

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 bekommt das Auto einen freien Parkplatz zugewiesen.

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. 2 Stunden und 47 Minuten nach der Einfahrt verlässt er das Parkhaus (und die Simulation).

Geben Sie jedes Mal, wenn sich etwas am Parkhaus ändert, dies aus. Erzeugen Sie eine interessante Beispielkonfiguration, bei der auch Konflikte (gleichzeitiges Ankommen, voll belegtes Parkhaus) auftreten.

technische Spezifikation

Implementieren Sie die Aufgabe in Java mit Threads (und Synchronisation). Der Haupt-Thread erzeugt das Parkhaus als ein Singleton-Objekt, welches über den Konstruktor allen Autos bekanntgegeben wird. Für jedes Auto wird, entsprechend der Angaben aus der CSV-Datei, ein neuer Thread erzeugt; wenn das Auto wieder heraus fährt, wird der Thread beendet. Nach 12 Stunden wird das ganze Simulationsprogramm beendet, unabhängig davon, ob noch Autos im Parkhaus, in der Warteschlange oder noch zu Hause stehen. Eine Stunde in der Realität soll einer Sekunde in der Simulation entsprechen. Bei jeder Änderung in Parkhaus oder Warteschlange wird diese vom Parkhaus ausgegeben.

Realisierung der Kommunikation

Achtung: Änderungen/Vereinfachungen gegenüber der ersten Version!

Ein Auto schläft so lange (sleep), bis die erste Zeit (Ankunft am Parkhaus) erreicht ist, dort stellt es sich in die Warteschlange. Alle 6 Minuten fragt es das Parkhaus, ob es vorrücken kann (falls die Warteschlangenposition nicht ganz vorne ist), oder ob es einfahren kann. Im positiven Fall wird der Vorgang ausgeführt, bei einer negativen Auskunft wird weitere 6 Minuten gewartet (sleep). Wenn das Auto eingefahren ist, sitzt es die Parkzeit ab (sleep), benachrichtigt dann das Parkhaus von seiner Abfahrt und beendet sich.

Der Haupt-Thread liest die CSV-Datei, erzeugt ein Parkhaus-Objekt, instanziiert und startet die

vorgegebene Anzahl von Autos. Danach legt er sich 12 Stunden schlafen und kommt um 20:00 Uhr zurück, um die Tore zuzuschließen (`System.exit(0)`).

In der eCampus-Übung finden Sie eine Musterlösung, welche einen (ziemlich sinnfreien ;-)) Ablauf realisiert.