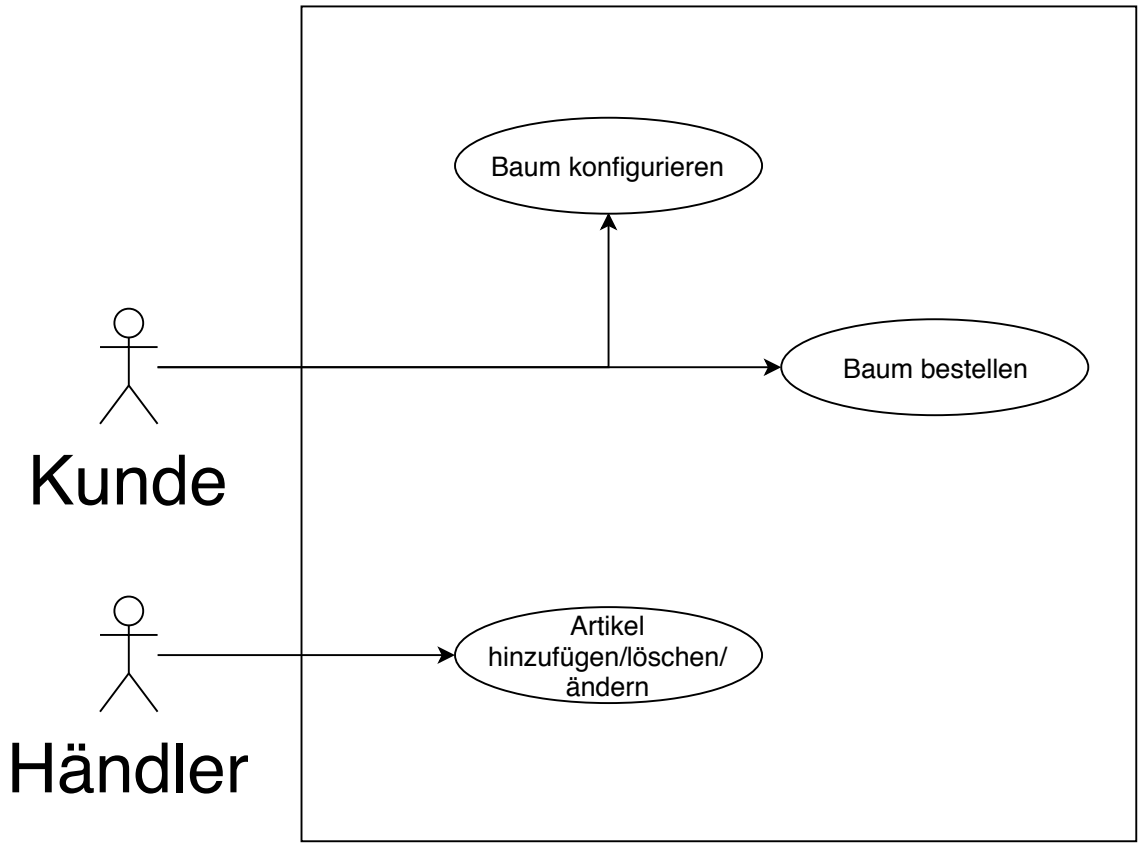
Aufgabe 4 - Konzept

Anwendungsfalldiagramm



Screenlayout

<body>

** (dynamisch)**

Baumart

dropdown list [id="treeType"]

Extras

X checklist

Candles

dropdown list [id="candleType]

Holder

dropdown list [id="holderType"]

** (dynamisch)**

Tree

Lorem ipsum dolor sit amet,

Extras

<ul>

Candles

Lorem ipsum dolor sit amet,

Holder

Lorem ipsum dolor sit amet,

TOTAL 999,-

** (statisch)**

Name

text input [id="checkoutName"]

Street

text input [id="checkoutStreet"]

Postal Code

text input [id="checkoutPostal"]

City

text input [id="checkoutCityname"]

E-Mail

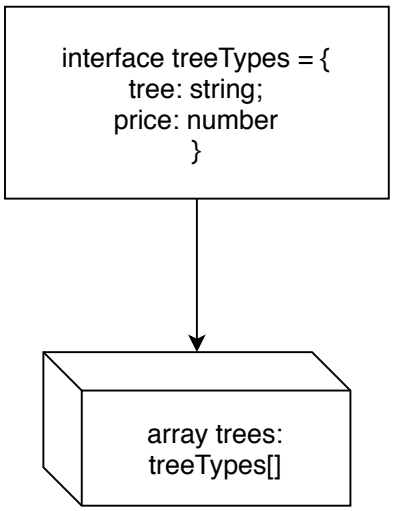
text input [id="checkoutEmail"]

<button id="checkoutConfirm">

Checkout

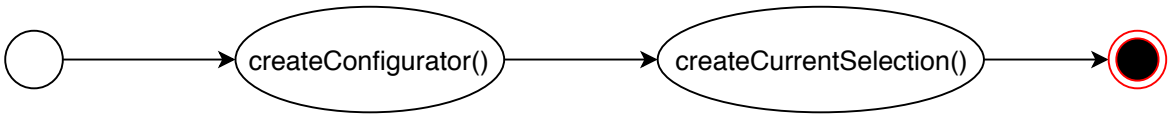
Datenstruktur

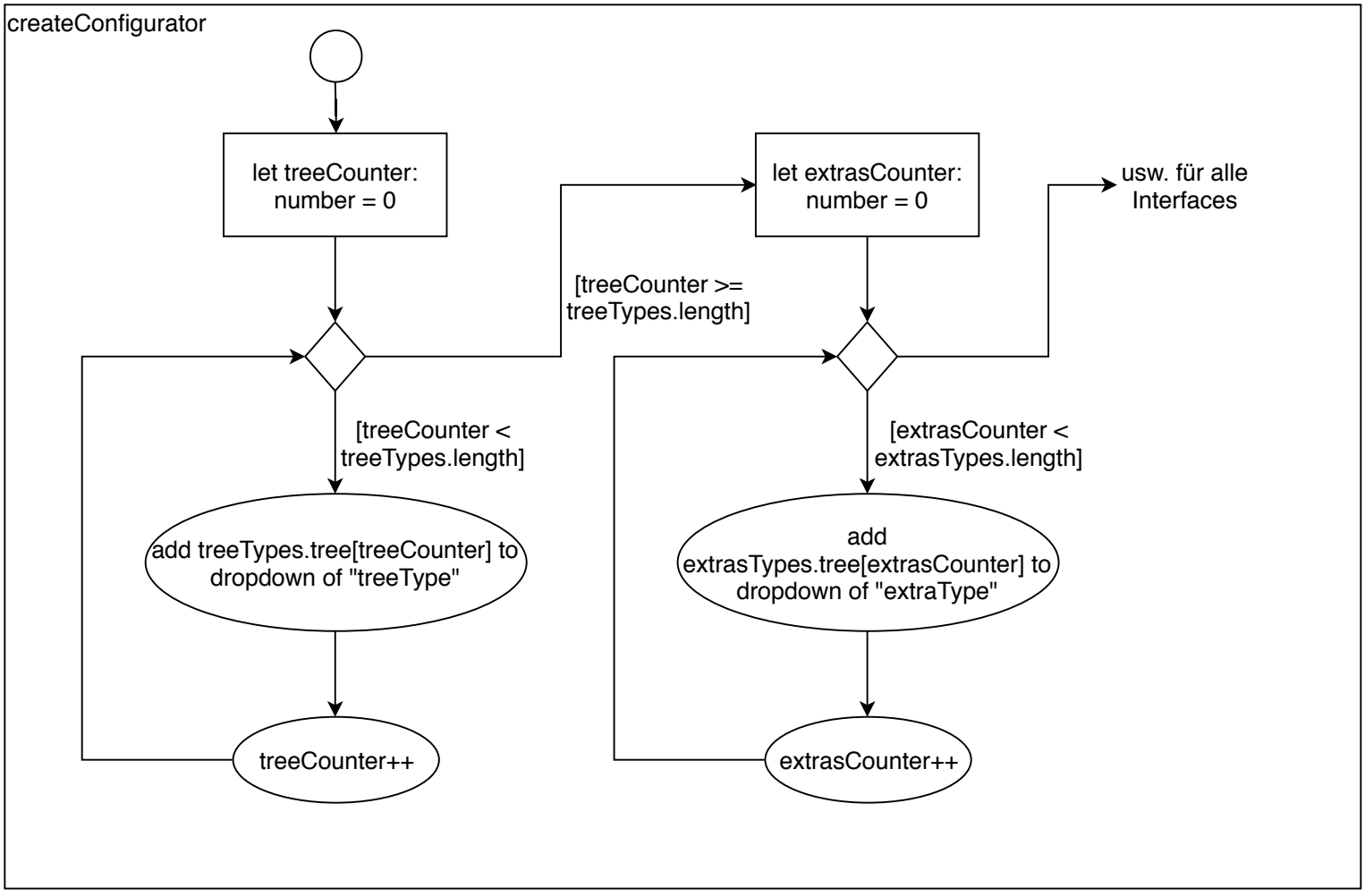
Bäume

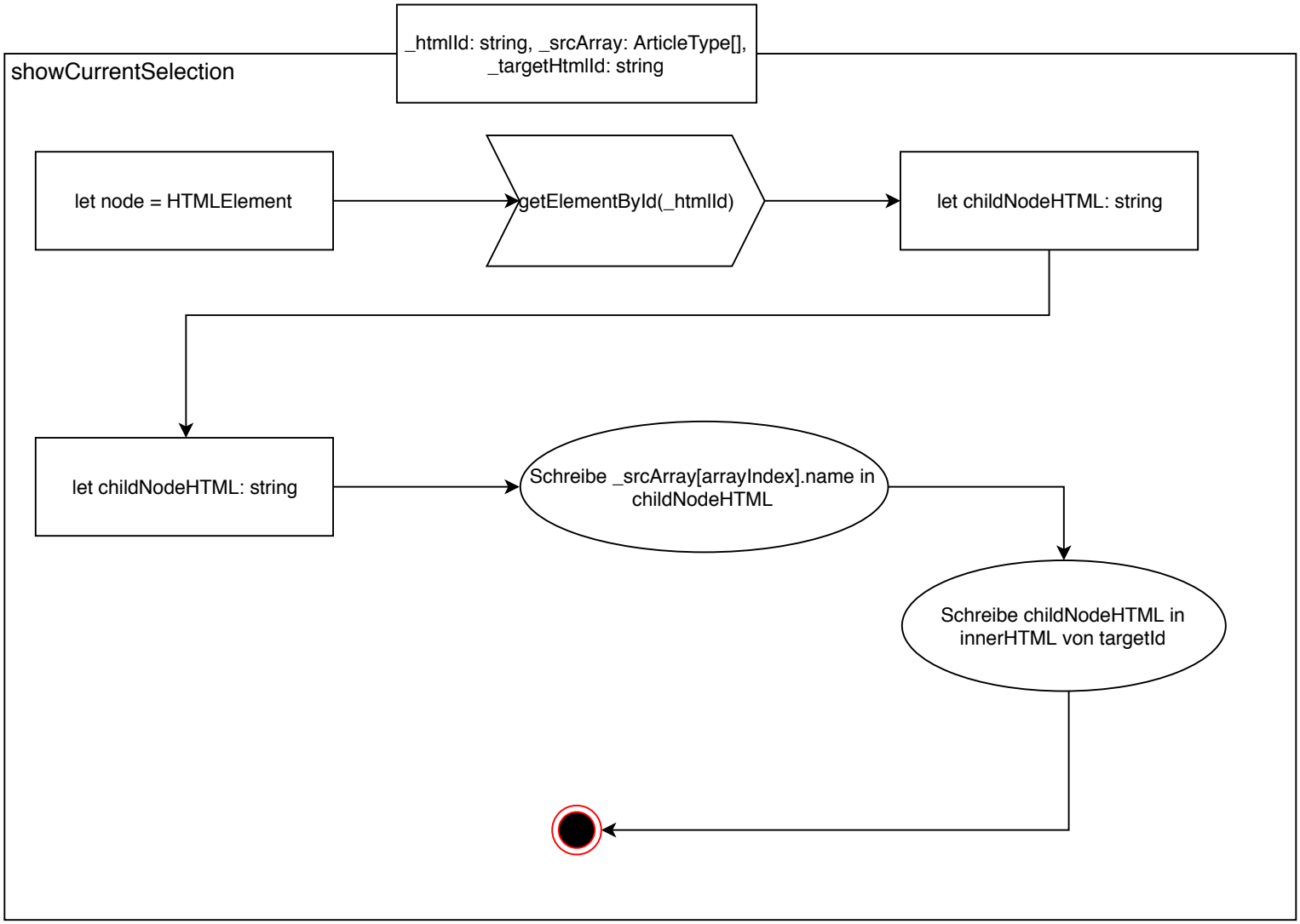


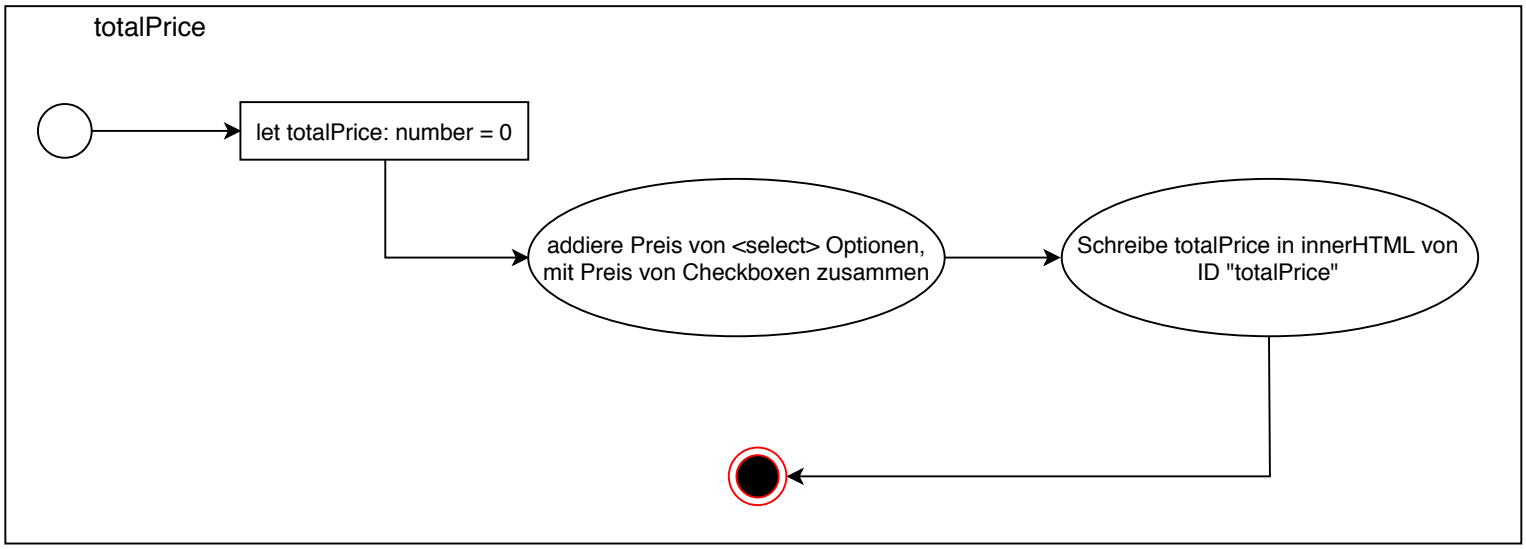
Alle Artikelkategorien bekommen ihr eigenes Interface und werden in einem jeweils neuen Array gespeichert

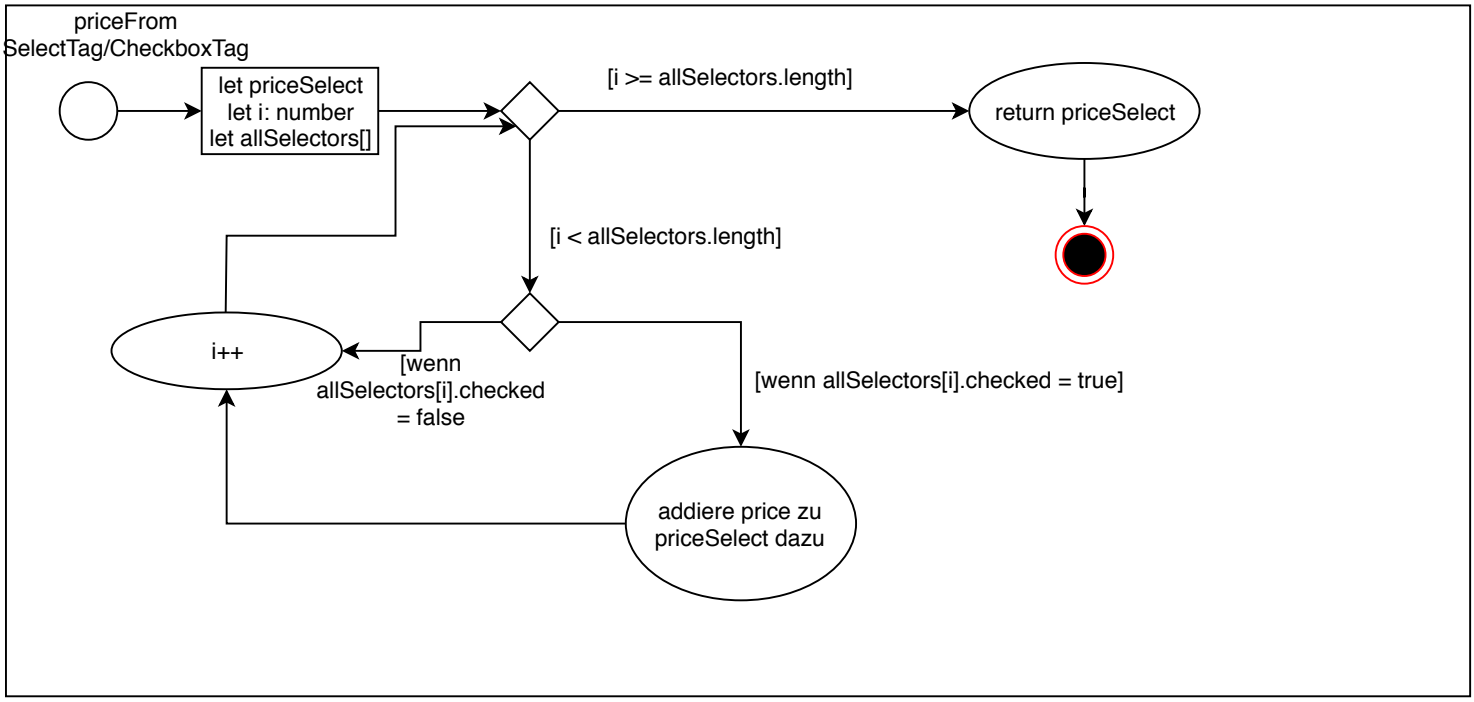
Flowcharts

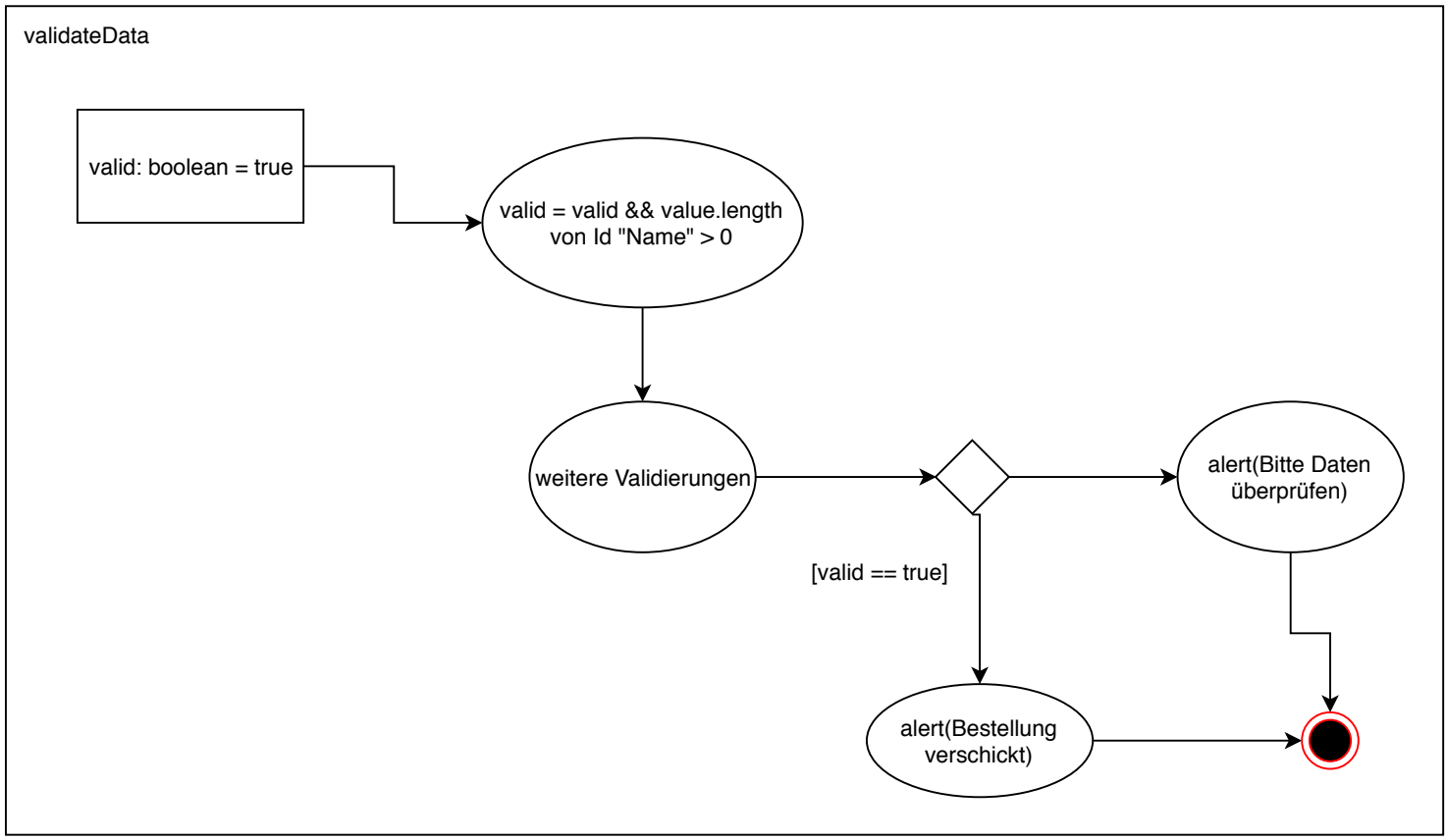
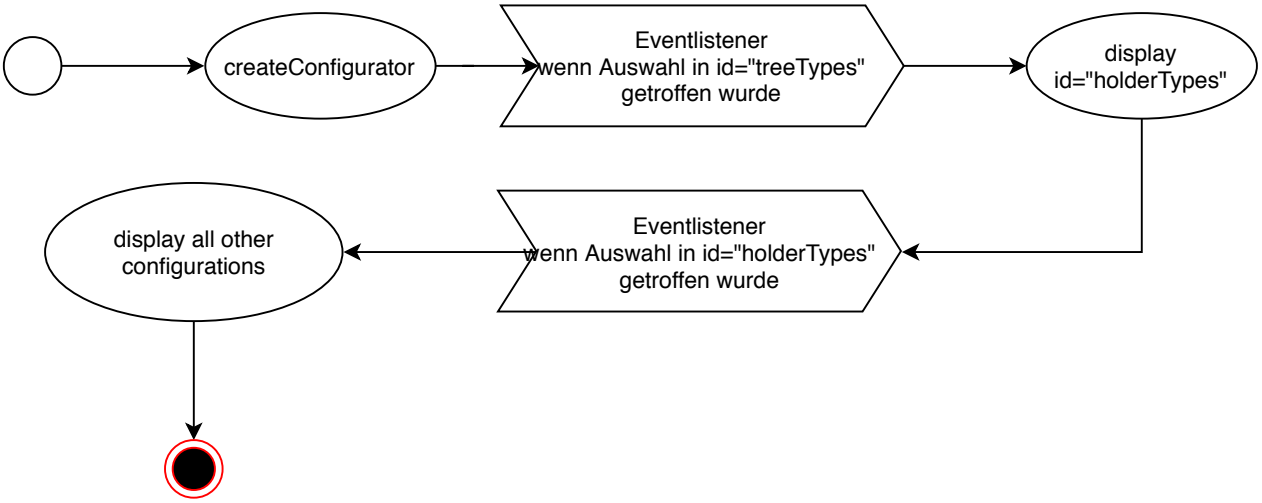





showCurrentSelection
_htmlId: string, _srcArray: ArticleType[], _targetHtmlId: string
let node = HTMLElement
getElementById(_htmlId)
let childNodeHTML: string
let childNodeHTML: string
Schreibe _srcArray[arrayIndex].name in childNodeHTML
Schreibe childNodeHTML in innerHTML von targetId


totalPrice
let totalPrice: number = 0
addiere Preis von <select> Optionen, mit Preis von Checkboxen zusammen
Schreibe totalPrice in innerHTML von ID "totalPrice"


priceFrom
SelectTag/CheckboxTag
let priceSelect
let i: number
let allSelectors[]
[i >= allSelectors.length]
return priceSelect
[i < allSelectors.length]
i++
[wenn
allSelectors[i].checked
= false
[wenn allSelectors[i].checked = true]
addiere price zu
priceSelect dazu


validateData
valid: boolean = true
valid = valid && value.length
von Id "Name" > 0
weitere Validierungen
alert(Bitte Daten
überprüfen)
[valid == true]
alert(Bestellung
verschickt)

Aufgabe 5 - Erweiterung