

# Linq4J

## Lehrziele:

- LambdaExpressions
- FunctionalInterfaces

Im Jahr 2007 veröffentlichte Microsoft NET 3.5 und führte mit Language Integrated Query (LINQ) eine neue Technologie ein, mit der die Arbeit mit Collections deutlich vereinfacht wurde. Ein Teil dieser Technologie waren LambdaExpressions, die nun (seit Java 8) auch unter Java verfügbar sind.

Linq ist praktisch eine Sammlung von Hilfsmethoden für Collections, die als sogenannte ExtensionMethods für das Interface IEnumerable zur Verfügung gestellt wurden. In Java wurde mit der Einführung der StreamingAPI ein ähnlicher Weg als Erweiterung des Interfaces Collection<T> beschritten.

In unserer Übung bilden wir Linq im kleinen Umfang nach, indem wir die Möglichkeit nutzen, Interfaces um Defaultmethoden zu erweitern. Da wir das Interface Iterable<T> leider nicht direkt erweitern können, erstellen wir ein neues Interface LinqIterable<T>, welches vorerst zwei „ExtensionMethods“ als DefaultMethods hinzufügt:

```
/**
 * Die Methode select() erweitert das Interface Iterable
 * um die Fähigkeit, alle Elemente für die die Predicate-Methode
 * where() true liefert in eine erstellte Zielcollection zu
 * übernehmen.
 * Als Ergebnistyp implementiert die Zielcollection wieder das
 * Interface LinqIterable und kann daher per MethodChaining
 * weiterverwendet werden.
 * @param whereClause eigentlich Methodenzeiger
 * @return Liste der passenden Elemente
 */
default LinqIterable<T> Select(Where<T> whereClause)
```

```
/**
 * Die Methode aggregate() ist eine universell einsetzbare Methode zur
 * Aggregation einer beliebigen LinqIterable-Collection. Durch die
 * Übergabe der Aggregationsfunktion (z.B. als LambdaExpression) kann mit
 * sehr wenig Aufwand eine beliebige Aggregation erreicht werden (z.B. Summe,
 * Mittelwert, Max, Min, ...)
 * @param <R> Typ des Aggregates
 * @param seed Startwert für die Aggregation (0 bei "Summen", 1 bei "Produkten")
 * @param aggregateFunction Funktion, die ein Element zur Aggregation hinzufügt
 * @return
 */
default <R> R aggregate(R seed, AggregateFunction<R,T> aggregateFunction )
```

Diese beiden Defaultmethoden sind zu implementieren!

Die Signatur der beiden zu übergebenden Funktionen ist über jeweils ein FunctionalInterface definiert (bereits in der Vorgabe enthalten):

```
@FunctionalInterface
public interface Where<T> {
    /**
     * Es wird geprüft, ob element die geforderte Bedingung erfüllt
     *
     * @param element
     * @return Erfüllt die bedingung?
     */
    public abstract boolean Where(T element);
}

/**
 * Generische Aggregationsmethode wird über ein Interface
 * definiert. Die konkrete Aggregation kann dann eine
 * Summe, Mittelwert, Erster Wert, letzter Wert, ... sein
 *
 * @param <R> Ergebnistyp
 * @param <T> Typ in der Collection
 * @author java@htl-leonding
 */
@FunctionalInterface
public interface AggregateFunction<R, T> {
    public R aggregateSingleElement(R aggregation, T element);
}
```

## Hauptprogramm:

Wir verwenden die neuen Möglichkeiten im Hauptprogramm, das folgende Aufgaben erfüllt

- Im Hauptprogramm verwenden wir eine Klassenliste (UTF-8 codiert mit LF als Zeilenumbruch), die wir in eine Liste<Pupil> einlesen. Die Klasse Pupil ist bereits vorgegeben.
- Selection und Ausgabe aller aus Linz stammenden Schüler (mit Select/Where)
  - Where als LambdaExpression realisiert
  - Sortierung nach dem Geburtsdatum
- Selection und Ausgabe aller Schüler mit gerader Katalognummer
  - Where wieder als LambdaExpression realisiert
- Erzeugung einer Aggregatsklasse „AggregateWeeks“, die eine Schülerliste aggregiert, indem sie das Alter aller Schüler aufsummiert und in Wochen ausgibt.

**Bildschirmausgabe:**

Schüler aus Linz, sortiert nach Geburtsdatum

15	Midžić	Enaidin	09.11.1998	Linz
3	Djukic	Pero	17.09.1999	Linz
16	Munoz Zapata	Joules	29.04.2000	Linz
14	Mayrhofer	Erik	27.01.2001	Linz
10	Hofmann	Daniel	13.07.2001	Linz

Schüler mit gerader Katalognummer

2	Deicker	René	11.05.2001	Scharnstein
4	Erhart	Lukas	19.11.1999	Engerwitzdorf
6	Gerstmayr	Hannes	15.11.2000	Sankt Florian
8	Hiebl	David	28.02.2001	Traun
10	Hofmann	Daniel	13.07.2001	Linz
12	Leithenmayr	Stefan	22.05.2001	Scharnstein
14	Mayrhofer	Erik	27.01.2001	Linz
16	Munoz Zapata	Joules	29.04.2000	Linz
18	Schürz	Jonas	03.06.2000	Kirchschlag bei Linz
20	Straßmayr	Christian	31.05.2001	Hofkirchen im Traunkreis
22	Wahl	Maximilian	03.01.2001	Ried in der Riedmark

Aus Leonding sind 2 Schüler

Alle Schüler zusammen sind 19696,73 Wochen alt