

Birçok müşterisinin; belirli şehirler arasındaki uçuşlar için koltuk rezervasyonu yapabildiği bir havayolu rezervasyon sistemi düşünün. Uçuşlar ve koltuklar hakkındaki bilgi ortak bir veritabanında bulunmaktadır. Veritabanı birçok girdiden oluşmaktadır. Bu girdilerin her biri belirli şehirler arasında belirli bir güne ait bir koltuğu temsil etmektedir. Tipik bir havayolu rezervasyon sisteminde, bir müşteri kendi ihtiyaçlarını karşılayan optimal bir uçuş aramaktadır. Yani, bir müşteri bir koltuğu rezerve etmeden önce birçok uçuş araması yapabilir. Bu arama sürecinde bir koltuk başka bir müşteri tarafından rezerve edilebilir. Böyle bir durumda eğer müşteri o koltuğu rezerve etmeye kalkarsa, müşteri bilginin bu süre içerisinde değiştiğini ve o koltuğun başkası tarafından alındığını fark eder.

Uçuş arayan müşteri reader'dır. Koltuk rezerve etmeye çalışan müşteri ise writer'dır. Anlaşıldığı üzere, birden çok sayıda reader paylaşılmış veriyi okuyabilirken, her writer'ın paylaşılmış veriye ulaşması için özel bir hakka sahip olması gerekmektedir. Aksi halde birden çok writer aynı koltuğu alabilir.

Birçok reader ve birçok writer thread'i olan ve her birinin tek bir rezervasyon kaydına erişmeye çalıştığı multithreaded bir Java programı yazınız. Bir writer thread'in **makeReservation** ve **cancelReservation** olmak üzere iki işlemi vardır. Reader thread'inin ise bir tane **queryReservation** işlemi vardır.

İlk önce programınızın, threadlerine; rezervasyon kaydına asenkron bir şekilde erişim izni verdiği halini yazınız. Bu haliyle programı çalıştırdığınızda veritabanındaki bütünlüğün bozulduğunu gösteriniz. Ardından programınızın, reader ve writerlar'ın paylaşılan veriye belirli bir protokol ile erişmesinin sağlanması için Java thread senkronizasyonunun Lock'lar ile gerçekleştirildiği yeni bir versiyonunu oluşturun. Yani programınızda hiçbir writer thread aktif değilken birden çok reader paylaşılan veriye erişebilmelidir.

Ancak dikkatli olunuz. Bu problemin incelikleri vardır. Örneğin, bir anda birçok reader thread paylaşılmış veriye erişmiş ve writer paylaşılan veriye erişmek isterse ne olacak? Eğer sürekli readerlere izin verilirse writer thread sürekli ertelenmiş olacaktır. Bu problemi çözmek için writerlara öncelik verebilirsiniz. Ancak bu çözümde de writerlar readerların sürekli önüne geçebilir. Bu problemleri nasıl çözdüğünüzü açıklayınız.

Programınız çalıştığı sürede neler olup bittiğini gösteren güzel tasarlanmış bir arayüz oluşturun. Yeterli sayıda örnek çıktı oluşturun.