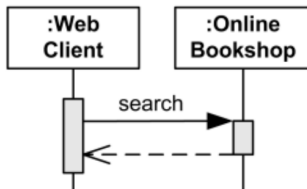


UML Sequenzdiagramme

Darstellung von Nachrichten*, die zwischen Akteuren und Objekten in begrenzten Zeitrahmen ausgetauscht werden.

→ zur Beschreibung des Systemverhaltens in Anwendungsfällen

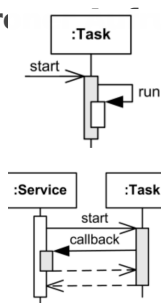


Lebenslinie – Lebenszyklus von (anonymen oder konkreten) Objekten und Akteuren

Nachricht – Darstellung einer einzelnen Kommunikation zwischen zwei Lebenslinien

Ausführung – Darstellung Zeitraums, in der Verhalten oder *Aktion ausgeführt*, ein *Signal gesendet* oder *gewartet* wird

→ **Synchroner Aufruf** – Senden Nachricht und Unterbrechen der Ausführung während *Wartezeit*



auf eine Antwort

→ **Asynchroner Aufruf** – Senden einer Nachricht und *sofortige Fortsetzung* der eigenen Ausführung

→ **Rückgabe** – Senden einer *Antwortnachricht* an das aufrufende Objekt

→ **Selbstdelegation** – Senden von Nachrichten eines Objekts an sich selbst

→ **Callback** – Ausführung, die synchron über eine Rückgabe für den originalen Aufrufer informiert

Benennung von Nachrichten: **Nummer** der Nachricht | **Name** der Methode | **Argumente** |

Rückgabotyp

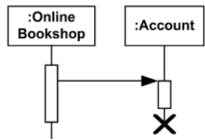
1: aMessage(num: int, value: double): String

Bedingungen, wenn Nachricht gesendet: 1: (x<0): aMessage(num: int, value: double): String



Nachrichten können **wiederholt** gesendet werden: `1: *[while(result<25)]: result=operation() |`

Instanziierung – Nachricht zur Erstellung einer Lebenslinie eines Objekts

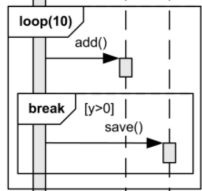


Terminierung – Nachricht zum Beenden einer Lebenslinie eines Objekts

Kombinierte Fragmente

alt : Alternativen: **Auswahl** des auszuführenden Verhaltens

opt : Optionen: **Auswahl** eines einzelnen Operanden



loop . Wiederholungen (Schleifen): **Iteration** im Ausführungsverhalten

break : **Abbrüche** oder Ausnahmen

par : **Parallele** Ausführungen

strict : Strikte Sequenzen: **Reihenfolge** einhalten

seq : Schwache Sequenzen: ausführen, **Reihenfolge** egal

Revision #5

Created 28 September 2022 07:14:20 by Merith Holtmann

Updated 28 September 2022 09:03:44 by Merith Holtmann