

Übungsblatt 4

Lösungsvorschlag
13.07.2014

Aufgabe 1)

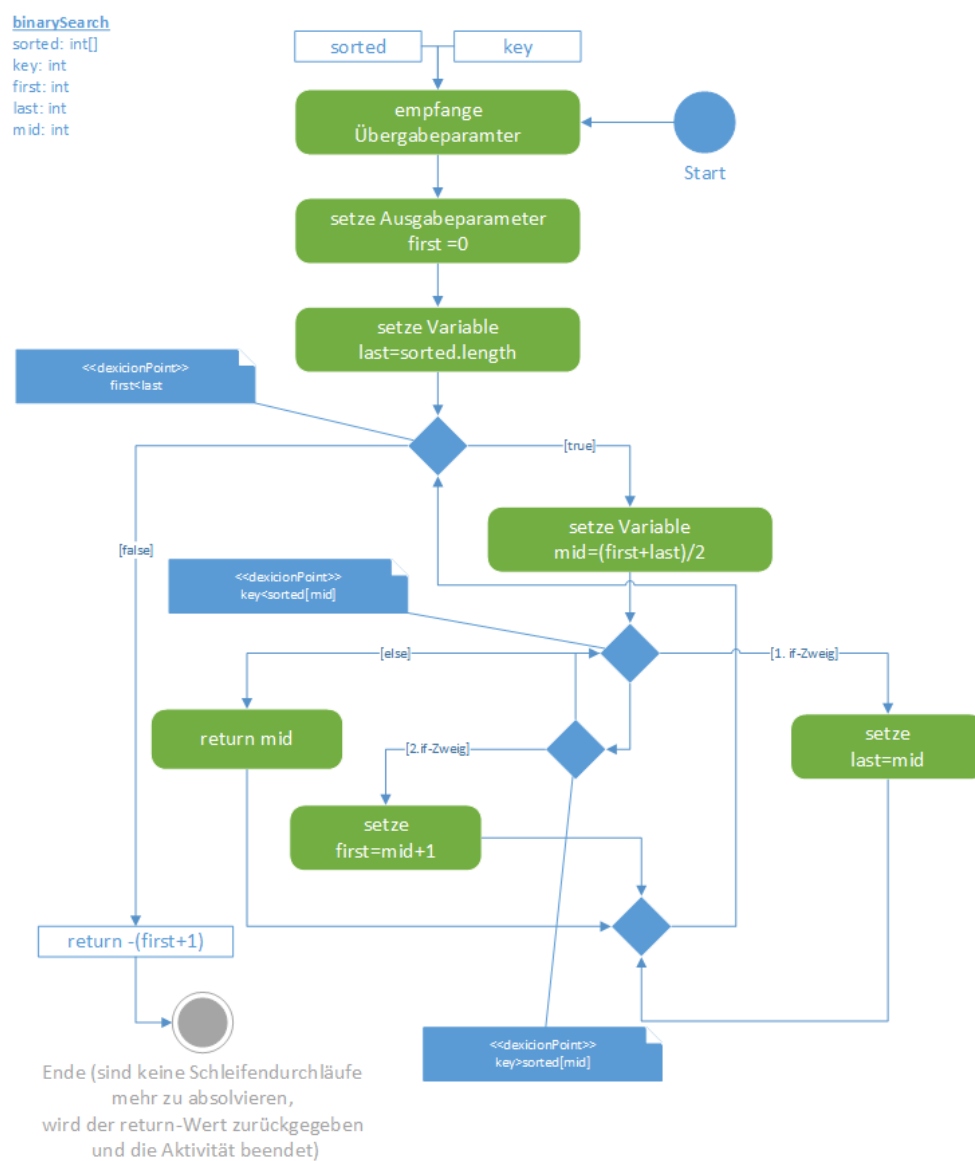


Abbildung 1: Kontrollfluss der Java-Methode `binarySearch` in einem Aktivitätsdiagramm dargestellt

Aufgabe 2)

a.

In diesem Aktivitätsdiagramm wird - von der Bestellung bis zum Servieren - der Zubereitungs-
vorgang eines Kaffeeautomaten veranschaulicht.

Der Ablauf des eigentlichen Prozesses gestaltet sich derart, als dass eine Bestellung beim Au-
tomaten eingeht (*Take order*) und vonseiten des Automaten darauf geprüft wird, ob Tee oder
Milchkaffee bestellt wurde.

Bei einer Teebestellung gestaltet sich der Restverlauf des Diagramms simpel, die Zubereitung
wird anschließend unter einem Punkt zusammengefasst (*Make Tea*) und zuletzt wird dann das
bestellte Getränk am Schlusspunkt des Aktivitätsdiagrammes serviert (*Serve Drink*).

Beim Milchkaffee gestaltet sich der Prozess nur dahingehend komplizierter, als dass die Zuberei-
tung zunächst in die Herstellung des Kaffees (*Make Coffee*) und das Aufschäumen der Milch
(*Steam Milk*) aufgeteilt wird, bevor diese Aktion im Hinzufügen der Milch zum Kaffee zusam-
menlaufen (*Add Milk to Coffee*) und zuletzt auch zum Servieren als Schlusspunkt führen.

Das Diagramm selbst weist im Bezug auf gängige Normen für Aktivitätsdiagramme Mängel
auf, so wird beispielsweise die Grundregel von einem Aktionszustand mit nur einer Eingangs-
aktion dadurch verletzt, dass bei der Aufteilung der Nebenprozesse der Milchkaffeeherstellung
der verwendete Fork Knoten nicht wieder zusammengeführt wird, was für sich bereits fehlerhaft
ist und zu zwei Eingangsaktionen beim Hinzufügen der Milch zum Kaffee führt. Diese Regel
wird auch beim letzten Aktionszustand wieder verletzt, da die eigentliche Diagrammaufteilung
in zwei Pfade durch jenen Bedingungsknoten, welcher die Art Bestellung prüft, hier nicht durch
einen Merge-Knoten wieder zusammengeführt wurde.

Dem gängigen Formalismus folgend ist es weiterhin ein Fehler, dass im Diagramm weder Start-
knoten noch Endknoten vorhanden sind. Das Diagramm beginnt und endet jeweils mit einem
Aktionszustand.

b.

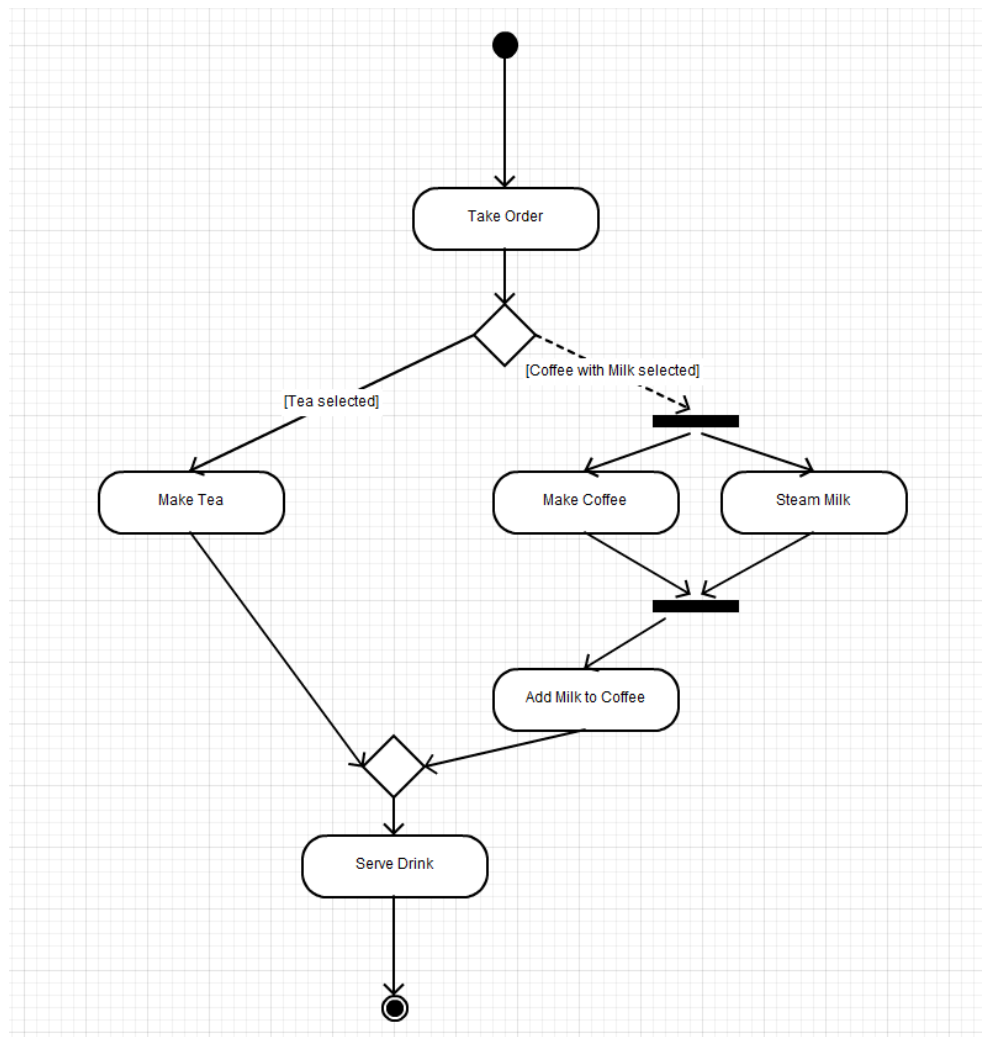
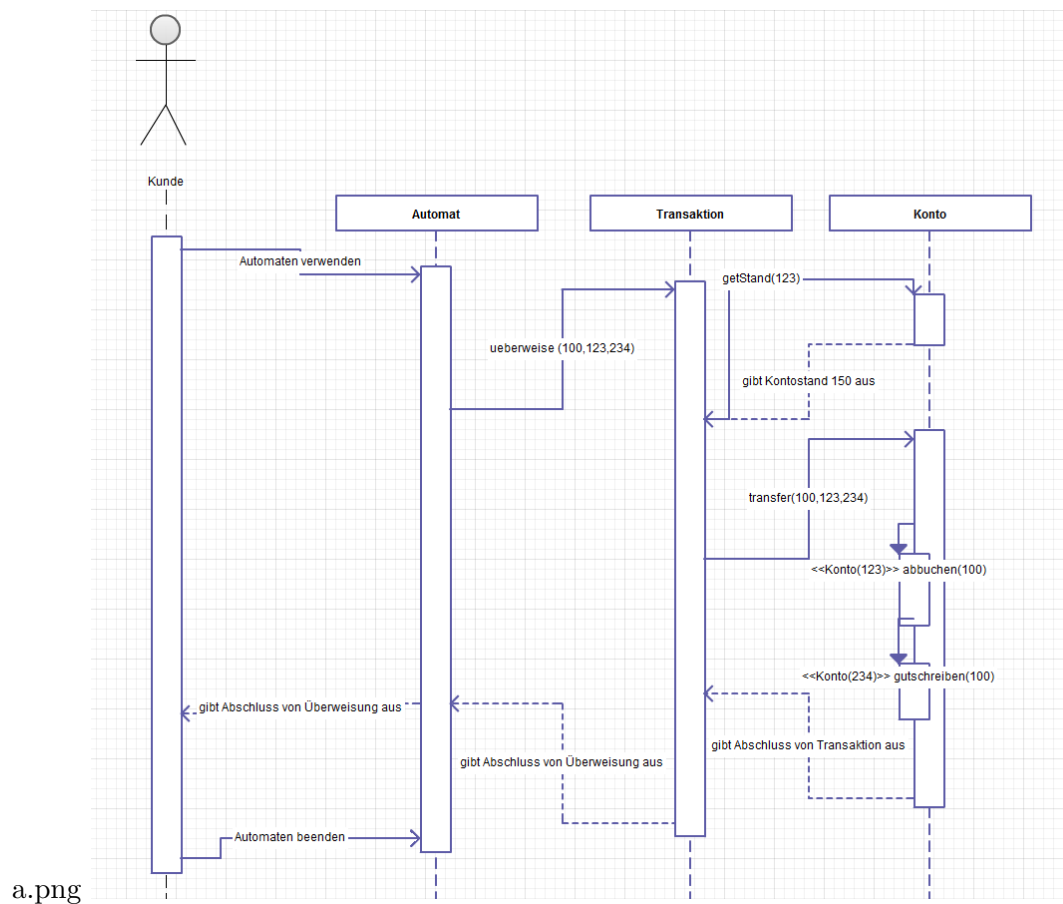


Abbildung 2: Korrigiertes Aktivitätsdiagramm

Aufgabe 3)

a.



a.png

Abbildung 3: Sequenzdiagramm 1 mit gedecktem Konto

b.

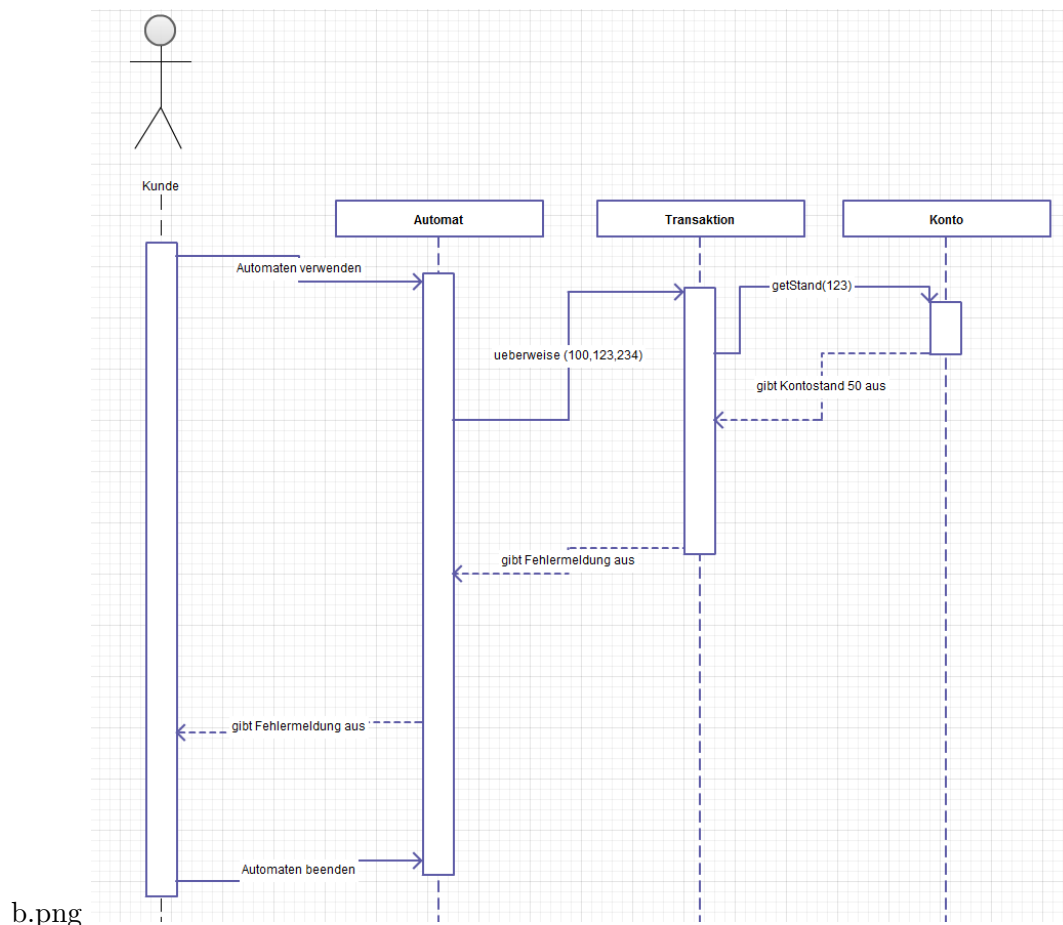


Abbildung 4: Sequenzdiagramm 2 mit ungedecktem Konto

Aufgabe 4)

Dargestellt ist der Ablauf in einem Online-Shop, visualisiert in einem Sequenzdiagramm. Dabei regelt die Klasse **ShoppingSession** die Shop-Sitzung und ist ebenso für die Darstellung des Shops zuständig. Die Klasse **Cart** stellt einen Einkaufswagen dar und die Klasse **ProductDB** repräsentiert den Zugriff auf die Produktdatenbank.

Beschreibung des abgebildeten Vorgangs:

Der Kunde ruft zunächst durch entsprechende Eingaben den Online-Shop auf. Es wird dabei eine Shop-Sitzung durch die Klasse **ShoppingSession** erstellt. Der Kunde sucht ein Notebook. Dazu gibt er in einem Suchfeld das Stichwort „Notebook“ ein. Die **ShoppingSession** verarbeitet die Suchanfrage des Kunden und überprüft mit ihrer Methode **searchProducts** die Produktdatenbank **ProductDB** auf Treffer.

Die **ProductDB** liefert der **ShoppingSession** eine Liste mit Produkten zurück, die dem Kunden dann, wahrscheinlich in ansprechender Form (für die Darstellung ist auch die Klasse **ShoppingSession** zuständig) umgehend angezeigt wird.

der Kunde möchte sich Produktdetails von *X123* anzeigen lassen. Die **ShoppingSession** ruft ihre Methode **getProductDetails** mit dem entsprechenden Parameter auf und fragt die Datenbank danach ab.

Die Datenbank liefert wiederum eine Liste mit Details zurück, die wieder in ansprechender Form dem Kunden von der **ShoppingSession** angezeigt werden.

Der Kunde entschließt sich dazu, das Gerät **X123** zu kaufen und legt es durch entsprechende Eingaben in den Warenkorb. Letzterer wird durch die Klasse **Cart** modelliert. Der Konstruktor der Klasse wird aufgerufen und mittels der **add**-Methode und dem entsprechenden Parameter wird das Gerät tatsächlich in den virtuellen Warenkorb bewegt, da die Methode **true** zurückliefert. Dies wird an die **ShoppingSession** weitergeleitet, die dann diese Information in verständlicher Form an den Kunden weiterleitet.

Zum Schluss überlegt es sich der Kunde doch noch mal anders und bricht seinen Einkauf ab, indem er durch entsprechende Eingaben (z.B. Löschen des Warenkorbs oder Ausloggen) der **ShoppingSession** den Abbruch mitteilt. Diese ruft die **clearProducts**-Methode auf und der entsprechende Teil der Datenbank wird geleert bzw. gelöscht. Schließlich liefert die Methode **true** zurück und die Aktion wurde erfolgreich abgebrochen. Der Warenkorb dürfte somit wieder leer sein.

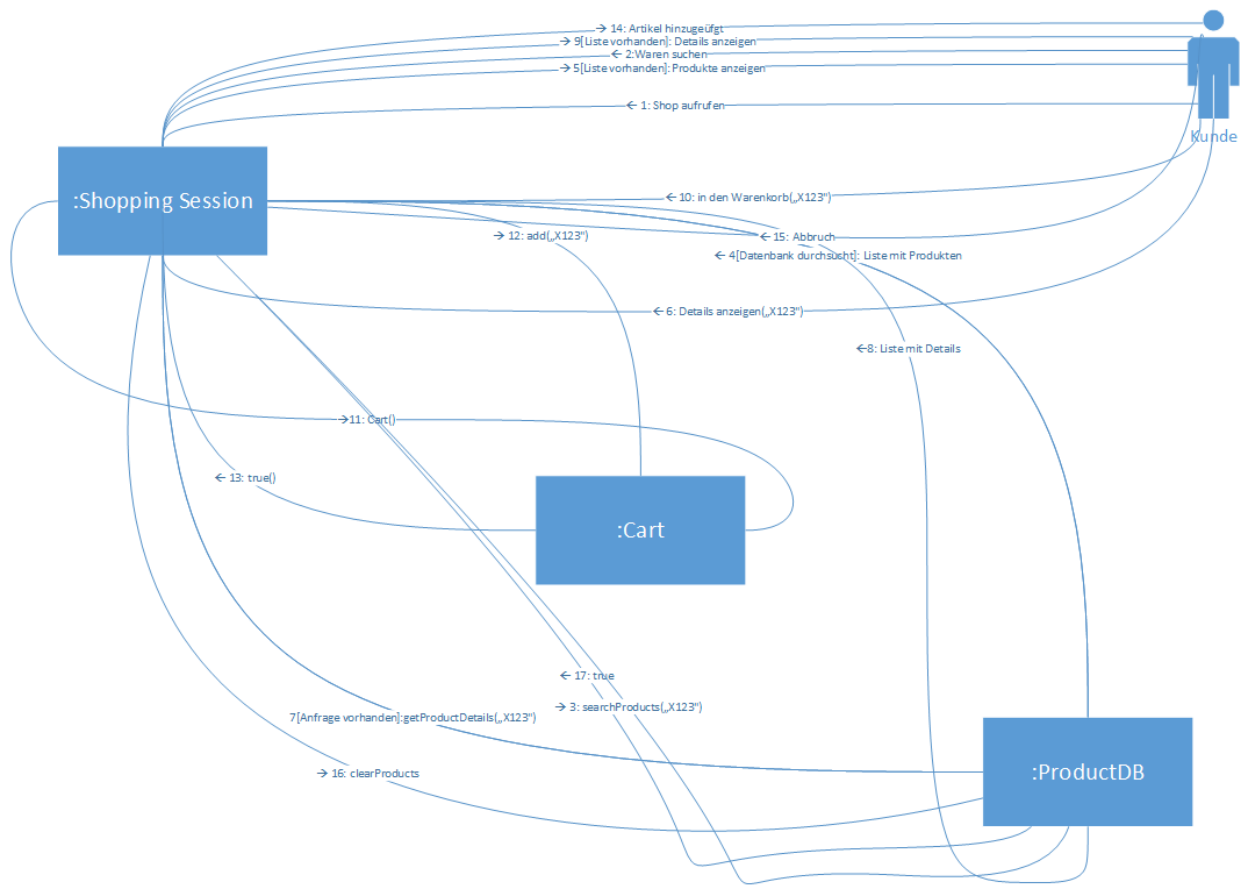
Äquivalentes Kommunikationsdiagramm:

Abbildung 5: Äquivalentes Kommunikationsdiagramm zum gegebenen Sequenzdiagramm