

Vorlesung Software-Entwicklung II SS 2004 - Übungsblatt 7

Ausgabe: 04.06.04 -- Abgabe: keine

Aufgabe 15 (Monitor, Bedingungssynchronisation: Entwurf)

Ziel: Die Sitzplatzbelegung in einem Restaurant soll simuliert werden. Besucher betreten in Gruppen das Restaurant, suchen sich einen Tisch, essen und verlassen das Restaurant wieder.

Modellierung:

- **Tische:** An jedem Tisch steht eine bestimmte Anzahl von Stühlen, einige davon sind belegt, die restlichen Stühlen sind frei.
- **Restaurant:** Ein Restaurant besteht aus mehreren Tischen. Alles andere ist für diese Simulation nicht relevant.
- **Besucherguppen:** Die Gruppen bestehen aus einer bestimmten Anzahl Personen und wissen im voraus, wie lange Sie im Restaurant bleiben werden (nachdem sie einen Platz gefunden haben). Besuchergruppen werden als Prozesse (*Threads*) modelliert. (Auch in der Realität handeln die Leute gleichzeitig und unabhängig voneinander).

Verhalten der Besuchergruppen

1. Restaurant betreten
2. Platzsuche gemäß "westfälischer" Strategie (siehe unten)
3. Aufenthalt im Restaurant
4. Restaurant verlassen

Strategie bei der Platzsuche einer Gruppe aus n Personen

Bei der westfälischen Sitzplatzsuche trennen sich Gruppen niemals auf. Außerdem will eine Gruppe lieber allein am Tisch sitzen als sich mit einer fremden Gruppe einen Tisch zu teilen. Später soll sich möglichst kein Fremder zu der Gruppe dazusetzen können. Formal läuft die Platzsuche also so ab:

1. Versuche, einen noch völlig leeren Tisch mit n Stühlen zu finden.
2. Versuche, einen völlig leeren Tisch mit mindestens n Stühlen zu finden.
3. Versuche, einen Tisch mit mindestens n freien Stühlen zu finden.
4. Beginne wieder ab 1. bis Sitzplätze gefunden wurden

Aufgaben: (Entwurf)

- a) Welche Objekte der Simulation sollen als Monitor wirken (siehe Folie 143)?
- b) Welche Monitor-Operationen werden benötigt (siehe Folie 143)? An welchen Stellen ist Bedingungssynchronisation erforderlich (siehe Folie 145)? Wie lauten diese Bedingungen ?

Hinweis: Legen Sie für die zusätzlichen Methoden zunächst nur die Signaturen und die zu erledigenden Aufgaben fest. Überlegen Sie für jede dieser Methoden, ob sie *synchronized* sein muss (Begründung!). Denken Sie beim Entwurf an die umzusetzende Strategie für die Platzsuche.

Aufgabe 16 (Monitor, Bedingungssynchronisation: Implementierung)

- a) Implementieren Sie Ihren Entwurf aus Aufgabe 15 in Java. Ergänzen Sie dazu die Klassen `Tisch`, `Restaurant` und `BesucherGruppe` um die noch fehlenden Methoden. Vervollständigen Sie die `run`-Methode, so daß sich jede Gruppe wie oben beschrieben verhält.

Die Datei `Tisch.java` enthält den Rahmen für eine entsprechende Klasse `Tisch`. Einen geeigneten Rahmen für die Klasse `BesucherGruppe` finden Sie in der Datei `BesucherGruppe.java`, einen Rahmen für die Klasse `Restaurant` in der Datei `Restaurant.java`.

- b) Testen Sie Ihr Programm mit der `main`-Methode aus der Klasse `Restaurant`.

Aufgabe 17 (Monitor, Auflösung von Verklemmungen: Implementierung)

Im folgenden Szenario wird ein Kfz-Werkstatt-Betrieb simuliert. Eine Werkstatt verfügt über die beschränkten Ressourcen Hebebühne, Mechaniker und Werkzeuge. Des Weiteren gibt es Autos, die beschädigt sind und in der Werkstatt repariert werden sollen. Schäden unterscheiden sich im Bedarf an Mechanikern und Werkzeugen, pro Auto wird immer eine Bühne benötigt.

- a) Schauen Sie sich die in `Werkstatt.java`, `Auto.java` und `Schaden.java` vorliegende Implementierung an. Welches Problem bzw. welche Probleme treten bei dieser Realisierung auf?
- b) Entwickeln Sie einen Lösungsansatz für das von Ihnen in Teil a) festgestellte Problem bzw. für die von Ihnen in Aufgabenteil a) festgestellten Probleme.
- c) Ändern Sie die Implementierung der `Werkstatt` entsprechend Ihrer Ergebnisse aus Aufgabenteil a) und b).

Hier finden Sie die Folien aus der Zentralübung `zue7.pdf`.