**Inverted Exam**

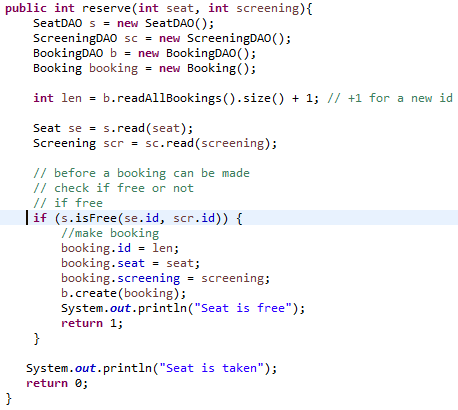
***Lösung***

**Dokumentation der gefundenen Performance Probleme & deren Lösungsumsetzung:**

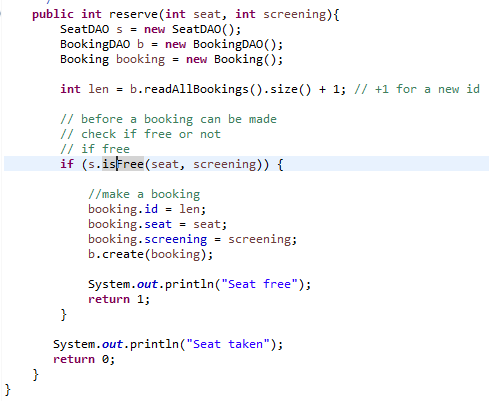
**1.)**

Im CustomerDAO können in der Methode reserve() gleich die übergebenen IDs verwendet werden anstatt zuerst ein Seat- und Screening-Objekt zu erstellen.

**Problem:**



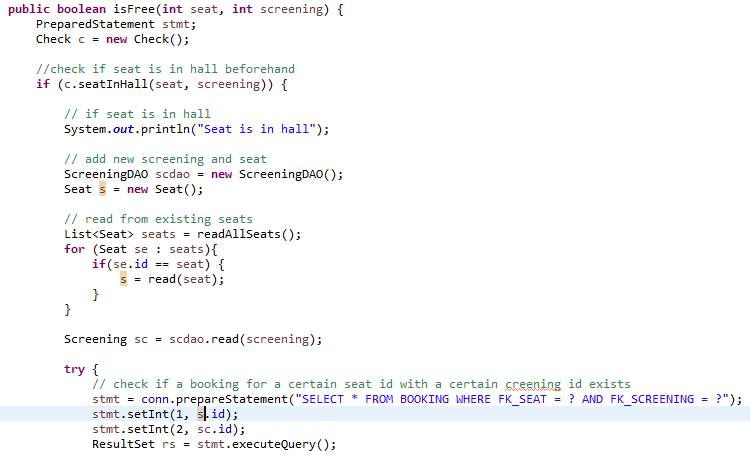
**Lösung:**



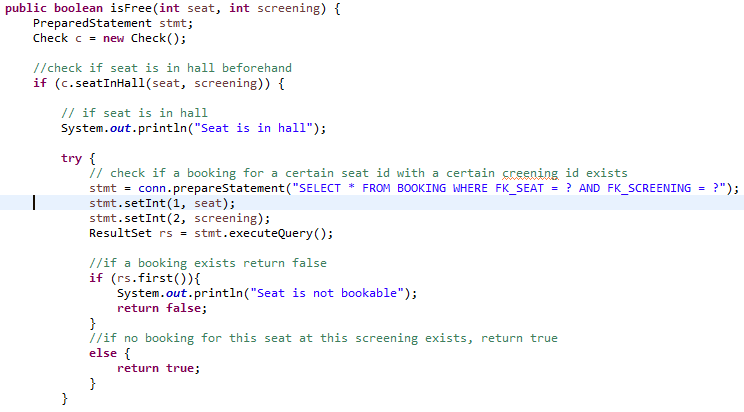
**2.)**

Des Weiteren können im SeatDAO in der Methode isFree() ebenfalls sofort die übergebenen IDs verwendet werden anstatt zuerst ein Seat- und Screening-Objekt zu erstellen.

**Problem:**



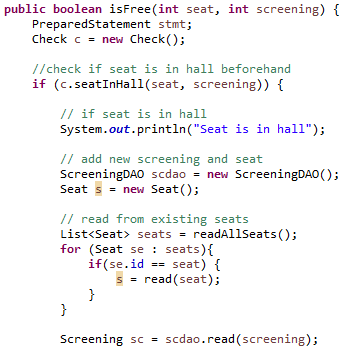
**Lösung:**



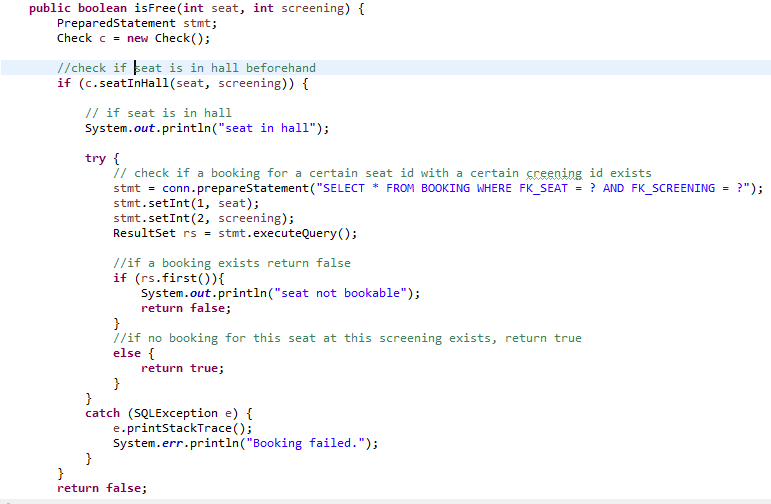
**3.)**

Zusätzlich werden unsinnigerweise im SeatDAO in der Methode isFree() alle Seats ausgelesen. Danach wird überprüft, ob die IDs mit der übergebenen übereinstimmen und wenn ja, wird der Seat in der Variable, die in Lösungspunkt 2 erstellt worden ist, gespeichert.

**Problem:**



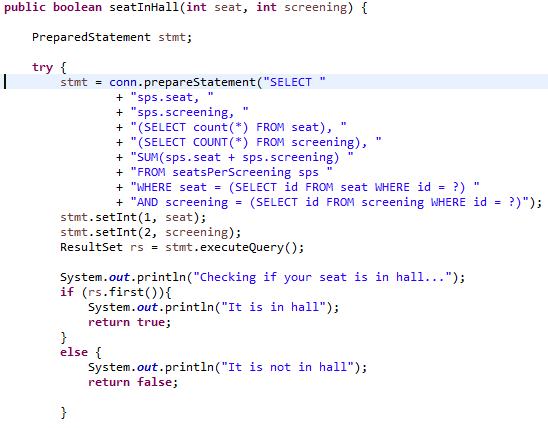
**Lösung:**



**4.)**

In der Klasse Check gibt es die Methode seatInHall(), in welcher überprüft wird – wie der Name schon sagt – ob sich ein bestimmter Seat in einer bestimmten Hall befindet. Dafür wird mitunter eine SQL-Abfrage nötig, doch bei der gegebenen SELECT-Abfrage sind einige Unterabfragen irrelevant. Wichtig ist ja nur, ob ein Seat mit ID x und ein Screening mit ID x dieselbe hall ID x in der „gejointen“ View seatsPerScreening besitzen, also ob der geforderte Seat bei dem betreffenden Screening in dem entsprechenden Saal ist oder nicht.

**Problem:**



**Lösung:**

