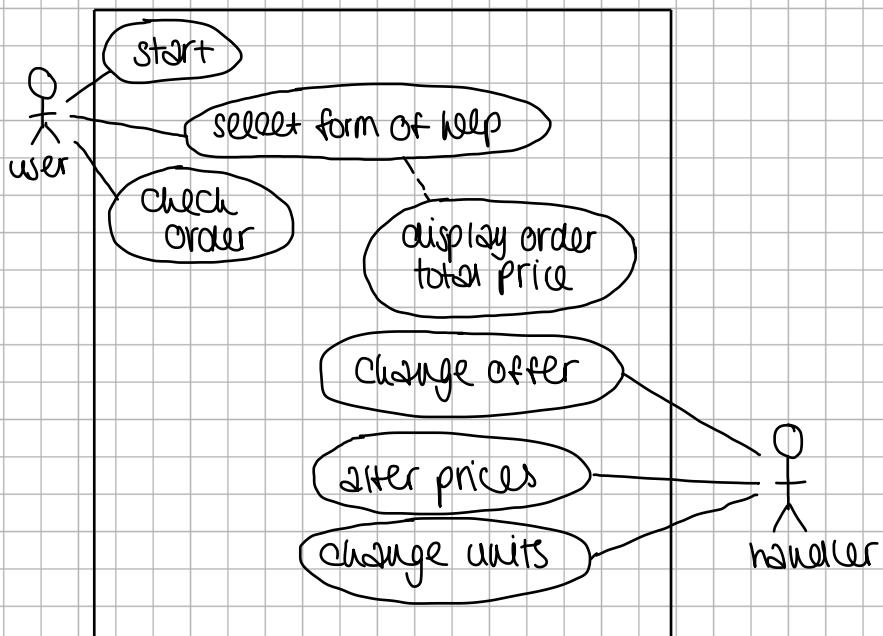


HOME HELPER: DATA STRUCTURES

category	name: string	price: number	unit: string	'value: string
Einkauf (Supermarkt)	Butter Milch Zitrone	0,79 1,20	Stück Liter	Butter Milch
Haushalt	Wanne Wanne	10,00 10,00		Rasen mähen Hund ausführen
Bank	bank bank	3,00 3,00		Geld abheben Geld einzahlen
Zahlung	Cash Cash			Bar Banküberweisung

USE-CASE



UI - Scribble

<h1>

<p>

HOME HELPER

Wählen Sie auf der linken Seite die Einkäufe und rechts sehen Sie die Aufstellung mit dem jeweiligen Preis und dem Endpreis.

<div>
id = "form" > change

<h2>

<div>
id = "order" >

input
type =
"text"
name =
"Liste"
dynamic

Wähle hier

<input type =
"text" name =
"Dekost supermarket"
number name =
"range" dynamic

Einkaufen

Wähle einen Artikel ▾

0 Wählte einen Super-
markt

<field>
set >

Haushaltarbeiten

Rasen mähen
 Wischen / Staubsaugen
 Hund ausführen
 Pflanzen gießen

Bank

abheben
 einzahlen

0€ 500€

Zahlungsweise

Bar Bankübertragung Überweisung

hier Bestellung

Reset

Order

<input>
type = "checkbox"
name = "home"
dynamic

<input>
type = "radio"
name = "bank"
dynamic

<input>
type = "radio"
name = "cash"
dynamic

<input>
type = "range"
name = "amount"
min = "0"
max = "500"
step = "1" input

id = "list" >
<button>
type = "submit" > click

A0

install load
listener

> load

change
form-div

input
slider

<button>
type = "reset"

handleLoad

B → generateContent

find form-div &
install change-listener

displayAmount
event: Event
amount: number
= target.value

progress.value = amount

final slider & insta
input - listener

A

displayOrder

delete previous
order

data = getDataFromForm()

printSummary

[data available]

item = getNextItemData
price = getItemPrice
unit = getItemUnit
portion = get("anzahl")

order.innerHTML = entry[1]
+ price * portion + unit;

order.innerHTML += portion;

order.innerHTML += entry[1]

order.innerHTML += entry[1]
+ price

" "

order.innerHTML += entry[1]

" "

printRow (item; price; unit;
portion)

[item == DataList]

[item == Anzahl]

[item == ParistSupermarkt]

[item == Home]

[item == Bank]

[item == Amount]

[item == Cash]

AD generateContent

item

name: string
price: number
unit: string
value: string

Data

[category: string] : Item[]

generateContent

- data : Data

[category available]

category: string = next key
items: Item[] = nextItemList

appendChild
to fieldset

fieldset = getFieldset
with id = category

[category is Einzelart]

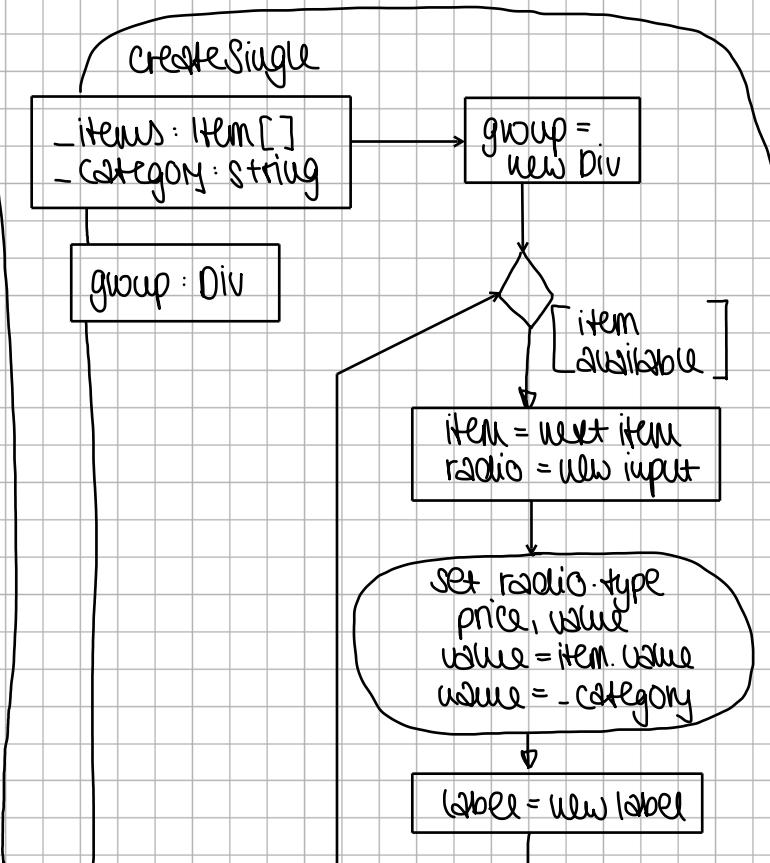
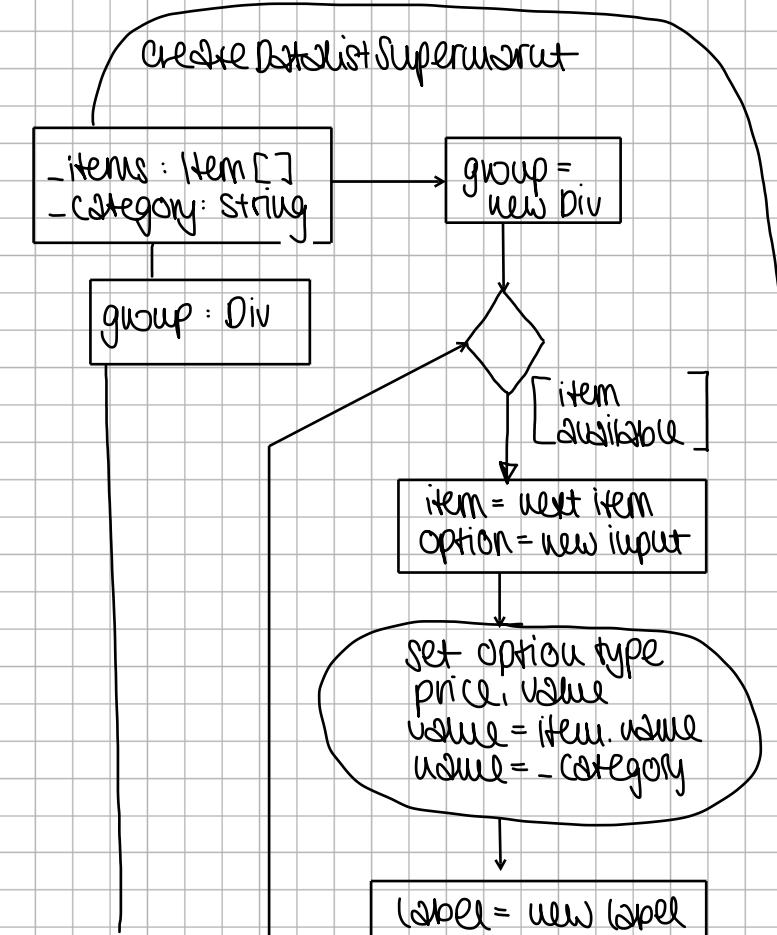
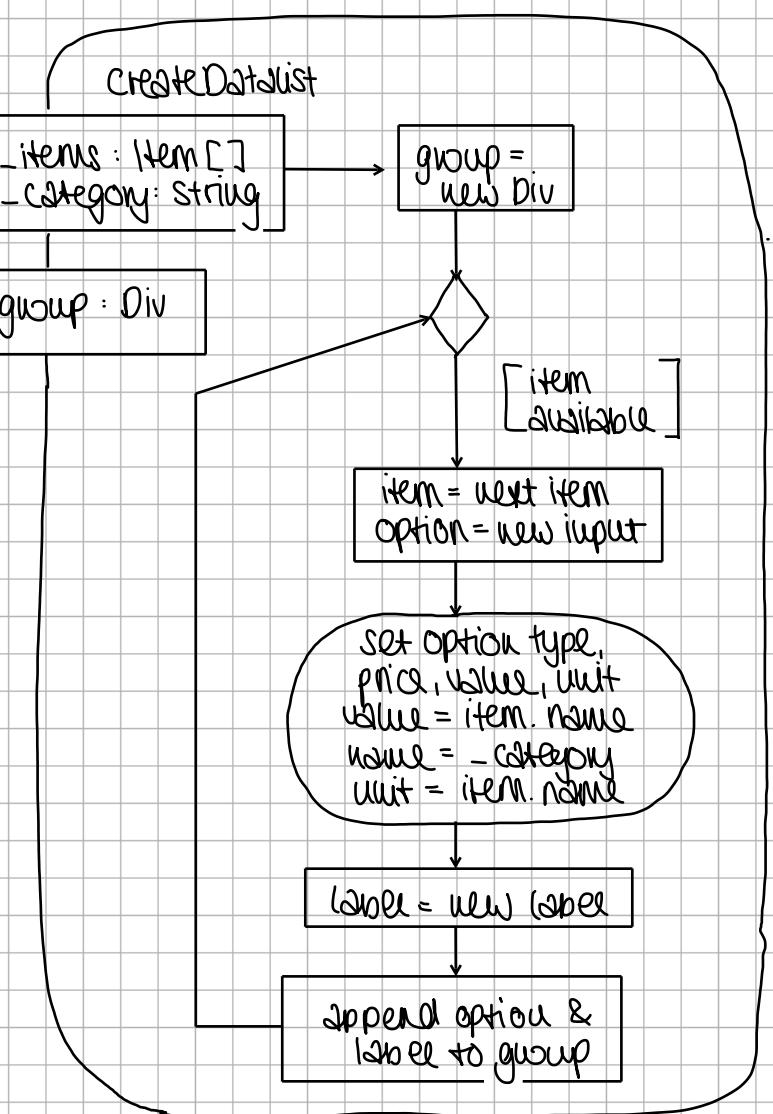
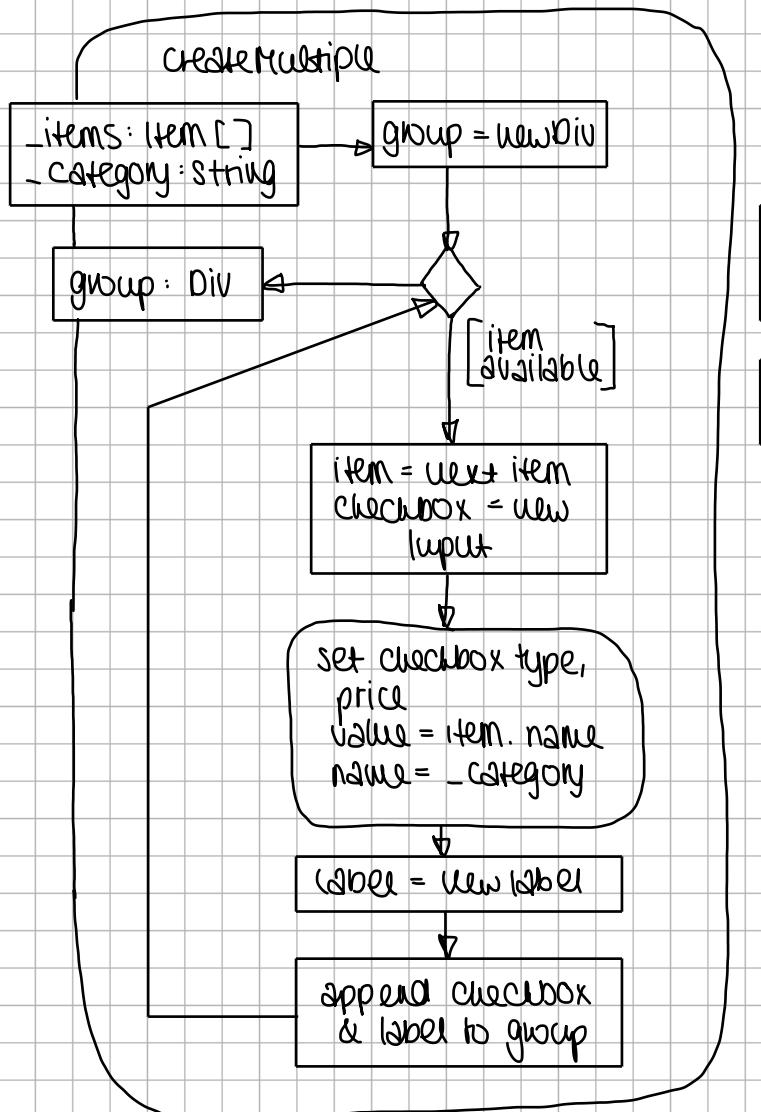
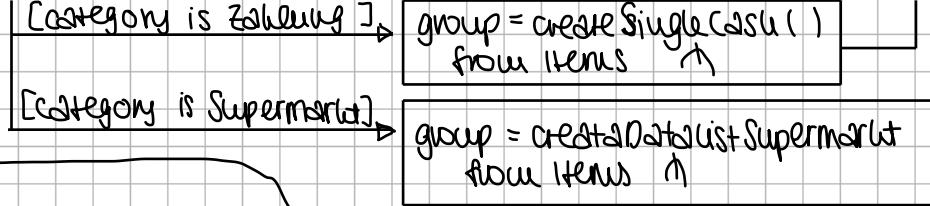
group = CreateDataList()
from Items

[category is Haushalt]

group = CreateMultiple()
from Items

[category is Bank]

group = CreateSingle()
from Items



append option &
label to group

append radio
& label to group

createSingleCash

- items: Item[]
- category: string

group =
new Div

group : Div

item
available
item = next item
radio = new input

set radio type
price, value
value = item.value
value = -category

label = new label

append radio
& label to group

AD Client

handleLoad

A
find submit button
install click listener
→ order

order

Send Order

sendOrder

order =
get dataflowform

query = create
query from order

send query to
server

wait for response

notify user
about request send

handleLoad

request offer - data

wait for response

response

extract offer from
response

wait for offer

offer

data = parsed offer
(JSON)

B