- Die zu übergebende Methode muss zu einem Interface gehören
- Dieses Interface darf maximal eine Nicht-Default Methode haben
- Ersetzt anonyme Klassendeklarationen mit nur einer Methode
(z.b. `new Thread(new Runnable(...))`)
- Aufruf: `(Parameter1,Parameter2,..) -> {Programmlogik}`
- IntelliJ schlägt eine automatische Umwandlung vor.

```
//Vor Java 8
new Thread(new Runnable()
{
    @Override
    public void run()
    {
        someMethod();
    }
}).start();

//Seit Java 8
new Thread(() -> someMethod()).start();
```

Aufgabe 1 – Umgang mit Lambda Expressions

1. Erstelle in der main-Methode einen neuen Thread und starte ihn
2. Baue den Aufruf wie von IntelliJ vorgeschlagen in einen Lambda-Ausdruck um
3. Erstelle ein Interface mit einer Methode A, welche einen String Parameter erwartet.
4. Schreibe in einer anderen Klasse eine Methode B, welche dieses Interface und einen String als Parameter erwartet und intern die Parameter an die Methode A des übergebenen Interfaces weiterreicht.
5. Rufe Methode B auf und übergib die Logik, welche den String Parameter von Methode A auf der Konsole ausgibt, als Lambda.
6. Erweitere Methode A um einen zwei weitere 'int' Parameter und passe deinen Aufruf in Methode B dahingehend an, sodass die beiden übergebenen Zahlen einmal addiert ausgegeben werden.
7. Implementiere nach deinem beliebigen weiteren Logik als Lambda.

Default Methoden in Interfaces

- Ermöglichen Programmlogik in einem Interface
- Keyword „default“ am Anfang des Methodenkopfes
- Dadurch muss man nicht mehr zwangsweise alle Methoden eines Interfaces implementieren
- Default Methoden können in jeder Implementierung überschrieben werden.
- Erlaubt es uns nachträglich Methoden in ein Interface aufzunehmen, ohne alle bisherigen Implementierungen anpassen zu müssen !!

Aufgabe 2 – Arbeiten mit Default Methoden in Interfaces

1. Erweitere das Automat-Interface aus deinem Stallprogramm um eine Methode „aufräumen“, um den entsprechenden Automaten selbst nach dem Arbeitsvorgang wieder sauber zu machen.
2. Wandle die Methode zu einer default-Methode um, welche in Abhängigkeit der Anzahl der Stalltiere eine Zeit braucht um sich zu reinigen.
3. Benutze für den FuetterAutomat die default-Methode und überschreibe diese für den SchlachtAutomat, wo das Aufräumen doppelt so lange braucht.

Stream API

- Streams ermöglichen uns ein vereinfachtes arbeiten mit dem Collection-Framework
- Wird durch die default-Methode „stream()“ im Collection-Interface ermöglicht
- Ein Stream ist ein temporäres Abbild der Elemente einer Collection, auf welchen wir diverse vorgefertigte Methoden ausführen können.
- Gängige Methoden auf Streams sind: `forEach`, `sorted`, `filter`, `peek`, `findFirst`, `anyMatch`, `map`
- Es ist gängig die Stream-Methoden zu verketteten. Z.b. `filter(x-> x > 5).findFirst()`;
- Wollen wir das Ergebnis wieder in unsere Collection zurückschreiben müssen wir die `.collect(Collectors.to?())` Methode aufrufen.
- Es gibt auch einen `parallelStream()`, welcher die Aufgaben automatisch versucht in mehreren Thread abzuarbeiten. Mit Vorsicht genießen!

Aufgabe 3 – Umgang Streams

1. Sortiere die Stalltiere aufsteigend nach Gewicht, bevor diese durch die Automaten bearbeitet werden. Gebe diese Liste den Automaten mit.
2. Das Verrichten der Arbeit innerhalb der einzelnen Automaten soll durch einen `parallelStream` realisiert werden.
3. Teste die Geschwindigkeit der beiden stream-Möglichkeiten (`stream()` und `parallelStream()`) bei 1.000.000 Stalltieren mit Hilfe der Methode `anyMatch()`
4. Filter eine Liste von gefütterten Stalltieren nach dem erreichten Schlacht-Gewicht, mache aus dem Namen ein `UpperCase`, sortiere die Liste alphabetisch nach Namen und gebe schließlich die Elemente auf der Konsole aus.
Versuche das ganze in einem Aufruf darzustellen.
5. Bonus: Baue das Einlesen der Stalltiere aus der csv/xml/json Datei mit Hilfe der Stream API um.

Abschlussprüfung

<Siehe Dokumente>

Bonusday

- JavaFX
- SQL in Java / JDBC / Hibernate
- Maven
- Java Distributionen
- Chatprogramm
- Java Server (Jboss, Tomcat, Jetty, etc.)
- JavaEE (Enterprise Edition)
- Android
- ...