

## Command Muster

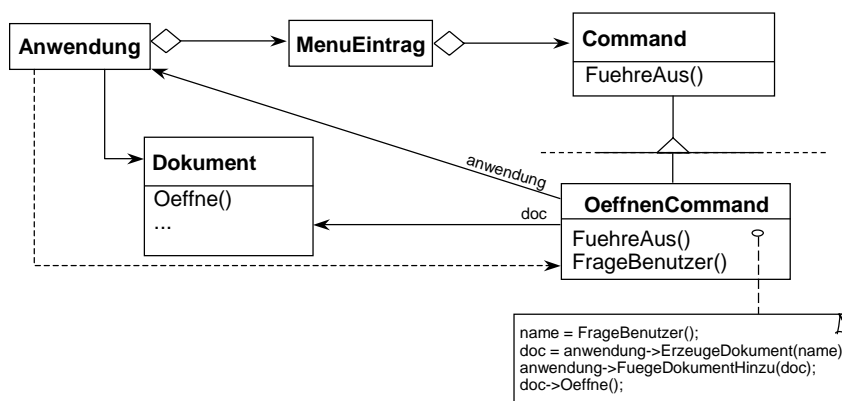
### Zweck

“Kapsle einen Befehl als ein Objekt. Dies ermöglicht es, Klienten mit verschiedenen Anfragen zu parametrisieren, Operationen in eine Queue zu stellen, ein Logbuch zu führen und Operationen rückgängig zu machen.”

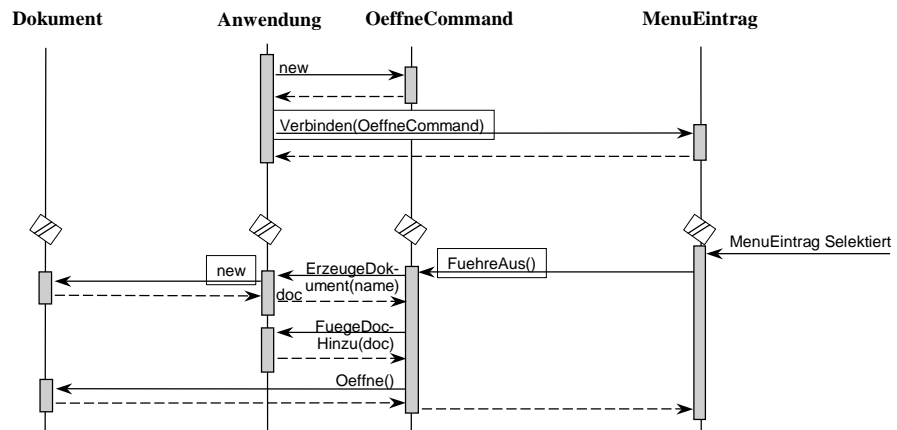
### Anwendbarkeit

- Ein Command-Objekt kann wie eine Callback-Funktion im prozeduralen Sprachen verwendet werden. Das Command stellt die objektorientierte Variante des Callbacks dar.
- Anfragen zu unterschiedlichen Zeitpunkten spezifizieren, aufreihen und ausführen. Ein Command-Objekt kann über eine von der ursprünglichen Anfrage unabhängige Lebensdauer verfügen und so zwischen Prozeßräumen transportiert werden.
- Command-Objekte erlauben das Mitprotokollieren von Änderungen und unterstützen das Rückgängigmachen von Operationen.

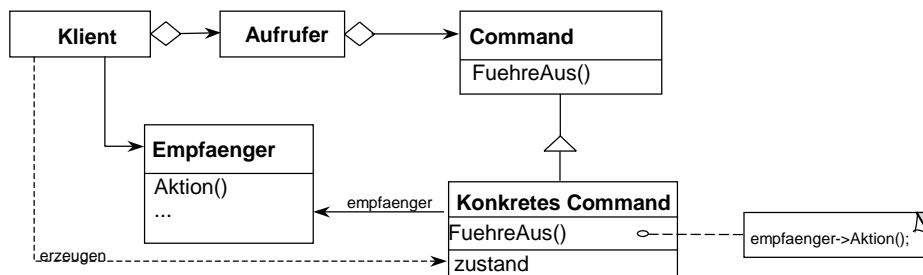
## Beispiel für das Command Muster - statisch -



## Beispiel für Command Muster - dynamisch -



## Struktur des Command Musters



## Diskussion des Command Musters

---

- Das Objekt, das die Anfrage auslöst, wird von demjenigen **entkoppelt**, das weiß wie die Anfrage umgesetzt wird.
- Soll eine Anfrage von unterschiedlichen Aufrufern ausgelöst werden, so kann jedesmal auf **dieselbe** Implementation möglicherweise sogar auf dasselbe Objekt zurückgegriffen werden.
- Es können **zusammengesetzte** Commands erstellt werden, die auf vorhandene Commands aufbauen.