

Kapitel 13 – GUI und Threads:

Zusammenfassung + Übungen

In diesem Kapitel lernst du die Grundlagen für grafische Benutzeroberflächen (GUIs) mit Swing sowie die Arbeit mit Threads. GUI ermöglicht die Interaktion per Fenster, Buttons und Eingabefeldern. Threads erlauben parallele Abläufe im Programm.

Themenübersicht

1. GUI mit Swing

Swing ist ein Java-Toolkit für grafische Oberflächen. Wichtigste Klassen: JFrame, JButton, JLabel.

2. Fenster erstellen (JFrame)

Ein Fenster wird mit JFrame erstellt:
`JFrame fenster = new JFrame("Titel");`

3. Komponenten hinzufügen

Elemente wie Buttons, Textfelder oder Labels werden per `add()` zum JFrame hinzugefügt.

4. Layouts

Bestimmen, wie Komponenten angeordnet werden. z. B. BorderLayout, FlowLayout.

5. Ereignisse (Events)

z. B. Mausklicks. Man verwendet ActionListener, um darauf zu reagieren.

6. Threads

Ein Thread führt parallel Code aus. Erstellt mit `extends Thread` oder `implements Runnable`.

7. Thread starten

Mit `start()` wird ein neuer Thread gestartet, nicht mit `run()` aufrufen!

8. Synchronisierung

Mit `synchronized` kann man kritische Abschnitte schützen, wenn mehrere Threads darauf zugreifen.

Übungsaufgaben

1. Erstelle ein einfaches JFrame-Fenster mit dem Titel "Mein Fenster".

2. 2. Füge dem Fenster einen JButton hinzu, der "Klick mich" anzeigt.
3. 3. Baue einen ActionListener, der bei Klick den Text "Button wurde gedrückt" ausgibt.
4. 4. Verwende ein FlowLayout, um zwei Buttons nebeneinander anzuordnen.
5. 5. Erstelle eine Klasse `MeinThread`, die `Runnable` implementiert und Text in einer Schleife ausgibt.
6. 6. Starte zwei Threads parallel, die jeweils Zahlen von 1 bis 5 ausgeben.
7. 7. Schütze den Zugriff auf eine gemeinsame Variable mit `synchronized`.
8. 8. Warum sollte man `start()` statt `run()` bei Threads verwenden? Erkläre.