

# Übung Open Closed Principle

In dieser Übung geht es darum das Open Closed Principle anzuwenden.

## Ausgangslage: Abfahrts-Fahrpläne

Sie befinden sich in einem Projekt bei einer bekannten Bahngesellschaft im Kanton Graubünden. Das Ziel des Projektes ist es Fahrplan-Daten so aufzubereiten, dass daraus Aushänge an den diversen Bahnhöfen mit den abfahrenden Zügen kreiert werden können. Hier ein kleiner Ausschnitt aus einem solchen Aushang:

# Abfahrt

## Bahnhof Klosters

05:00

| Zeit  | Linie  | Zielort  | Gleis |
|-------|--|--|-------|
| 05:21 |  RE12 | Landquart - Chur<br>Klosters Dorf 05:24 - Saas 05:33 - Küblis 05:38 -<br>Fideris 05:42 - Jenaz 05:46 - Furna 05:47 -<br>Schiers 05:52 - Grösch 05:57 -<br>Seewis-Valzeina 05:59 - Malans 06:03 -<br>Landquart 06:07 - Landquart Ried 06:12 -<br>Igis 06:13 - Zizers 06:16 -<br>Untervaz-Trimis 06:19 - Haldenstein 06:24 -<br>Chur 06:29 | 1     |
| 05:29 | M  | Davos Platz<br>Cavadunli 05:34 - Davos Laret 05:40 -<br>Davos Wolfgang 05:44 - Davos Dorf 05:50 -<br>Davos Platz 05:56   | 1     |

06:00

| Zeit  | Linie | Zielort   | Gleis |
|-------|-------|---|-------|
| 06:17 |       | Landquart - Chur<br>Klosters Dorf 06:20 - Saas 06:29 - Küblis 06:34 -<br>Fideris 06:38 - Jenaz 06:41 - Furna 06:43 -<br>Schiers 06:49 - Grösch 06:55 -<br>Seewis-Valzeina 06:57 - Malans 07:02 -<br>Landquart 07:06 - Landquart Ried 07:11 -<br>Igis 07:12 - Zizers 07:15 - | 3     |

08:00

| Zeit  | Linie | Zielort   | Gleis |
|-------|-------|---|-------|
| 08:25 | RE12  | Landquart - Chur - Disentis/Mustér<br>Klosters Dorf 08:28 - Saas 08:37 - Küblis 08:42 -<br>Jenaz 08:50 - Schiers 08:56 - Malans 09:00 -<br>Landquart 09:10 - Landquart Ried 09:16 -<br>Igis 09:20 - Zizers 09:22 -<br>Untervaz-Trimis 09:27 - Haldenstein 09:32 -<br>Chur 09:38 - Reichenau-Tamins 10:05<br>➔ weiter nach Disentis/Mustér | 2     |
| 08:28 | RE10  | Davos Platz<br>Cavadunli 08:33 - Davos Laret 08:39 -<br>Davos Wolfgang 08:44 - Davos Dorf 08:50 -<br>Davos Platz 08:56  | 1     |
| 08:32 | RE10  | Landquart<br>Küblis 08:48 - Schiers 08:59 - Grösch 09:04 -<br>Landquart 09:13   | 3     |
| 08:34 | RE12  | Scuol-Tarasp<br>Sagilains 08:53 - Lavin 08:56 - Guarda 09:00 -<br>Ardez 09:04 - Scuol-Tarasp 09:17  | 2     |
| 08:51 |       | Davos Platz<br>Cavadunli 08:57 - Davos Laret 09:03 -<br>Davos Wolfgang 09:08 - Davos Dorf 09:14 -<br>Davos Platz 09:20  | 1     |

Vor etwa 2 Iterationen haben Sie eine Komponente entwickelt, welche die Züge für die Darstellung in einem solchen Fahrplan auswählt. Hier ein Ausschnitt aus dem Source Code:

```
/**
 * Table of trains to display the relevant trains departing in a station.
 */
public class DepartureTable {

    private List<Train> departingTrains;

    /**
     * Initialize a departure table from a list of trains by filtering this list.
     *
     * @param trains the trains to use for the departure table (not yet filtered)
     * @param stationCode the code of the station for which to filter the trains
     */
    public DepartureTable(List<Train> trains, String stationCode) {

        departingTrains = new ArrayList<Train>();

        for (Train train : trains) {

            if (train.isPublic()) {

                Stop stop = train.getStop(stationCode);

                if (stop != null) {

                    if (stop.isStopToGetOn()) {

                        departingTrains.add(train);

                    }

                }

            }

        }

    }

}
```

```
/**
 * @return all the trains to display in departure table.
 */
public List<Train> getDepartingTrains() {
    return departingTrains;
}
}
```

Dieser Code wurde bereits released und wird mittlerweile sogar schon erfolgreich in anderen Projekten wiederverwendet. So verwendet zum Beispiel die Web-Applikation zur Verwaltung der Fahrplandaten diese Komponente um eine ähnliche Ansicht auch in der Webapplikation zur Verfügung zu stellen.

### Neue Anforderungen: Ankunfts-Fahrpläne

Das Projekt neigt sich dem Ende zu und Sie freuen sich bereits auf Ihr nächstes Projekt, als Sie einen Anruf vom Projektleiter erhalten. Dieser teilt Ihnen folgende neue Anforderung mit:

- Die Komponente soll es auch ermöglichen, ähnliche Fahrplantabellen zu kreieren, welche die ankommenden Züge an einem Bahnhof zeigen.

### Aufgabe 1: Erweiterung Ankunfts-Fahrpläne

Ihre Aufgabe besteht nun darin die Klasse DepartureTable so zu erweitern/refactor, dass auch die ankommenden Züge in einem Bahnhof aufgelistet werden können.

Sie können dafür die Interfaces Züge und deren Stops verwenden, wie sie in den folgenden 2 Seiten aufgelistet sind.

Stellen Sie sicher, dass ihre allfälligen Änderungen das Open Closed Principle für allfällige weitere ähnliche Änderungen erfüllen.

**Fertig?** Melden Sie sich bitte beim Kursleiter.

```
/**
 * Interface to Train objects as used in the Design Principles exercise.
 */
public interface Train {
    /**
     * @param station code of the station to get the stop for.
     * @return the stop of this train at the queried station. Returns 'null' if
     *         this train does not stop at the queried station.
     */
    public Stop getStop(String station);

    /**
     * @return true if train is public
     */
    public boolean isPublic();

    /**
     * @return special kind code for this train.
     *
     * Possible codes are:
     * <ul>
     * <li> "S" - Sonderzug, verkehrt nur an speziellen Tagen</li>
     * <li> "D" - Dispositionszug, verkehrt nur in speziellen Ausnahmefällen</li>
     * </ul>
     */
    public String getKind();
}
```

```
/**
 * Stop of a train in a station.
 */
public interface Stop {

    /**
     * @return true if this is a departure stop where passengers can get on.
     */
    public boolean isStopToGetOn();

    /**
     * @return true if this is an arrival stop where passengers can get off.
     */
    public boolean isStopToGetOff();

    /**
     * @return the code string of the station this stop is for (this is the
     *         unique string representing the station).
     */
    public String getStationCode();

    /**
     * @return the departure time in format "hh:mm" with leading zeroes.
     */
    public String getDepartureTime();

    /**
     * @return the arrival time in format "hh:mm" with leading zeroes.
     */
    public String getArrivalTime();
}
```