Outline

1. What is a Design Pattern?  
   In software engineering, a design pattern is a general reusable solution to a commonly occurring problem in software design.
   1. Why patterns?
   2. Criticism
2. GoF: Elements of a Design Pattern
   1. Pattern name
   2. Problem
   3. Solution
   4. consequences
3. Organizing Design Patterns
   1. Purpose
   2. Scope
4. Overview about important Design Patterns
   1. Singleton
   2. Composite
   3. Decorator
   4. Memento
5. Patterns for Concurrent and Networked Objects
   1. Wrapper Façade
   2. Reactor
   3. Double-Checked Locking Optimization
   4. Active Object
6. Demo application for ActiveObject Pattern

# Sources:

* **GoF book:**  
  Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software
* **POSA1:**  
  Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerland, Michael Stal: Pattern-Oriented Software Architecture – A System of Patterns
* **POSA2:**  
  Frank Buschmann, Hans Rohnert, Michael Stal, Douglas Schmidt: Pattern-Oriented Software Architecture – Patterns for Concurrent and Networked Objects
* <http://www.tutorialspoint.com/design_pattern/design_pattern_overview.htm>
* <http://www.tutorialspoint.com/design_pattern/command_pattern.htm>
* <https://en.wikipedia.org/wiki/Software_design_pattern>
* <http://www.cs.wustl.edu/~schmidt/POSA>
* <http://shairosenfeld.com/concurrency.html> and <http://shairosenfeld.com/sbvc/>
* Excursion about Double Checked Locking: <http://www.cs.umd.edu/~pugh/java/memoryModel/DoubleCheckedLocking.html>
* List of different patterns: <https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Architecture/Design_Patterns#cite_note-POSA2-8>
* <http://www.philipphauer.de/study/se/design-pattern.php>