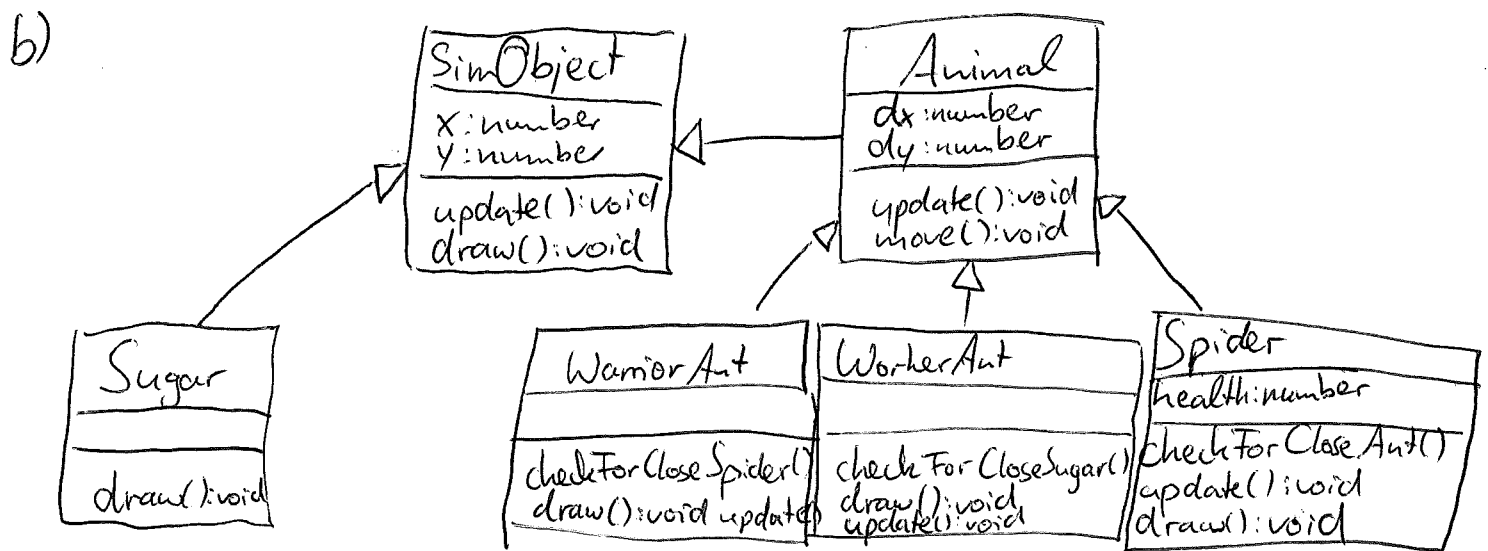
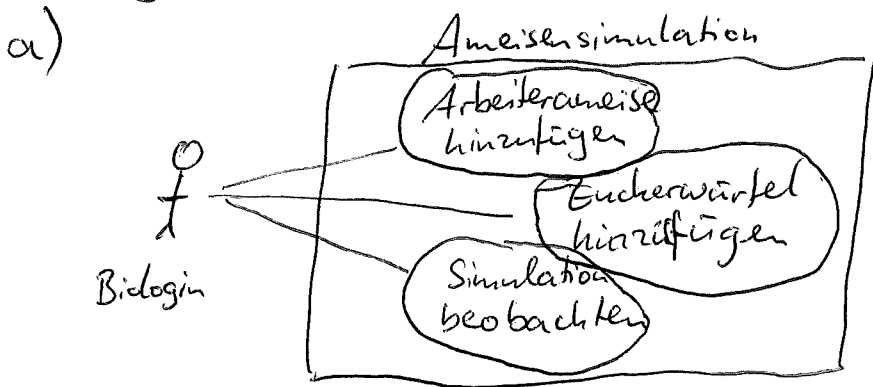


# Aufgabe 1

1. richtig
2. falsch. git ist ein Versionskontrollsystem, GitHub ist ein Hostinganbieter für git.
3. richtig
4. richtig
5. falsch
6. falsch Der Debugger hilft beim Finden von Laufzeitfehlern
7. falsch Die GET Methode gibt die Infos über die URL weiter
8. richtig
9. falsch Der Ursprung ist oben links
10. falsch Es heißt Subklasse der Superklasse

## Aufgabe 2

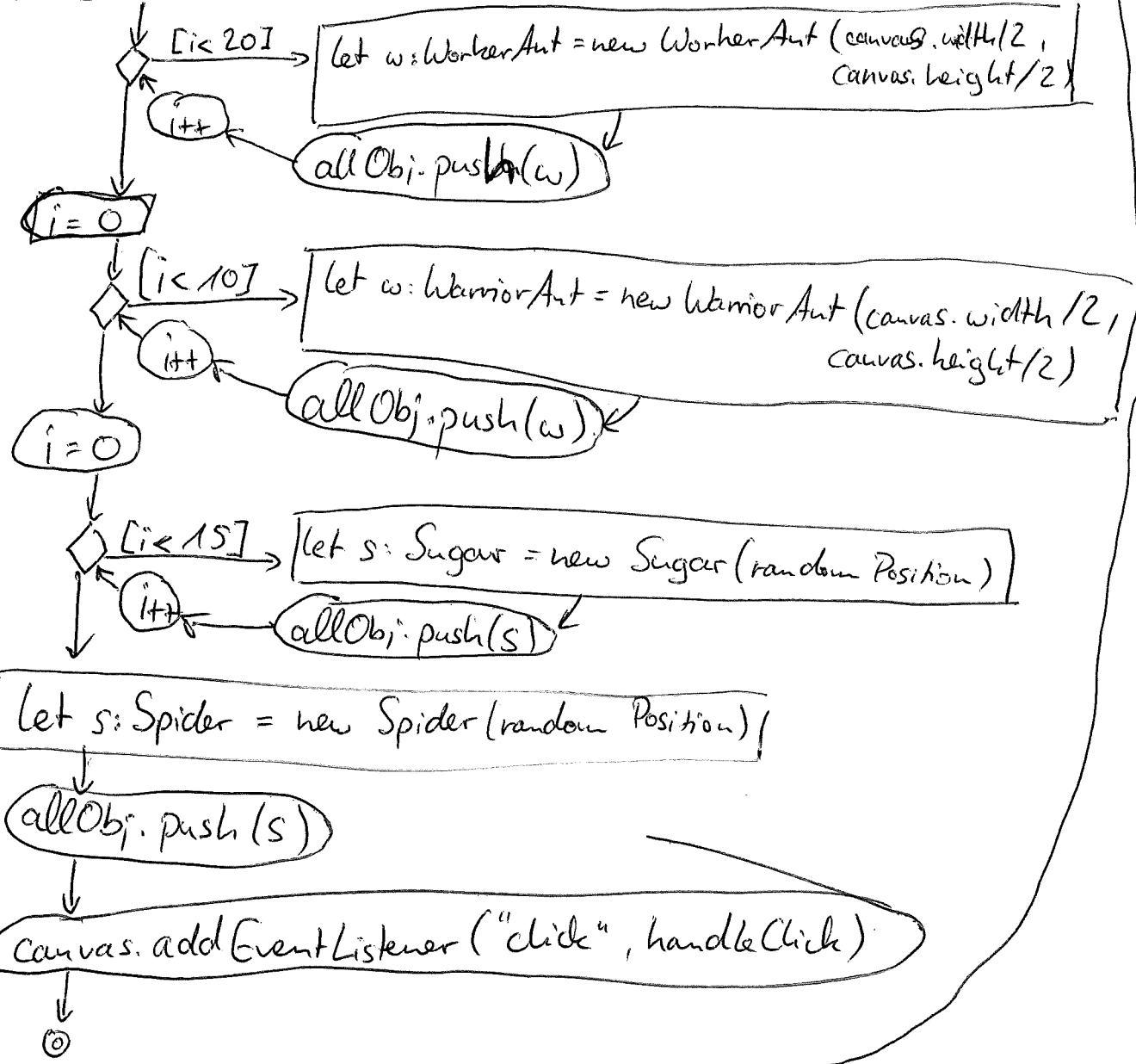


c)

prepare

let allObj: ~~Sim~~ SimObject[] = []

let i: number = 0



d)

WarriorAnt.move

super.move()

⊙

Animal.move

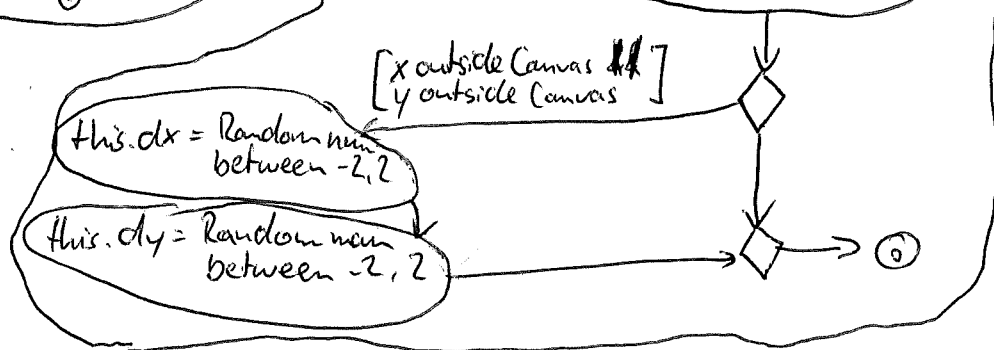
this.x += this.dx  
this.y += this.dy

[x outside Canvas  
y outside Canvas]

this.dx = Random num  
between -2, 2

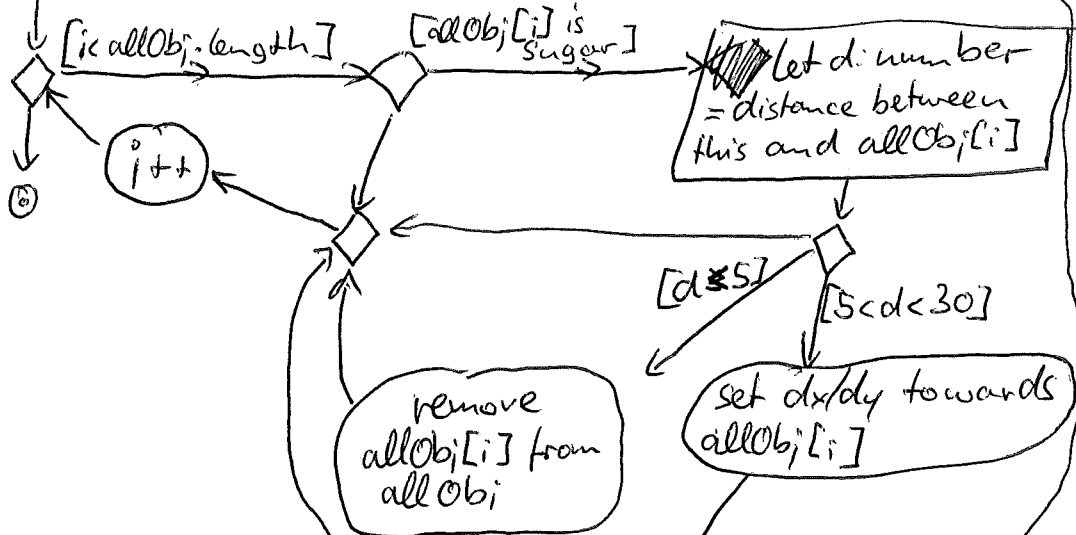
this.dy = Random num  
between -2, 2

⊙



## Check For Close Sugar

[let i: number = 0]



e)

[ev: MouseEvent]

handleClick

[ev.button = rightclick]

[ev.button = leftclick]

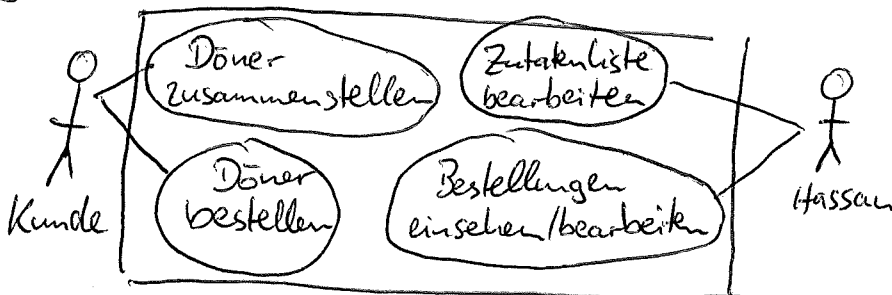
allObj.push(  
new Sugar(-ev.clientX, -ev.clientY))

allObj.push(  
new WorkerAnt(-ev.clientX, -ev.clientY))



## Aufgabe 3 Dönerlieferdienst

a)



## b) Dein Döner <h1>

input type: radio

input type: checkbox mit labels

input type: text

input type: submit id=submit

dynamic

static

Bestellen

Gesamtpreis: XX €

span id=totalprice

fieldset id=Bro

fieldset id=Belag

div id=dyncontent

fieldset

form

name

Strasse

PLZ, Ort

Salat 0,50€

Fleisch (Huhn) 0,50€

Fleisch (Rind) 0,50€

Käse 0,50€

Soße 0,13€

Fladenbrot Fladenbrot Glutenfrei

c)

interface

Zutat

name: string

preis: number

interface

Zutatenliste-kategorie

[name: string]: Zutat[]

type: string

Das hier macht die nächste Aufgabe unnötig kompliziert. Kann also weggelassen werden.

let alleZutaten: Zutatenkategorie[] =

[ { type: "radio", "Brot": [ { name: "Fladenbrot", preis: 0,5 }, { name: "Glutenfrei", preis: 0,5 } ] }, { type: "checkbox", "Belag": [ { name: "Salat", preis: 0,5 }, { name: "Soße", preis: 0,13 } ] } ]

f) Allgemein: /?key1=value1&key2=value2&...  
Konkret: /?brot=Fladenbrot&Salat=ja&Soße=ja&...  
Erklärung spare ich mir, das bekommt ihr selbst hin.

d)

UI Generation

-z: Zutatenkategorie[]

```
let div: HTMLDivElement =
  document.getElementById("dyncontent")
```



⑥

[let z1 of -z]

```
let fi: HTMLFieldsetElement = document.createElement("fieldset")
```



```
let le: HTMLLegendElement = document.createElement("legend")
```

~~le.innerHTML = z1~~