

Thread-Übung Bank

Ein Objekt Bank modelliert eine reale Bank, die Konten für verschiedene Personen verwaltet. Diese hat eine Anzahl von Überweisungen durchzuführen. Da auch in einer realen Bank mehrere Angestellte beschäftigt sind, die sich gleichzeitig um die Bearbeitung solcher Überweisungen kümmern, werden auch im Programmmodell die Überweisungen in parallelen Threads abgearbeitet.

In der folgend dargestellten Implementierung wird die Bank durch eine Klasse `SimpleBank` realisiert. Diese Bank besitzt das Array `konten`, das die Stände der einzelnen Konten enthält. Der Array-Index soll als Kontonummer dienen. Für Transaktionen zwischen zwei Konten stellt `SimpleBank` die Methode `überweisung()` zur Verfügung. Ihr werden die beiden beteiligten Kontonummern sowie der Betrag der Überweisung übergeben.

Zu Demonstrationszwecken wird die Ausführung der Überweisung zwischen der Berechnung des neuen Kontostands und dessen Zurückschreiben unterbrochen. Hierzu wird auf die bekannte Methode `sleep()` zurückgegriffen.

Als letztes verfügt `SimpleBank` über die Methode `kontostand()`, mit der die aktuellen Stände aller Konten angezeigt werden können.

Die Überweisungen sollen durch Bankangestellte erfolgen. Diese werden durch die Klasse `Angestellter` implementiert. Damit mehrere Angestellte gleichzeitig Transaktionen vornehmen können, werden sie von [Thread](#) abgeleitet. Jeder Angestellte gehört zu einer Bank. Deshalb wird dem Konstruktor ein Verweis auf `SimpleBank` übergeben. Über diesen Verweis wird die Methode `überweisung()` aufgerufen. Die beiden beteiligten Kontonummern sowie der Betrag werden dem Konstruktor übergeben und in entsprechenden Datenelementen der Klasse gespeichert.

Diese Datenelemente werden dann in `run()` benutzt, um die Methode `überweisung()` der Bank aufzurufen. Nachdem die Transaktion durchgeführt ist, wird eine aktuelle Kontenübersicht ausgegeben.

Setzen Sie die Problemstellung um und überprüfen Sie die Ausgaben auf der Konsole. Machen Sie Ihr Programm threadsicher.