

Multithreading in Swing

Dr. Andreas Schroeder



Dieses Video behandelt Multithreading-Pannen in Swing:

- Eingefrorene Benutzerschnittstelle
- Nebenläufige Änderungen

- Symptome
 - UI-Elemente reagieren nicht
 - UI wird nicht neu gezeichnet
 - Frame kann nicht geschlossen werden
- Ursache
 - Ausführung längerer Berechnungen im Event Dispatch Thread (EDT)
 - Warten auf Antworten über Netzwerk im EDT
 - Endlos-Schleife im EDT, die durch Benutzerinteraktion abgebrochen werden soll.



EDT Aufgaben

- UI-Events über Listener an die Anwendung ausliefern
- UI neu zeichnen



Während der EDT im Anwendungskern gehalten wird

- Können keine UI-Events weitergegeben werden
- Kann die UI nicht neu gezeichnet werden





- Wissen, durch welchen Thread eine Methode aufgerufen wird – alle Listener werden vom EDT aufgerufen!
 - Bei Unsicherheit: `SwingUtilities.isEventDispatchThread`
- Längere Aufgaben mit `SwingWorker` im Hintergrund ausführen lassen
 - `doInBackground` läuft im Hintergrund
 - `done` läuft im EDT
- Thread-Wechsel zurück zum EDT mit `SwingUtilities.invokeLater` oder `SwingUtilities.invokeAndWait`





*Bild von Astrorek,
Wikimedia commons*

- Symptome
 - UI läuft weiter, aber Aktion bricht halb ausgeführt ab
 - Exception stack trace in der Konsole:
NullPointerException,
(Array-)IndexOutOfBoundsException
- Ursache
 - Swing ist nicht Thread-Safe



- Alle Methoden von Swing-Objekten müssen im EDT aufgerufen werden
- Ausnahmen
 - `isEventDispatchThread`
 - `invokeLater`
 - `invokeAndWait`
 - `repaint`
- Wissen, durch welchen Thread eine Methode aufgerufen wird – Hintergrund-Thread oder EDT?
 - Check mit `isEventDispatchThread`
 - Wechsel mit `invokeLater` und `invokeAndWait`



Was in diesem Video behandelt wurde sind Symptome, Ursachen und Lösungsansätze für

- Eingefrorene UIs
- Exceptions durch Nebenläufige Änderungen

<http://docs.oracle.com/javase/tutorial/uiswing/concurrency/>

<http://docs.oracle.com/javase/tutorial/essential/concurrency/>

