

Aufgabe 5 - Aktivitätsdiagramm

.addEventListener ("DOMContentLoaded", init) !!

init

-event: Event

function writeHTML () !!

function changeListener (event) !!

changeListener

-event: Event

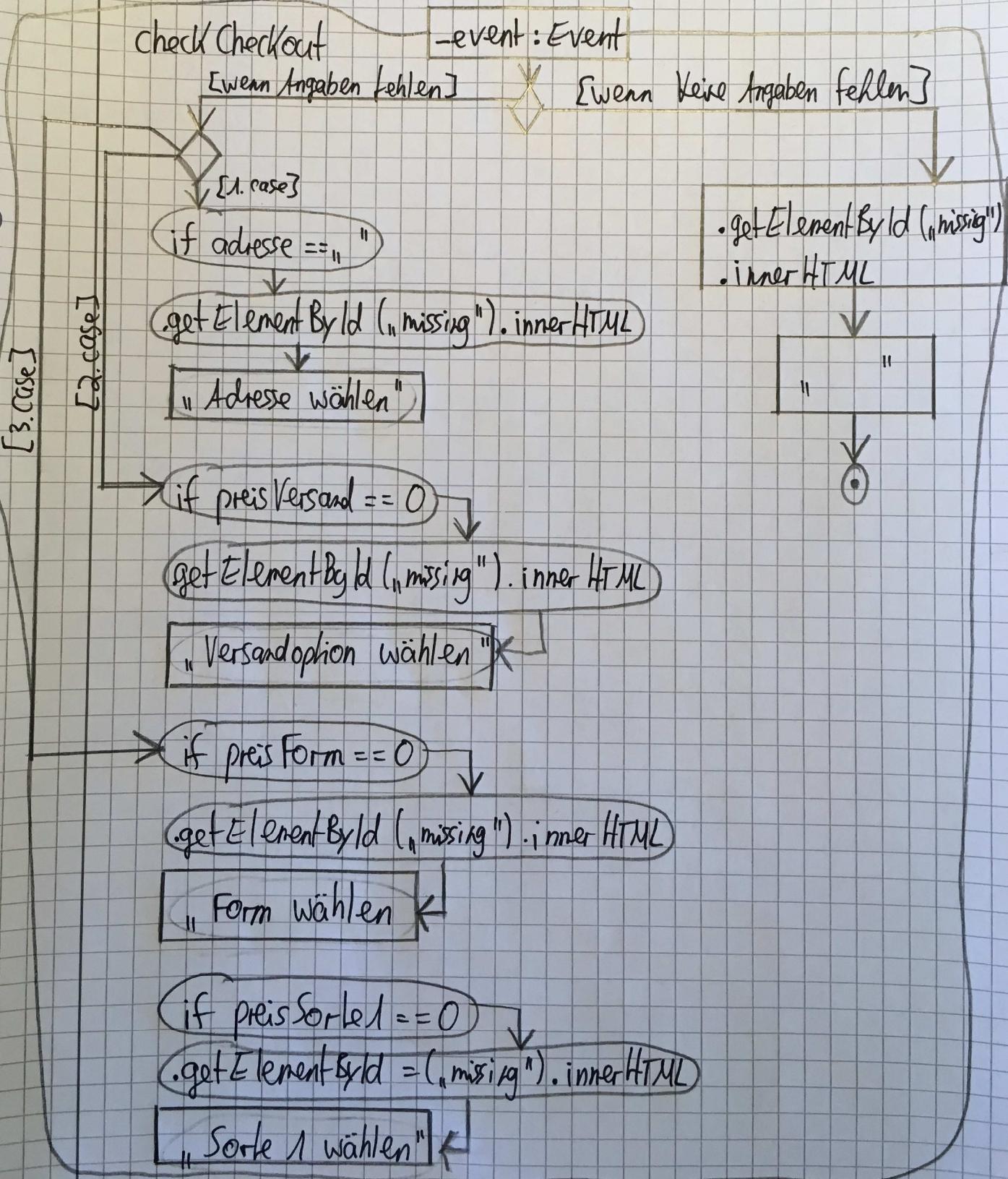
let fieldset: HTMLElement = getElementById ("fieldset")

.addEventListener ("change", handleChange) !!

let preisForm: number = 0
let preisSorte1: number = 0
let preisSorte3: number = 0
let preisSorte4: number = 0
let preisVersand: number = 0
let adresse: string = "

let adresse: string = "

document.getElementById("button").
addEventListener("click", checkCheckout) ⚡



event : Event

handleChange

let target : HTMLInputElement = <HTMLInputElement> event.target

[if target ID = "adresse"]

let Node : HTMLElement =
get ElementBy Id ("adresse")

let childNode HTML : string

ChildNode HTML += "..."

[if target ID = "shipping"]

let Node : HTMLElement =
get ElementBy Id ("shipping")

let value : string =
target . Value

let preisIndex : number =
parseInt (value . substr (0,1))

let childNode HTML : string

childNode HTML += "

[if target = "RadioGroup"]

let Node : HTMLElement =
. get ElementBy Id ("form")

let childNode += "name" + - Preis

[if target = Sorte 1-4]

let node : HTMLElement =
. get ElementBy Id ("Sorte")

let value : string = target . Value

let preisIndex : number =
parseInt (value . substr (0,1))

let childNode HTML :
string

childNode HTML += "

write HTML

```
let node : HTMLElement = .getElementById ("fieldset")
```

```
let childNodeHTML : string = "
```

[Wenn Anzahl der Kugeln kleiner
als Max. Anzahl Kugeln ist]

[wenn alle Eissoßen erstellt
sind geht es weiter]

```
childNodeHTML +=  
"Kugel X"
```

```
childNodeHTML +=  
"Select name='Select' id='KugelX'"
```

↓ [for let i: number; i < eissofex.length; i++)

```
childNodeHTML +=  
.option value" + i +  
eissofex[i].name +  
eissofex[i].preis
```

```
childNodeHTML +=  
</Select>
```

Auswahl von Form

childNode HTML += „Waffel oder Becher?“

[for let i: number = 0; i < formOptionen.length; i++)

childNode HTML += „<input type='radio'

name='Radiogroup' value=''" + i +

formOptionen[i].name + formOptionen[i].preis

childNode HTML += (label for='check' + i

+ formOptionen[i].name

[Lieferoptionen wählen]

childNode HTML += „Lieferoptionen“

childNode HTML += <select name='Select'
id='Ship'>

[for let i: number = 0; i < Lieferoptionen.length; i++)

childNode HTML += <option value=" + i

+ Lieferoptionen[i].name + Lieferoptionen[i].preis">

childNode HTML += </select>

[Adressfeld]

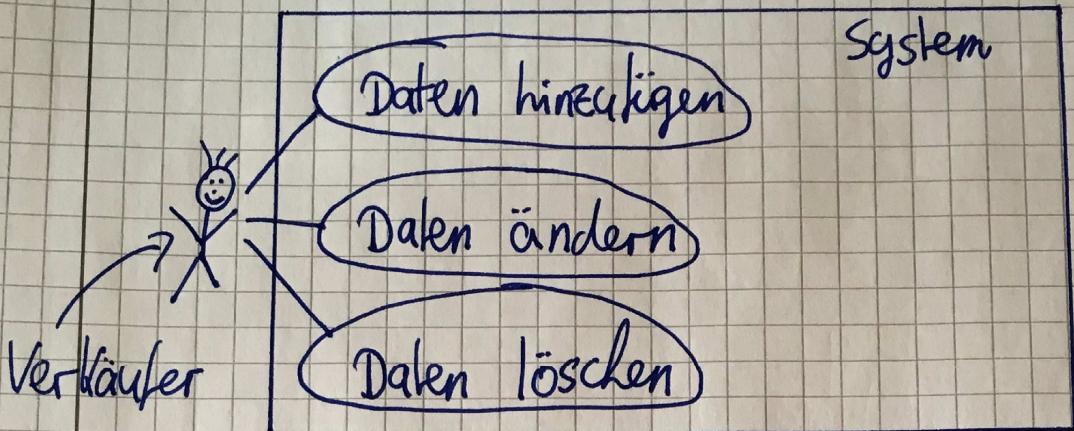
childNode HTML += „Adresse“

childNode HTML += <input type='text'

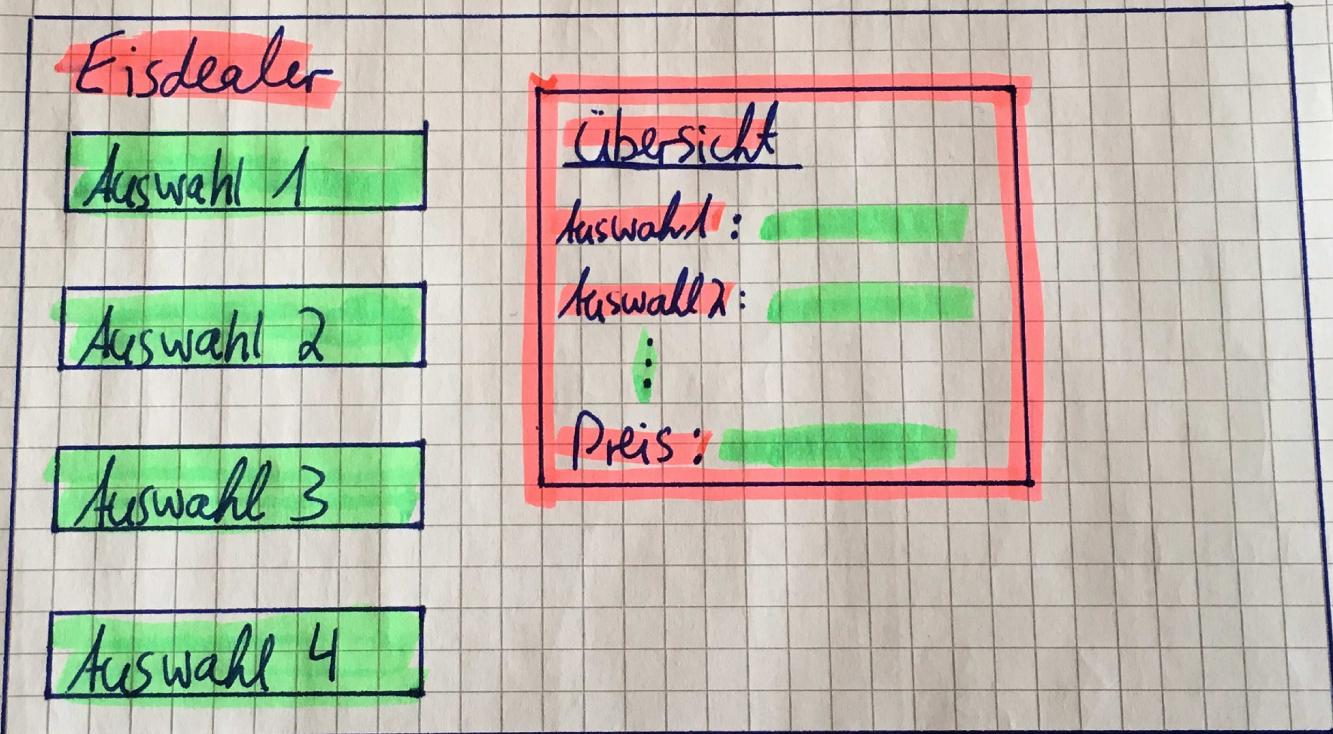
name='text' id='adresse'>

node.innerHTML += childNode HTML

Aufgabe 5 - Anwendungsfalldiagramm



Aufgabe 5 - Skizze



Werden vom HTML generiert (statisch)

Werden über TS/JS generiert

Aufgabe 5 - Datenstruktur

Als erstes soll ein Interface erstellt werden, das für die Optionen den Namen (String) und den Preis (number) definiert, da das zwei Eigenschaften sind, in denen sich die Optionen voneinander unterscheiden.

Anschließend wird für jede Auswahlmöglichkeit ein Array erstellt, in das die einzelnen Objekte eingefügt werden können (oder ergänzt/bearbeitet/gelöscht)

Beispiel:

```
export interface products {  
    name: string,  
    preis: number,  
}
```

```
export let eissorten: products [] = [{name: "Vanille", preis: 1},  
export let versandoptionen: products [] = [{name: "DHL", preis: 4}, {...}];
```