Übung Java Thread Diagonal

Ausgangslage

Mit dieser Übung soll ein Punkt durch 2 Threads so innerhalb des Koordinatensystems bewegt werden, dass dieser die Diagonale nicht verlässt. Die folgende Grafik zeigt dies auf:Hierzu sind die folgenden Klassen zu programmieren:Zu den einzelnen Klassen:Die Klasse Point beschreibt den Punkt. Die Attribute x und y enthalten die Koordinaten des Punktes. Die Methode void movePoint (int x, int y) verschiebt den Punkt relativ. Die Methode boolean isDiagonal () gibt den Wert true zurück, falls sich der Punkt auf der Diagonalen (x == y || -x == y) is true befindet, andernfalls false. Die Methode String toString() soll den Zustand der Punktinstanz als String zurückgeben.Diese Klasse MovePoint verschiebt den Punkt über die Method movePoint(..) der Point-Instanz. Die Klasse implementiert das Interface Runnable und damit die Methode void run(). MovePoint enthält ein Attribut vom Typ Point Die Attribute xmove und ymove enthalten den relativen Wert der Verschiebung.Diese Klasse Diagonal Diese Klasse bildet die eigentliche Anwendung. Innerhalb der Methode main() wird eine Point-Instanz erzeugt und alsdann die beiden MovePoint-Instanzen, wobei eine Instanz den Punkt nach Nordwesten mit den Werten -1,1 und die andere den Punkt nach Südosten mit dem Wert 1,-1 verschiebt. Alsdann werden beide Threads erzeugt und gestartet. Innerhalb einer Endlosschleife sollen die Positionen des Punktes über die Methode toString() am Bildschirm angezeigt werden.

Vorgehen

Lösen Sie bitte die Aufgabe wie folgt:Erzeugen Sie ein neues Java Projekt für diese Übung.Programmieren Sie die Klasse Point gemäss der Beschreibung aus.Programmieren Sie die Klasse MovePoint gemäss der Beschreibung aus.Programmieren Sie die Klasse Diagonal gemäss der Beschreibung aus.Kompilieren Sie das Programm und führen Sie es bitte aus.Verifizieren Sie, ob sich der Punkt immer auf der Diagonalen befindet.Korrigieren Sie das Programm, bis der Punkt wirklich diagonal verläuft.

Lösung

Eine mögliche Lösung finden Sie hier

Zusatzaufgabe ReentrantLock

Programmieren Sie die Synchronisation und damit Threadsafety mit dem Lock Interface und der Implementation ReentrantLock um. Damit ersetzen wir das synchronized Keyword.

Lösung ReentrantLock

Eine mögliche Lösung finden Sie hier

Zusatzaufgabe ExecutorService

Neu sollen die Threads durch einen fixen Thread Pool und damit den ExecutorService ersetzt werden. Programmieren Sie das Dialog Beispiel entsprechend um und verwenden Sie einen fixen Thread Pool mit maximal 2 Threads.

Lösung ExecutorService

Eine mögliche Lösung finden Sie hier

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen
Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsauschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

-ausbildung-java-kurs-java-advanced-kurs-java-advanced---ressourcen-kurs-java-advanced---übunge