

Neue Anforderungen: Bahnhofs-Übersicht in Webapplikation

Gratulation. Ihre erweiterte Komponente wurde erfolgreich released und auch bereits in der Webapplikation verwendet.

Dies hat allerdings die Entwickler und die Benutzer der Webapplikation auf neue Ideen gebracht ...

Man will Ihre Komponente nun gerne verwenden um einen speziellen „Bahnhofs-Übersichts-Fahrplan“ zu generieren. Dieser soll sowohl die abfahrenden Züge, wie auch die ankommenden Züge in der gleichen Liste enthalten.

Aufgabe 2: Bahnhofs-Übersicht

Skizzieren Sie, wie diese neue Anforderung einfach umgesetzt werden kann, ohne dass dafür ihre Komponente modifiziert werden muss.

Falls eine entsprechende Erweiterung ihrer Komponente nicht einfach so möglich ist, passen Sie bitte das Design wieder gemäss Open Closed Principle an.

Wartung der Komponente: Spezielle Züge

Nach Auslieferung der Applikation wird plötzlich bemerkt, dass in den Fahrplänen Züge auftauchen, welche eigentlich nicht auftauchen sollten.

Eine Analyse des Problems zeigt, dass es sich dabei um spezielle Züge mit folgenden Eigenschaften handelt:

- **Dispozug (kind = „D“):** Züge welche provisorisch geplant werden, als Ersatzzüge bei allfälligen Problemen auf dem Streckennetz.
- **Sonderzug (kind = „S“):** Sonderzüge für spezielle Anlässe, welche nicht im öffentlichen Fahrplan erscheinen.

Aufgabe 3: Spezielle Züge unterdrücken

Beide Zugarten sollen in allen Fahrplan-Aushängen nicht erscheinen. Auch in der Webapplikation sollen sie nicht in der Bahnhofs-Übersichts-Tabelle erscheinen.

Passen Sie Ihre Komponente entsprechend an.

Falls Sie Ihr Klassen-Design geschickt gewählt haben sollte diese Änderung ganz einfach an einem Ort möglich sein.

Zusatz-Aufgabe: Sortierung

Ihre Komponente soll nun neu, die Tabellen-Einträge nicht nur sinnvoll auswählen können sondern auch noch sortieren (je nach Tabellen-Typ nach Ankunfts- oder Abfahrts-Zeit). Skizzieren Sie ein sinnvolles Design ...)