

Lab 06 – More LINQ

Hinweis 1: Das Aufgabenblatt enthält neben den Aufgaben zum Modulpraktikum auch Zusatzfragen zum Verständnis und zur Repetition des Stoffes. Diese Fragen müssen nicht für die Fallstudie beantwortet werden und sind freiwillig. Sie können jedoch zur Prüfungsvorbereitung verwendet werden.

Hinweis 2: In diesem Lab sollen Sie auch komplexere Abfragen mittels geeigneten LINQ und Lambda Ausdrücken erstellen. Die LINQ-Abfragen lassen oft sich als Einzeiler (oder mit sehr wenigen Codezeilen) durchführen.

Aufgabe 1

Implementieren Sie in der Klasse Links die folgenden Methoden für die folgenden Abfragen:

1. `int GetCountOfThreeBiggestCitiesInLinks()`
Bei wie vielen Links treten die drei bevölkerungsreichsten Städte aller Cities in den Links als Start-Stadt auf?
2. `int GetCountOfThreeCitiesWithLongestNameInLinks()`
Bei wievielen Links treten die Städte mit den drei längsten Städtenamen in den Links insgesamt auf, entweder als Start- oder als Ziel-Stadt?

Sie benötigen für diese Aufgabe Zugriff auf die interne City-Liste der Cities Klasse. Führen Sie dazu in der Klasse Cities z.B. eine readonly-Property `CityList` ein, die den Zugriff auf die Liste ermöglicht.

Aufgabe 2

Implementieren Sie in der Klasse Cities folgende Methode:

`int GetPopulationOfShortestCityNames()`

Wie gross ist die Bevölkerungszahl der drei Städte mit den kürzesten Städtenamen?

Aufgabe 3

Sie sollen den `RouteRequestWatcher` so erweitern, dass er pro Abfrage die City und das Datum erfasst werden.

Erweitern Sie zunächst die Klasse `RouteRequestWatcher` um folgendes:

```
public virtual DateTime GetCurrentDate { get { return DateTime.Now; }}  
private List<Tuple<City, DateTime>> cityRequestsDate = new List<Tuple<City, DateTime>>();
```

Stellen Sie nun sicher, dass bei jeder City Abfrage das Datum in der Liste `cityRequestsDate` abgelegt wird.

Hinweis:

Diese Methoden werden in der Test-Klasse überschrieben, um Tests mit verschiedenen Daten durchführen zu können. Ändern Sie weder Namen noch Datentypen davon.

Aufgabe 4

Implementieren Sie in der Klasse `RouteRequestWatcher` die Methoden für folgende Abfragen:

1. `IEnumerable<City> GetThreeBiggestCityOnDay(DateTime day)`
Was waren die drei bevölkerungsreichsten Städte, die an einem bestimmten Tag abgefragt wurden?

2. `IEnumerable<City> GetThreeLongestCityNamesWithinPeriod(DateTime from, DateTime to)`
Geben Sie die drei Städte mit den längsten Städtenamen zurück, die im gegebenen Zeitraum (inklusive from und to) abgefragt wurden.
3. `IEnumerable<City> GetNotRequestedCities(Cities cities)`
Welche Städte wurden in den letzten zwei Wochen nie als Ziel angefragt?

Hinweis: Sie können die Methode `LogRouteRequest` erweitern und die Anfragen mit Datum z.B. in einem Tupel speichern, welches Sie für obige Abfragen verwenden können:

```
private readonly List<Tuple<City, DateTime>> cityRequestsDate = new List<Tuple<City, DateTime>>();
```

Aufgabe 5

Testen Sie Ihre Implementierung mit den Unit-Tests in den Files «*Test_Lab06.cs» im «caseStudyFiles» Ordner.

Updaten Sie auch die neue Version der TestHelpers Klasse im File `TestHelpers.cs`.

Testfragen

- Über welchen Datentyp geben LINQ-Abfragemethoden ihre Resultate am besten zurück? Was sind die Vor- und Nachteile von `List<T>`, `IEnumerable<T>` oder `T[]`?
- Was sind Vor- und Nachteile von Local Queries gegenüber Remote Queries?
- Wie kann man die Ausführung einer LINQ-Abfrage als Local oder Remote Query forcieren?
- Welche LINQ-Methoden werden verzögert (*deferred*) ausgeführt, welche sofort?
- Was sind «Expression Trees»?