

Sequenzdiagramm

„Ein Sequenzdiagramm (englisch sequence diagram) ist ein Verhaltensdiagramm, welches eine Interaktion im Sinne der Unified Modeling Language (UML) grafisch darstellt. Im Rahmen der UML, einer Modellierungssprache für Software und andere Systeme, ist das Sequenzdiagramm eine von vier Interaktionsdiagrammarten.

Sequenzdiagramme beschreiben den Austausch von Nachrichten zwischen Objekten mittels Lebenslinien. [...]

Ein Sequenzdiagramm stellt in der Regel einen Weg durch einen Entscheidungsbaum innerhalb eines Systemablaufes dar.“¹

Elemente eines Sequenzdiagramms sind benannte (links) oder unbenannte (rechts) Objekte mit ihren gestrichelten Lebenslinien.

Rechts vom Doppelpunkt steht die Bezeichnung der Klasse und links vom Doppelpunkt steht bei benannten Objekten der Objektname.

Die Kommunikation zwischen den Objekten erfolgt mittels Nachrichten (Botschaften). Das Senden einer Nachricht entspricht einem Methodenaufruf.

Die Erzeugung eines Objektes erfolgt durch den Aufruf einer Konstruktormethode. Der gestrichelte Pfeil endet auf der Mitte des Rechtecks.

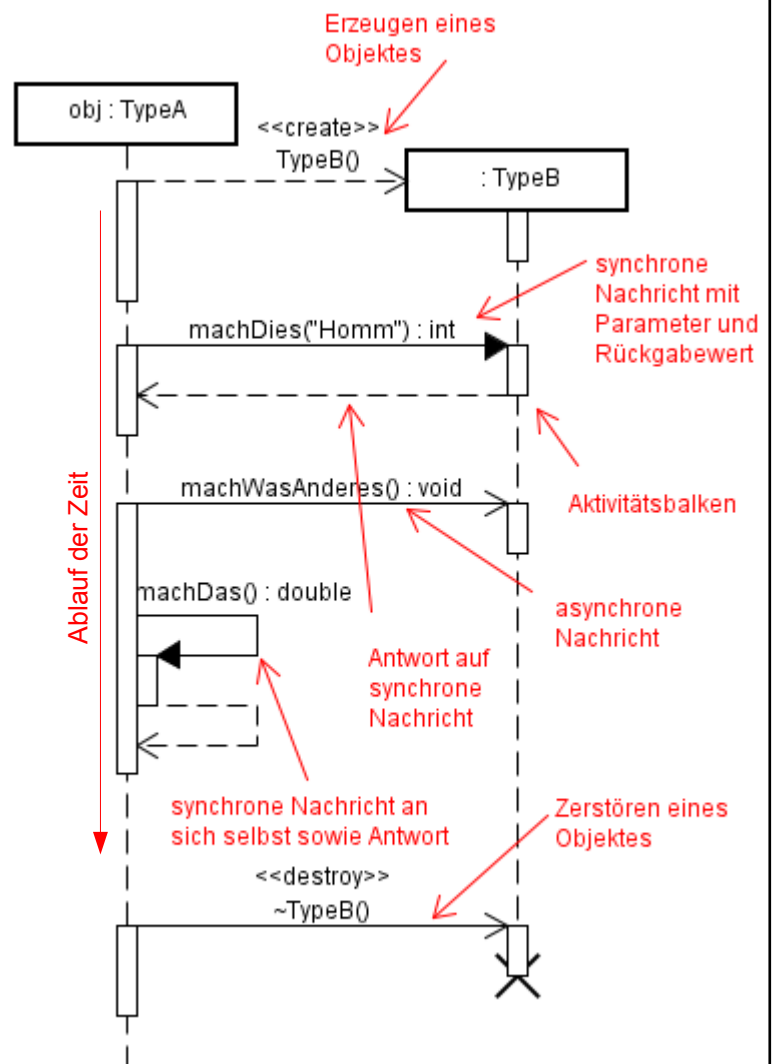
Die Konstruktormethode kann durch den Stereotyp <<create>> begleitet sein.

Es gibt synchrone und asynchrone Nachrichten, die an Objekte gesendet werden können.

Bei asynchronen Nachrichten wartet der Sender nicht, bis der Empfänger die Nachricht verarbeitet hat.

Synchrone Nachrichten werden mit durchgezogener Linie und ausgefüllter Pfeilspitze dargestellt. Asynchrone Nachrichten hingegen mit einer gestrichelten Linie und einer offenen Pfeilspitze.

Objekte können Nachrichten an sich selbst senden. Dies entspricht dem Aufruf einer klasseneigenen Methode. Diese kann u.U. auch eine Antwort zurück senden. Des Weiteren kann ein Objekt eine Nachricht senden, die das Empfänger-Objekt zerstört (Destruktor-Methode).



¹ Siehe <https://de.wikipedia.org/wiki/Sequenzdiagramm> abgerufen am 27.11.2017

Kombinierte Fragmente

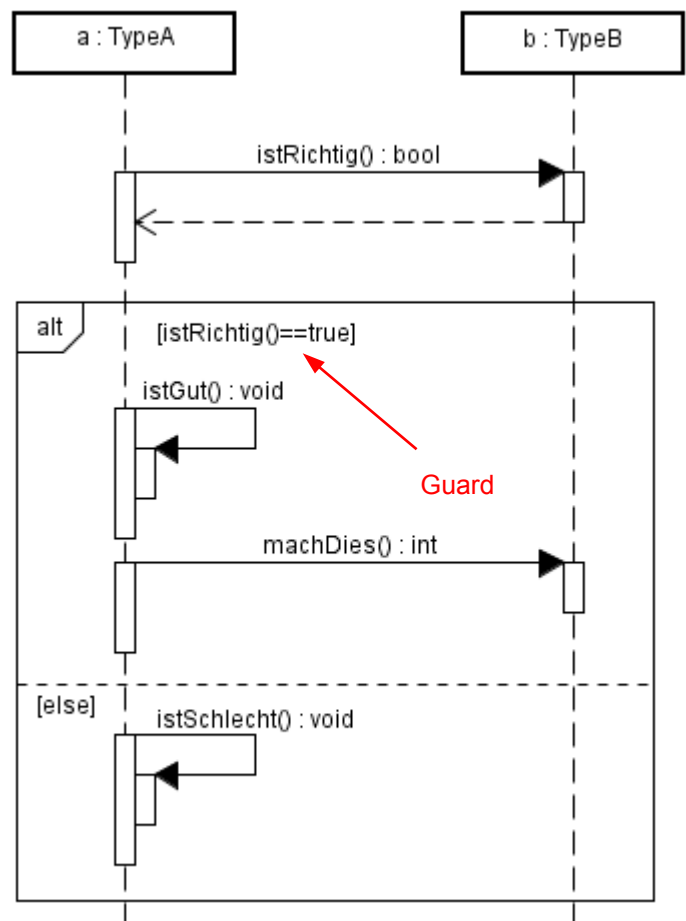
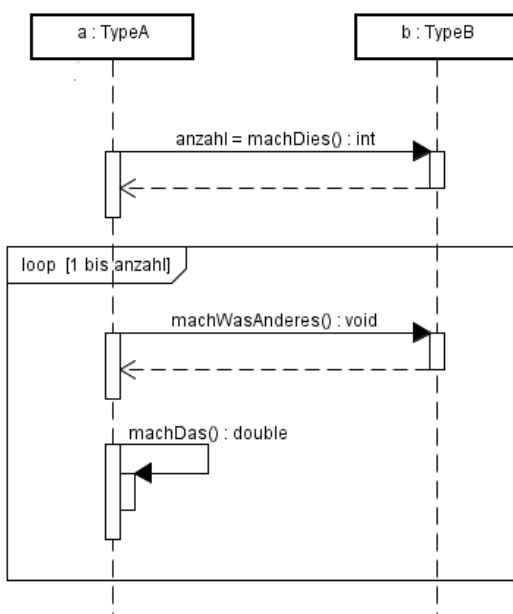
„Ein Sequenzdiagramm enthält Objekte, Lebenslinien, Aktivierungen, Nachrichten und kombinierte Fragmente (combined fragment). Mit kombinierten Fragmenten können in Sequenzdiagrammen die Kontrollstrukturen höherer Programmiersprachen ausgedrückt werden. Jedes kombinierte Fragment wird als rechteckiger Block dargestellt. In der linken, oberen Ecke trägt er eine Bezeichnung, die den Typ der Kontrollstruktur angibt (interaction operator).“²

Es gibt verschiedene Typen von kombinierten Fragmenten³.

Eines ist das *alt*-Fragment, welches eine Auswahl anhand einer Bedingung ermöglicht.

Objekt a sendet die Nachricht `istRichtig()` an Objekt b. Der Rückgabewert von b kann nun bei a ausgewertet und mittels eines kombinierten Fragments vom Typ *alt* entsprechend behandelt werden.

Die Bedingung wird in dem Guard (innerhalb der rechteckigen Klammern) angegeben.



Wiederholungsstrukturen lassen sich mit dem Typ *loop*-Fragment darstellen.

Im Guard wird entweder der Start- und Endwert bei einer Zählschleife oder bei offenen Schleifen die Bedingung angegeben; z.B. `[solange i < 100]`.

Ein Sequenzdiagramm basiert zumeist auf einem Anwendungsfall und stellt die Interaktion der jeweils beteiligten Objekte dar. Zwingend erforderlich ist dies jedoch nicht!

² Siehe <https://www.fbi.h-da.de/labore/case/uml/sequenzdiagramm.html> abgerufen am 27.11.2017

³ Siehe ergänzend hierzu https://wiki.thm.de/Sequenzdiagramm#Kombinierte_Fragmente abgerufen am 27.11.2017 und <https://www.uml-diagrams.org/sequence-diagrams-combined-fragment.html> abgerufen am 28.10.2018

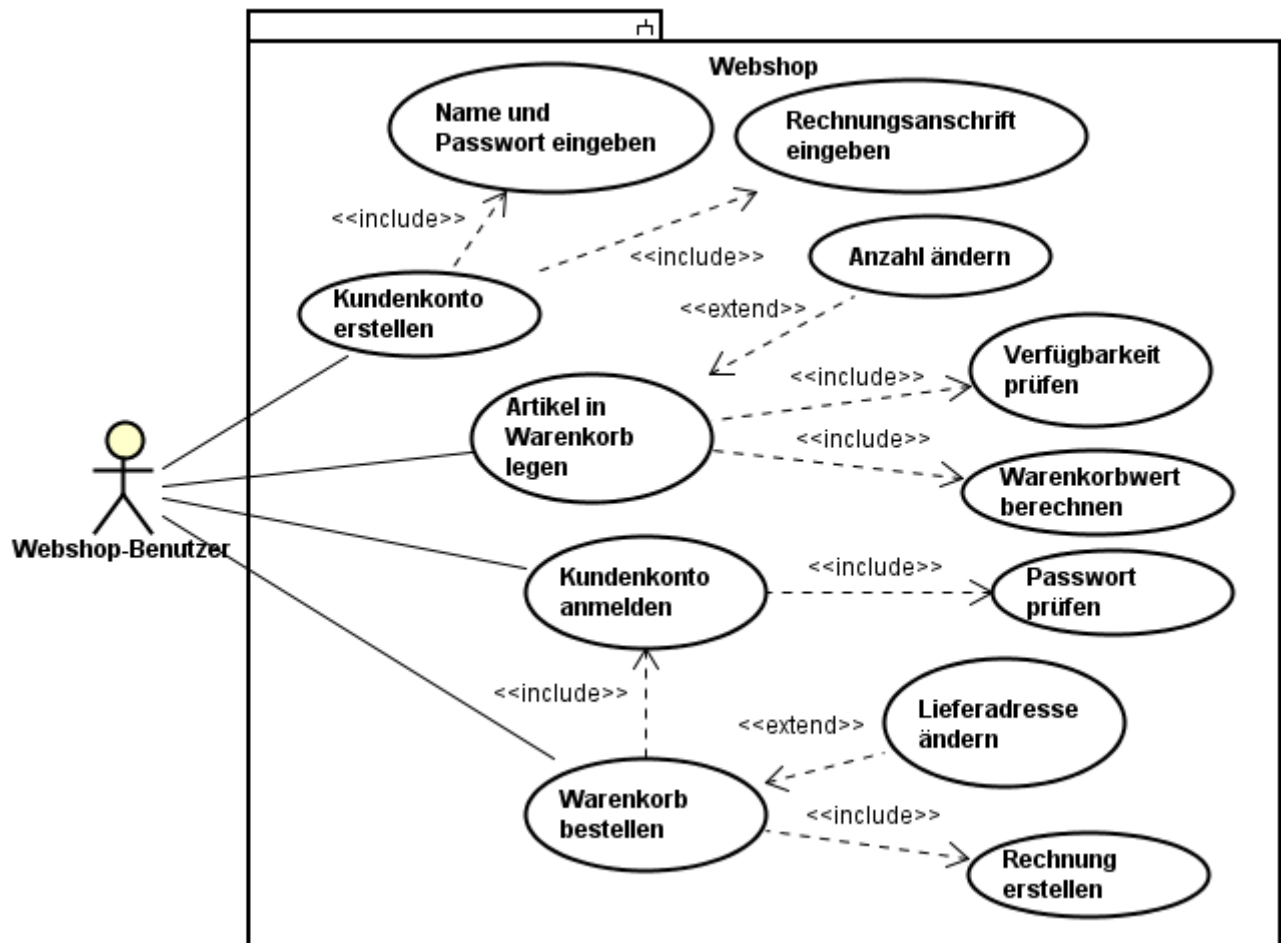
Beispiele für Sequenzdiagramme

Ein Webshop soll es möglich machen, dass ein Webshop-Benutzer für sich ein Kundenkonto erstellen kann. Hierfür wird ein Kontoname und ein Passwort abgefragt. Anschließend wird in der Liste mit allen bereits existierenden Kundenkonten überprüft, ob ein Kundenkonto mit dem gewählten Kontonamen bereits existiert. Falls nicht, muss der Webshop-Benutzer noch seine Rechnungsadresse angeben, die dann auch automatisch zur Lieferadresse wird. Im Anschluss wird das Kundenkonto der Liste mit allen Konten hinzugefügt.

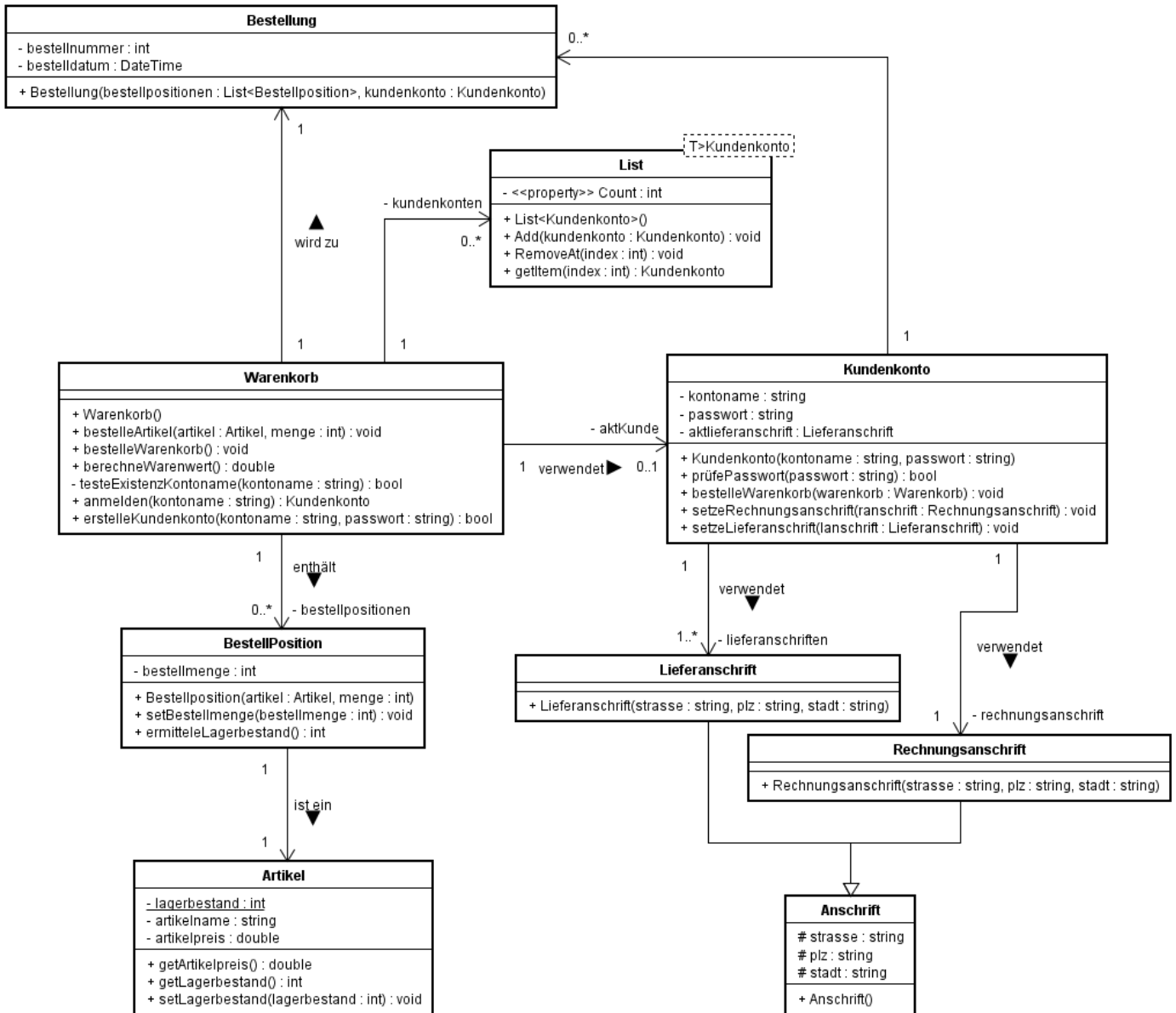
Jeder Webshop-Benutzer, ob angemeldet oder nicht angemeldet, verfügt über einen Warenkorb, in dem er gewünschte Artikel ablegen kann. Die Artikelanzahl ist immer erst „1“, kann aber nachträglich verändert werden. In jedem Fall wird überprüft, ob der Artikel in der gewünschten Anzahl zur Verfügung steht. Jeder bestellte Artikel wird zu einer Bestellposition. Nach Ablage eines Artikels im Warenkorb oder nach einer Mengenänderung, wird dessen Wert neu berechnet.

Der Webshop-Benutzer kann den Inhalt des Warenkorbes mit all seinen Bestellpositionen bestellen. Hierzu muss er sich mit seinem Kundenkonto anmelden, falls dies noch nicht erfolgt ist.

Im Rahmen des Bestellvorgangs kann er dann die Lieferadresse ändern bzw. eine neue Lieferadresse erstellen. Für jede abgeschlossene Bestellung wird automatisch eine Rechnung erstellt.



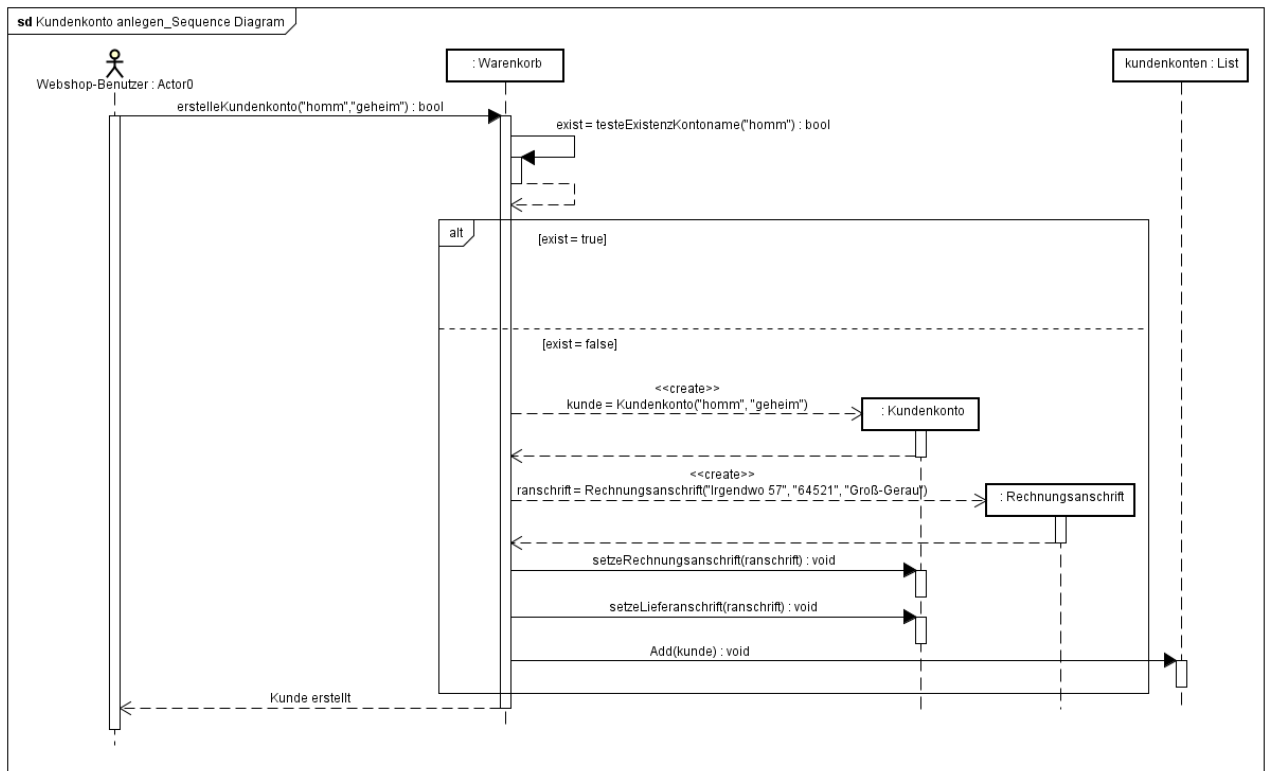
Zu dem Webshop wurde das nachfolgende (unvollständige) Klassendiagramm entwickelt:



Zentrales Element dieses Klassendiagramms ist die Klasse Warenkorb. Jeder Webshop-Benutzer erhält ein Objekt dieser Klasse. Von hier ausgehend sind alle Anwendungsfälle realisiert.

Anwendungsfall „Kundenkonto erstellen“

Der Webshop-Benutzer übermittelt den gewünschten Kontonamen („homm“) und das Passwort („geheim“). Das Warenkorb-Objekt überprüft, ob der Kontoname bereits existiert. Falls nicht, wird das neue Kundenkonto-Objekt erstellt und anschließend der Liste kundenkonten hinzugefügt.



Anwendungsfall „Artikel in Warenkorb legen“

Der Webshop-Benutzer betrachtet die Beschreibung eines Artikels und legt diesen Artikel in den Warenkorb. Hierzu wird eine Bestellposition erzeugt. Anschließend wird für diese Bestellposition der Lagerbestand des betreffenden Artikels ermittelt und abschließend der Wert des Warenkorbes neu ermittelt.

