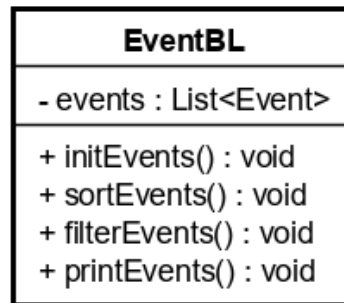
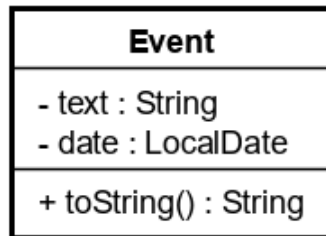


Pair-Programming-Exercise - Lambdas

Schreibe die Klassen **Event**, und **EventBL** um Termine, bestehend aus einem Text und einem Datum in einer Collection zu speichern.



A: Implementiere die Klasse **Event** entsprechend dem Klassendiagramm. Die Methode **toString()** gibt den Inhalt eines Objekts in folgender Form zurück:

POS-PLF – Di 12.11.2019

B: Implementiere die die Klasse **EventBL** entsprechend dem Klassendiagramm. Erzeuge eine generische Collection vom Typ **Event**. Schreibe die Methode **initEvents()**, die 25 zufällige Termine erzeugt. Das Datum der Termine ist ein zufälliger Wochentag (Kein Sa oder So) zwischen dem 9.9.2019 und dem 20.12.2019. Der Text setzt sich zufällig aus einem von 4 Fächern (z.B.: POS, DBI, NVS, TINF) und einer von 3 Aktivitäten (z.B.: PLF, Übung, Aufgabe) zusammen.

A: Implementiere Methode **sortEvents(boolean upwards)** die alle Termine auf- oder absteigend zuerst nach Datum dann nach Text entweder aufsteigend oder absteigend sortiert. Verwende dazu einmal eine Lambda-Expression und einmal statische Methoden aus dem Comparator-Interface. Kommentiere die nicht mehr benötigte Variante aus.

B: Implementiere Methode **printEvents()** um alle Termine zeilenweise auf der Konsole auszugeben. Verwende dazu keine for-Schleife sondern die **forEach()**-Methode der Liste. Verwende einmal eine Lambda-Expression und einmal eine Methoden-Referenz.

A: Implementiere Methode **filterEvents()** die alle Termine aus der Liste herauslöscht, die eine bestimmte Aktivität (z.B. PLF) haben. Verwende dazu keine for-Schleife sondern die **removeIf()**-Methode der Liste. Verwende eine Lambda-Expression.

B: Implementiere Methode **main()** um alle Methoden zu testen.