

## Verteilte Systeme WS12/13

### Übung 2 – Nebenläufigkeit mit Threads

#### Aufgabe: Parkhaus

**Achtung:** Änderungen gegenüber der Aufgabe aus Übung 1!

Ein Parkhaus mit 4 Parkplätzen hat von 8:00 Uhr bis 20:00 Uhr geöffnet. Über den Tag verteilt kommen Autos (identifiziert durch ihr Kennzeichen), um eine schon vorher bekannte Zeit zu parken. Wenn das Parkhaus voll ist, stellen sich die Autos in die (beliebig große) Warteschlange vor die Eingangsschranke. Nach Einlass in das Parkhaus darf das Auto sich einen freien Parkplatz aussuchen.

Der Ablauf der Simulation ist in einer Konfigurationsdatei (csv) beschreiben. Ein Beispiel: die Zeile „B-MW4444;8:25;2:47“ bedeutet: der BMW kommt um 8:25 Uhr vor dem Parkhaus an. Im Gegensatz zum wirklichen Leben stellt sich der Fahrer so lange in die Warteschlange, bis die Schranke für ihn aufgeht. Vom Zeitpunkt der Einfahrt ab kauft er ca. zweieinhalb Stunden lang ein, das heißt, 2 Stunden und 47 Minuten nach der Einfahrt verlässt er das Parkhaus (und die Simulation).

Geben Sie mit einer einfachen zeichenbasierten Darstellung die Situation von Parkhaus und Warteschlange davor zu jedesmal aus, wenn sich der Zustand (Belegung der Parkplätze, Warteschlange) sich ändert. Erzeugen Sie eine interessante Beispielformatierung, bei der auch Konflikte (gleichzeitiges Ankommen, voll belegtes Parkhaus) auftreten.

#### Realisierung

Implementieren Sie die Aufgabe in Java mit Threads (und Synchronisation). Der Haupt-Thread ist das Parkhaus (einschließlich Warteschlange), er verwaltet im Wesentlichen die Ressourcen wie Parkplätze oder Warteschlange. Dieser Thread wird in den folgenden Übungen zum Server. Für jedes Auto wird ein neuer Thread erzeugt; wenn das Auto wieder herausfährt, wird der Thread beendet.