

# Exzerpt 4:

## Dokument:

Entwurfsm\_modularisierung\_06.pdf

|                  |   |
|------------------|---|
| <b>Preview:</b>  | <p><b>Überschriften:</b></p> <p>Software Module</p> <p><b>Beispiel:</b><br/>Modul ErrorLogger zum Speichern von Fehlermeldungen</p> <p>Erwünschte Eigenschaften von Modulen</p> <p>Stabile und verständliche Modulschnittstellen durch Information Hiding</p> <p>Die Kunst beim Entwurf eines Moduls und seiner Schnittstelle besteht darin,</p> <p>Balance zwischen Kopplung und Kohäsion</p> <p>Kohäsion und Kopplung von Modulen</p> <p>Ausprägungen von Modulen</p> <p>Modul als Abstrakte Datenstruktur (ADS)</p> <p>Modul als Abstrakter Datentyp (ADT)</p> <p><b>Abbildungen und Diagramme:</b></p> <ul style="list-style-type: none"><li>- Abb. 1 Schematische Darstellung des Moduls ErrorLogger</li><li>- Abbildung .3: Kohäsion und Kopplung von Modulen:</li><li>- Abbildung 4 Zwei Instanzen der Klasse <b>ErrorLogger</b></li></ul> |
| <b>Question:</b> | <ul style="list-style-type: none"><li>[1] Was sind Module?</li><li>[2] Wie modelliert man mit Modulen?</li><li>[3] Wie sehen die verschiedenen Ausprägungen von Modulen aus?</li><li>[4] Wie sieht ein Modul als ADT aus?</li><li>[5] Was versteht man unter Kohäsion?</li></ul>  |
| <b>Read:</b>     | <ul style="list-style-type: none"><li>- <b>Zu [1]: Module</b> hingegen fassen mehrere Prozeduren u. Funktionen sowie die Daten, auf denen diese operieren, zu einer Einheit zusammen.</li><li>- <b>Zu [2]:</b> Durch die geschickte Zusammenfassung von Funktionen, Prozeduren und Daten können Abstraktionen gebildet werden, die den Entitäten der realen Welt, die es in einem Software-System abzubilden gilt, entsprechen.</li><li>- <b>Zu [3]:</b> Modul als abstrakte Datenstruktur, bzw. als abstrakter</li></ul>   |

|                 |   |
|-----------------|---|
|                 | <p>Datentyp</p> <ul style="list-style-type: none"> <li>-</li> <li>- <b>Zu [4]:</b> Benötigen wir mehrere Exemplare von einem Modul, ist es wünschenswert, einen Modul als Typ zu deklarieren, von dem beliebig viele Instanzen gebildet werden können</li> <li>- <b>Zu [5]:</b> Unter Modulkohäsion verstehen wir den Zusammenhalt eines Moduls. Der ist dann gegeben, wenn ein Modul Operationen und Daten zu einer Einheit zusammenfasst, die logisch zusammengehören.</li> </ul> |
| <b>Reflect:</b> | <p><b>Bekannte Aspekte:</b></p> <ul style="list-style-type: none"> <li>- Weitestgehend alles bekannt, nur etwas tiefer</li> </ul> <p><b>Unbekannte Aspekte:</b></p> <ul style="list-style-type: none"> <li>- Kohäsion</li> </ul>  |
| <b>Recite:</b>  | <p>Der Text beleuchtet den Modulaspekt und dessen Ausprägung nochmals etwas genauer. Außerdem zeigt er zur Verdeutlichung den Entwurf eines ErrorLogger- Moduls.</p>  |
| <b>Review:</b>  | <p>Dieser Text ist sehr wichtig für meine Ausarbeitung, da er das für das Geheimnisprinzip essentielle Modulkonzept näher beleuchtet.</p>   |

## Keywords:

Abstrakte Datentypen,  
Schnittstelle,  
Kohäsion,  
Kopplung,  
Information Hiding