

Sommersemester 2013

Praktikum Software-Architektur

Blatt 3

Termine: 11. und 18. April 2013

## Softwaresystem zur Verwaltung von Laufdaten

### Aufgabe 1

Das erstellte UML-Klassendiagramm (vgl. Blatt 1, Aufgabe 1) soll nun implementiert werden.

1. Erstellen Sie für die Entitäten (Läufer, Veranstaltung, ...) entsprechende Klassen in Java. Überlegen Sie sich geeignete Methoden zum Traversieren von Assoziationen. Hinweis: Die Persistenz der Klassen wird in einer späteren Praktikumsaufgabe behandelt und muss daher in dieser Lösung nicht behandelt werden.
2. Des Weiteren erstellen Sie Java-Interfaces für die Schnittstellen zu den externen Systemen wie z.B. der Laufzeiterfassung.
3. Überprüfen Sie, ob die wichtigen Use Cases durch das implementierte Objektmodell abgedeckt werden.
4. Schreiben Sie Testfunktionen, die die korrekte Umsetzung der wichtigen Abläufe prüfen. Rufen Sie die Testfunktionen aus einer entsprechenden Test-Klasse auf.

### Aufgabe 2

Implementieren Sie die folgende grob-granulare Schnittstelle auf Basis der Lösung von Aufgabe 1:

```
package swa.runningeasy.services;

public enum Auswertung {STARTLISTE, GESAMTERGEBNISLISTE, NICHTSTARTER, ABBRECHER};

package swa.runningeasy.services;

import swa.runningeasy.dtos.AnmeldungDTO;
import swa.runningeasy.dtos.LaufzeitDTO;
import swa.runningeasy.dtos.VeranstaltungDTO;
import swa.runningeasy.dtos.VereinDTO;
import swa.runningeasy.dtos.LaeuferDTO;
import swa.runningeasy.dtos.ListeneintragDTO;
import java.util.List;

public interface RunningServices {
    void erzeugeVeranstaltung(VeranstaltungDTO v);
    void erzeugeVerein(VereinDTO v);
    void erzeugeAnmeldung(AnmeldungDTO a);
    void erzeugeLaeufer(LaeuferDTO a);
    void erzeugeLaufzeit(LaufzeitDTO l);

    List<VeranstaltungDTO> getVeranstaltungen();
    List<VereinDTO> getVereine();
}
```

```
List<LaeuferDTO> getLaeufer();  
List<AnmeldungDTO> getAnmeldungen(String Veranstaltung);  
List<LaufzeitDTO> getLaufzeiten(String Veranstaltung);  
List<ListeneintragDTO> getAuswertung(Auswertung a, String Veranstaltung);  
  
void init();  
}
```

Die als Parameter verwendeten “DTO”-Typen stellen hierbei Wertetypen (ohne Identität und Funktionalität) dar und werden nur zum Austausch von Daten zwischen Client und Server benutzt. Den Quellcode dieser Typen ist in der Datei `runningeasy_interface.jar` zu finden.

**Hinweis:** Achten Sie auf “sauberen” Programmcode, da dieser in zukünftigen Aufgaben verfeinert werden wird.

**Abgabe: 11. bzw. 18. April im Praktikum.**