

Sorte wählen

Topping wählen

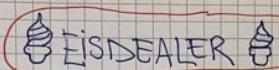
Cream wählen

Streusel wählen

Behälter wählen

Lieferdienst wählen

Persönliche Daten  
angeben



statisch

Eis							
<input type="radio"/> ①	Vanille						
<input type="checkbox"/> ②	Schokolade						
<input type="radio"/> ③	Erdbeere						
<input type="radio"/> ④	Pistazie						
<input type="radio"/> ⑤	Mango						
<input type="radio"/> ⑥	Zitrone						
<input type="radio"/> ⑦	Dealer's Special						
Topping							
<input type="checkbox"/> Sahne	<input type="checkbox"/> Marshmallows	<input type="checkbox"/> Nüsse	<input type="checkbox"/> Oreo				
Cream							
<input type="radio"/> Schokolade	<input type="radio"/> Karamell	<input type="radio"/> Erdbeere	<input type="radio"/> Dealer's Special				
Streusel							
<input type="checkbox"/> Bunte	<input type="checkbox"/> Schokolade	<input type="checkbox"/> Fruity	<input type="checkbox"/> Glitzer	<input type="checkbox"/> Kokos			
Behälter							
<input type="radio"/> Waffel	<input type="radio"/> Becher	<input type="radio"/> Kübel					
Lieferdienst							
<input type="radio"/> UPS	<input type="radio"/> DHL	<input type="radio"/> Hermes	<input type="radio"/> DPD				
Lieferart							
<input type="radio"/> Standard	<input type="radio"/> 24h-Express	<input type="radio"/> Morning-Express					

dynamisch

Bestellübersicht:

1 x Eis (Typ: Vanille) à 2€  
 + Topping 'Nüsse' 2€  
 + Cream 'Karamell' 1,50€  
 + Streusel 'Glitzer' 0,70€  
 Behälter : Becher 1€  
 Versand mit DPD 4,50€  
 zzgl. 24h Express - zuschlag 5€  
 Gesamtpreis: 16,70€

Deine Lieferadresse

dynamisch

Personliche Angaben

Nachname

Vorname

straße

Nr.

PLZ

Ort

(Mailadresse)

statisch

## Objekt Struktur

interface "Produkt" {

    name: string

    preis: number

}

    ] interface für  
    einzelnnes Produkt

interface "AlleProdukte" {

    [ key: string ]: "Produkt" [ ]

}

    ] interface für  
    Liste von Produkten

let "Produktliste": "AlleProdukte" = {

    "Produktart": [

        { name: string, Preis: number },

        { }

    ],

    "Produktart2": [ "████████████████" ]

        { name: string, Preis: number }

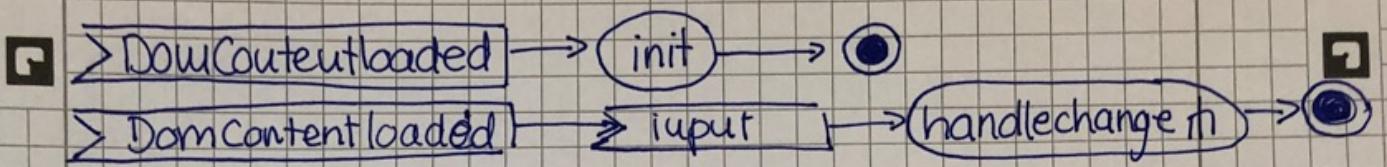
        { }

    ].

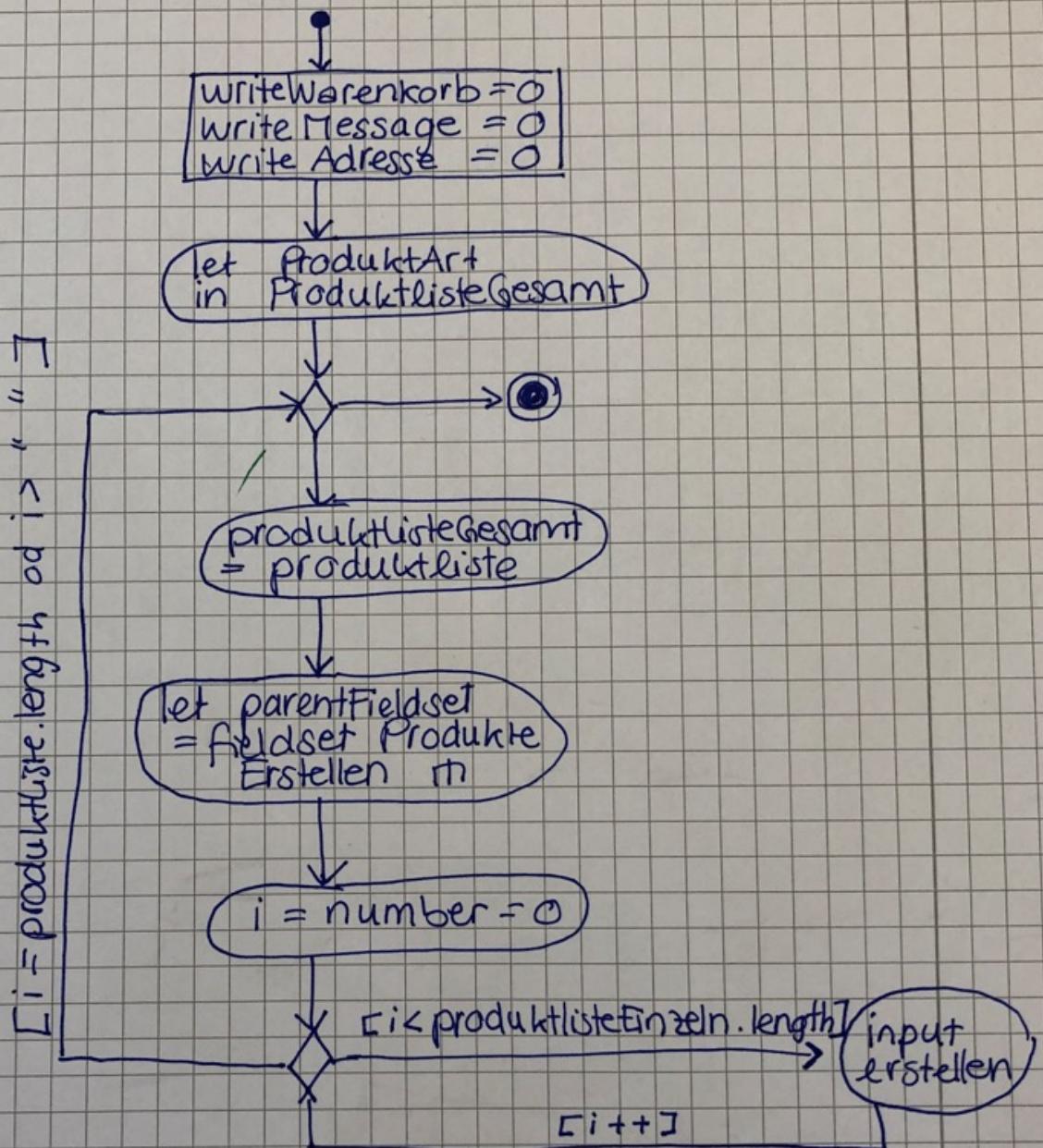
};

array mit  
Listen für  
Produkte

> hier kann man  
dann auf alle  
Produkte +  
Ergebnissen  
zu greifen !!



init



fieldset erstellen  
Produkte : HTMLFieldsetElement

-parent : HTMLDivElement  
-produkt : string

Create Element  
"Fieldset"

Create Element  
"Legend"

fieldsetid zuordnen  
Attribut zuordnen

legend.innertext =  
- produkt

legend in fieldset  
& fieldset in  
parent appenden

return fieldset

"Eis"  
"Topping"

inputErstellen  
void

- parent: HTMLFieldsetElement  
- produktEinzeln: produkt  
- produktArt: string

create Element  
"input"

input.type =  
- produktEinzeln.inputType

id zuordnen

let Label: HTMLElement

[else]

[inputType = number]

inputValue  
= produktname

zuordnen  
inputName

CreateLabel

min  
max  
step zuordnen

label = create<p>

label "for"  
- input id

label innerHTML  
= produktname

(A)

vanille, wort  
schokolade  
auf der  
seite anzeigen

A

attribute  
setzen

"produkt"  
"sorte"  
"preis"

attribut  
setzen

"show"  
"false"

ihput in  
-parent &  
label in  
-parent  
laden



paragraph Erstellen

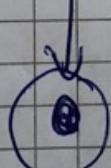
-parent : HTMLDivElement :  
HTMLParagraphElement

Paragraph : HTMLParagraphElement  
createElement "p"

Set Attribute  
"location", "Warenkorb"

-parent.appendChild (paragraph)

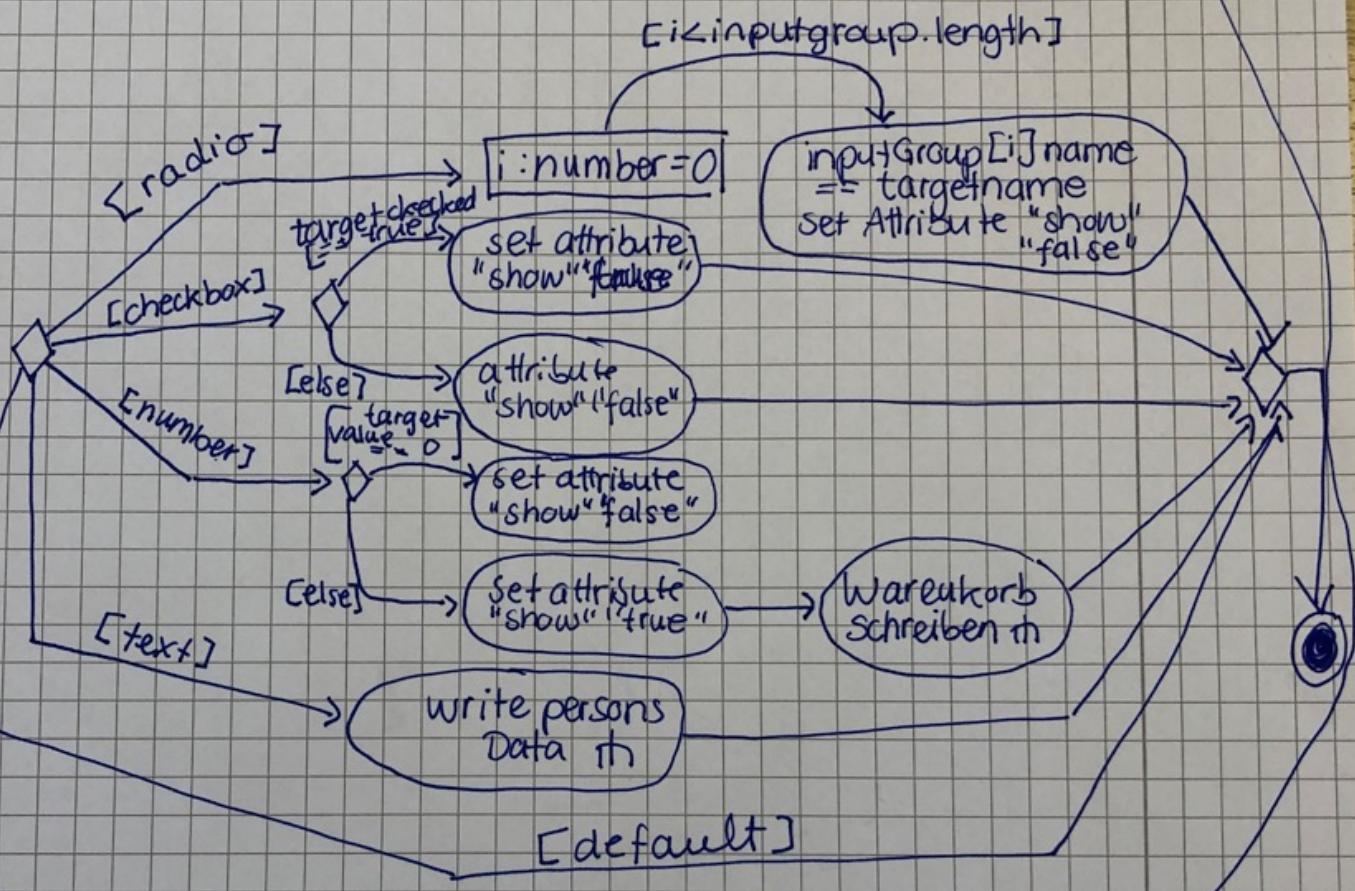
return paragraph



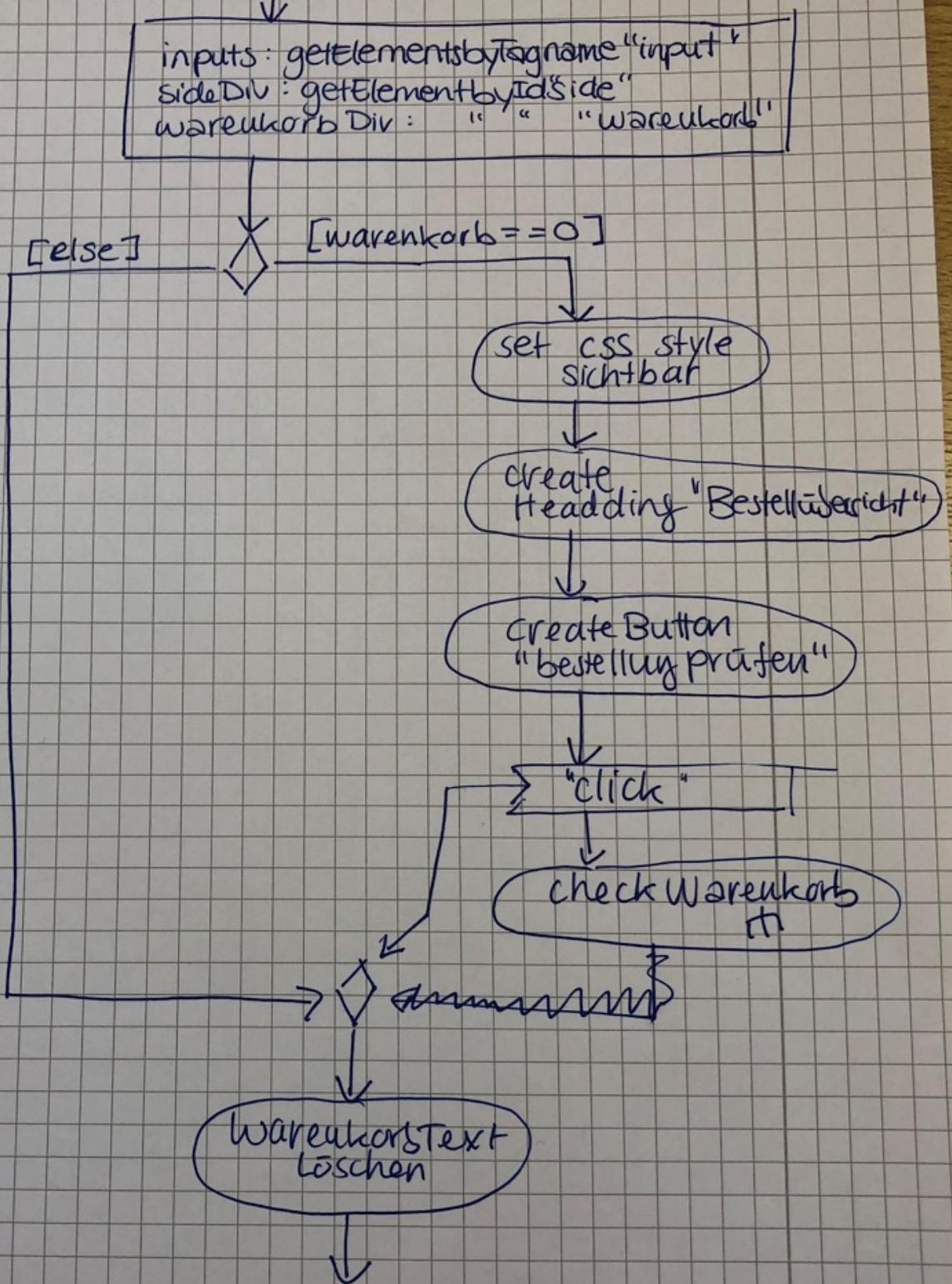
```

handlechange
    -event : Event
    target = -eventtarget
    inputGroup = getElementbyTagName

```



## Warenkorb Schreiben



preisBerechnen

- basketDiv: HTML(DIV)lement

preis: number = 0

basketPS = getElementsByTagName "P"

Einzelpreis

i: number = 0; i < basketPS.length; i++

Stück  
zahl

[else]

[basketPS[i].getAttribute("calc" == true)]

singlePreis: number = parseFloat  
get Attribute "Preis"

value: number = Number  
basketPS[i].getAttribute  
"value"

preis = (singlePreis \* value)

Einzelpreis  
x Stück  
zahl

preisString: String = pres.toString() + "€"  
return preisString

checkwarenkorb  
: void

- event : Event

messageBestellng : string " ",  
message Perso : string " ",

MessageBestellng += Checkbestellng() th

MessagePerso += Checkperso() th

[ writeMessage != 0 ] ↓ [ writeMessage == 0 ] ↓

inhalt von  
P löschen

Create 'P'

inner  
Text für  
<P>

[ messageB == 0 ]  
[ messageB != 0 ]  
[ messageP == 0 ]  
[ messageP != 0 ]

"Alle Angaben"  
korrekt

innertext  
message  
Perso

P innertext  
messageB +  
messageP

innertext  
message  
Bestellng

X

writeMessage  
++



Übersicht erstellen

-event·Event-

produkt : getElementById "Warenkorb"  
"P"

let kundenantrag  
h: stry = "0"

h += Bestellübersicht

h += "\n"

+xt: stry "

i : number = 0; i < produkt.length - 1; i++

+xt += produkt[i].innerText + "\n"

endpreis : stry = ""

endpreis += produkt.length - 1

endpreis += "\n"

nachricht : stry = h + xt + endpreis

window.alert(nachricht)