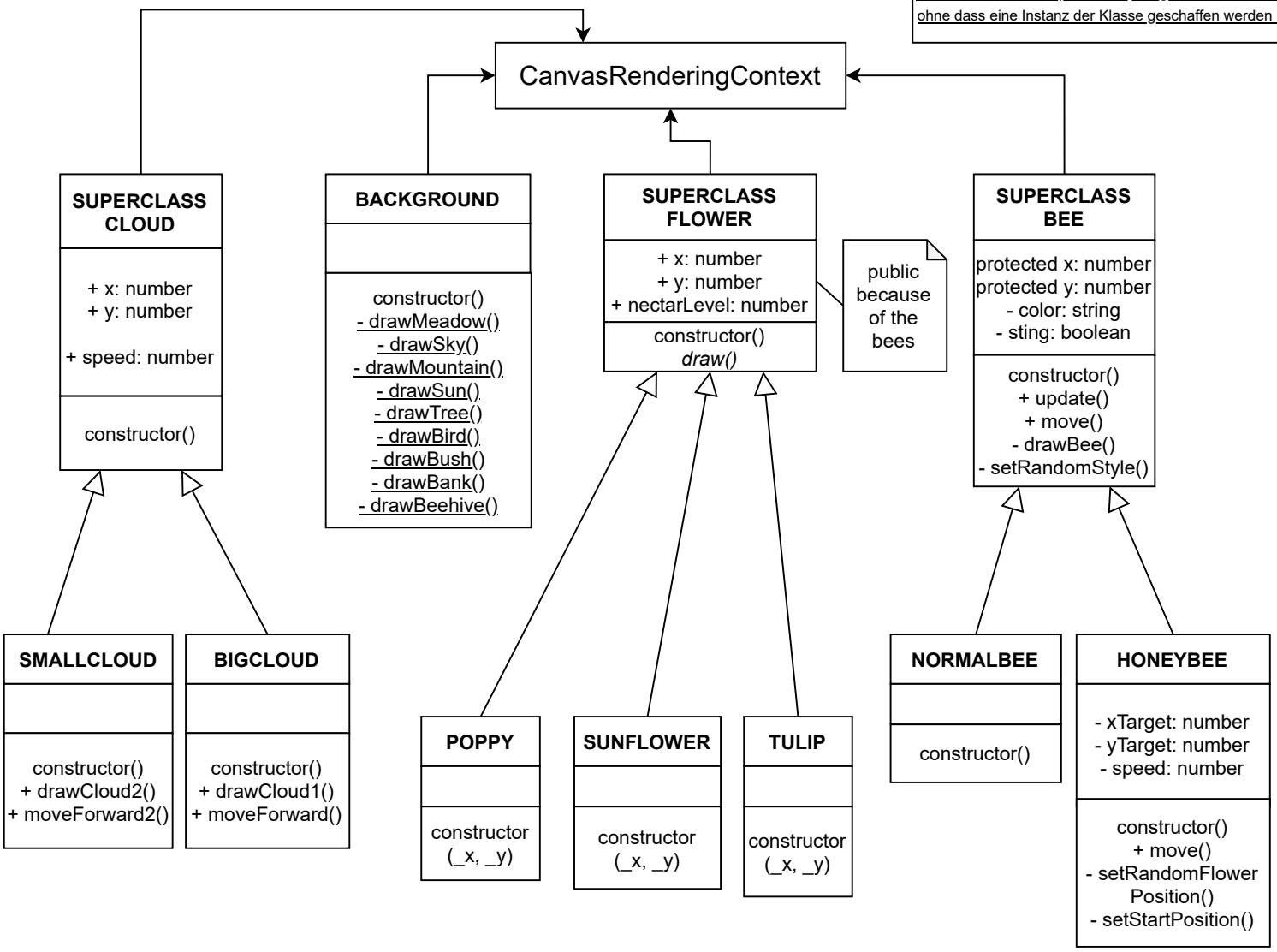
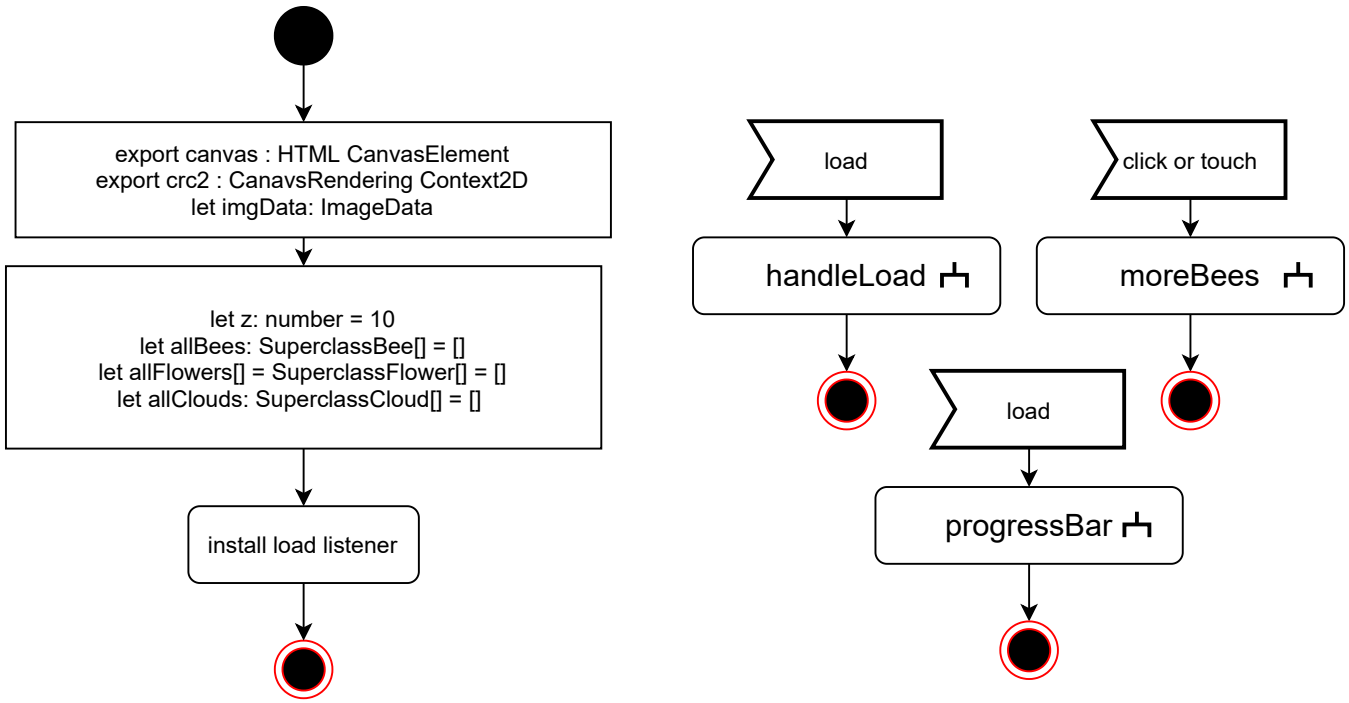
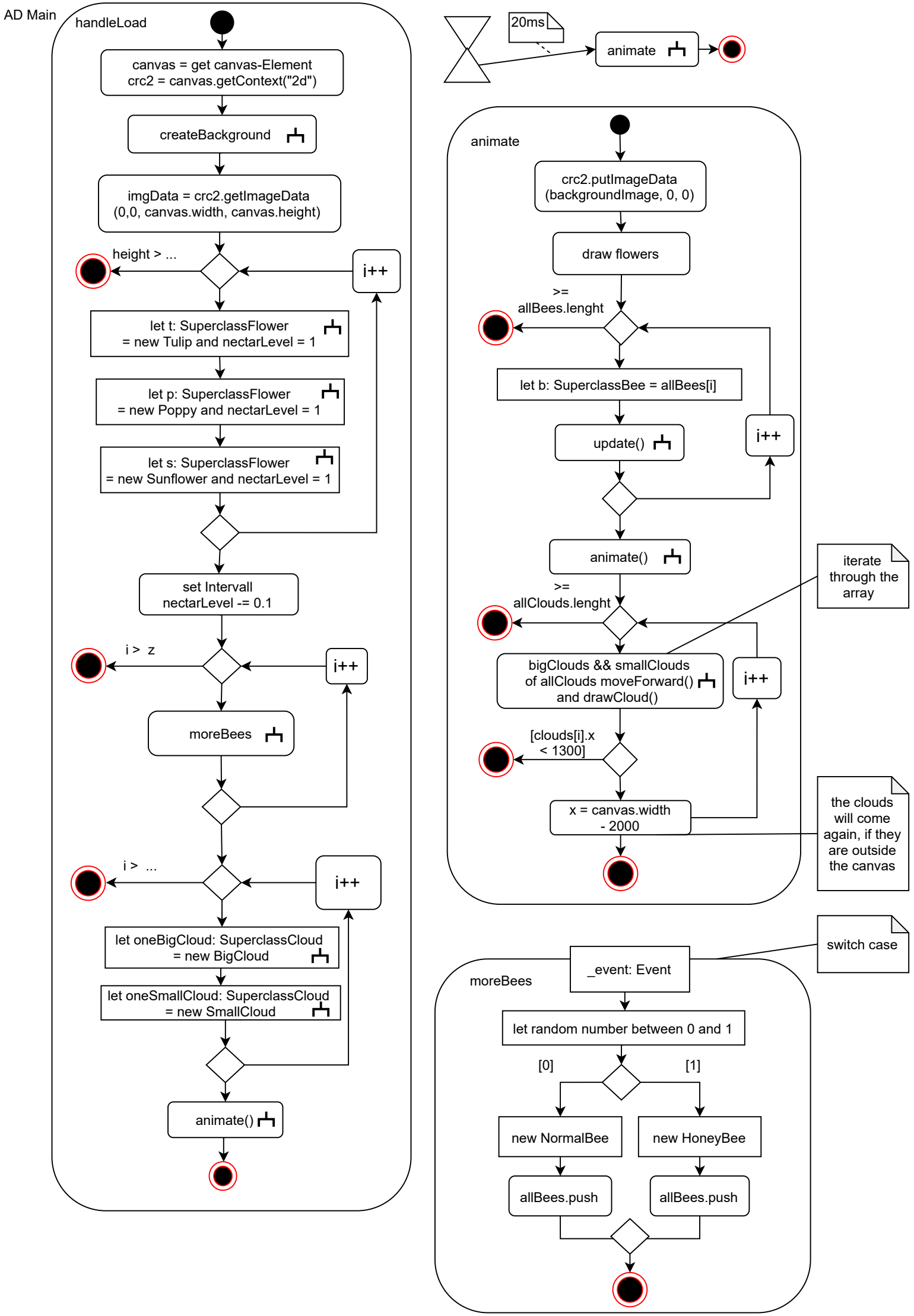
CLASS-DIAGARM MEADOW

kursiv = Abstrakt (Wenn Superklassen Methoden aufweisen, die nur die Subklassen brauchen)
+ = public (wenn alle darauf Zugriff brauchen)
- = private (Zugriff nur in der Klasse)
protected = Zugriff gleichen Klassen und deren Subklassen
unterstrichen = static (Methoden, die genutzt werden können, ohne dass eine Instanz der Klasse geschaffen werden muss)

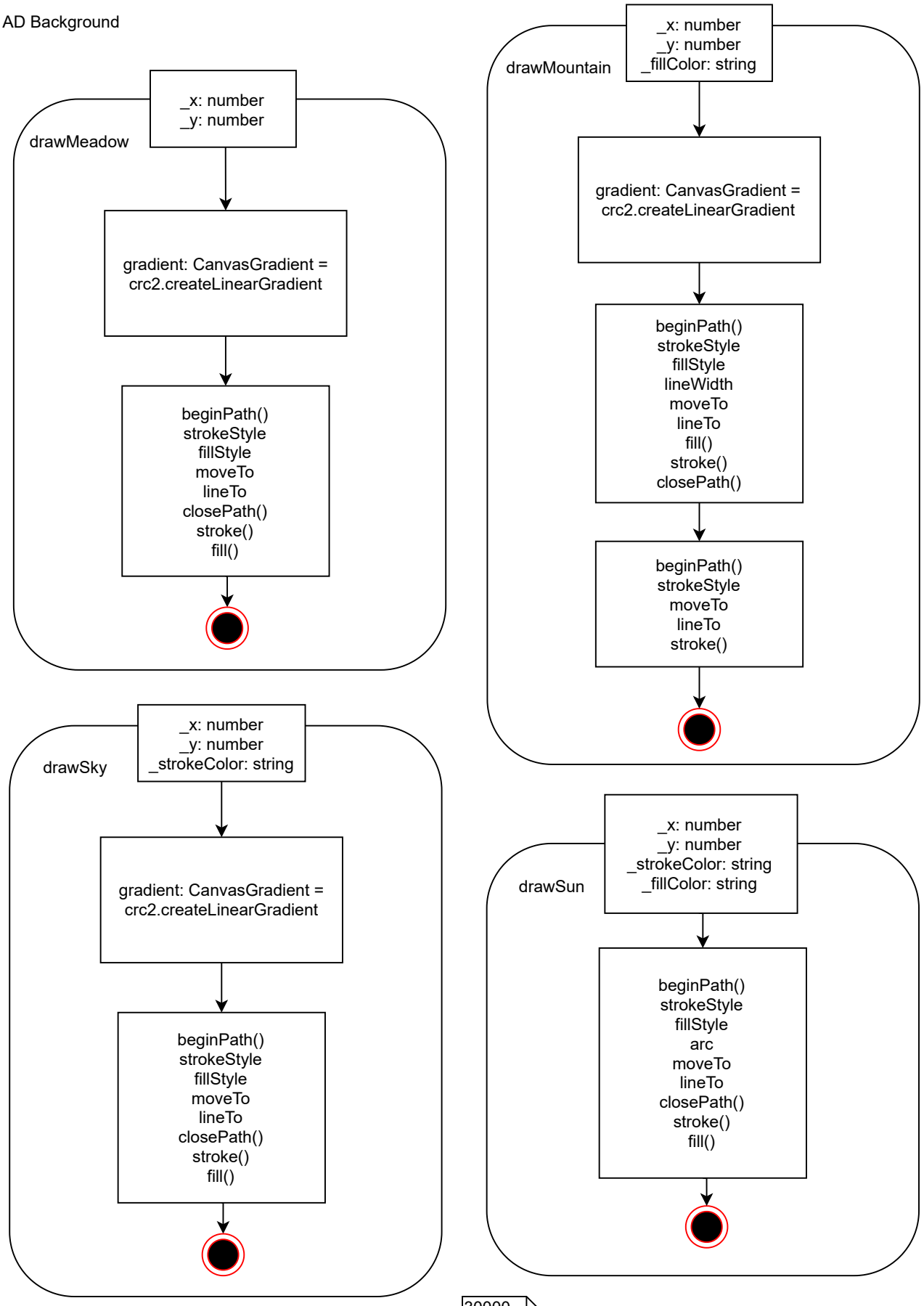


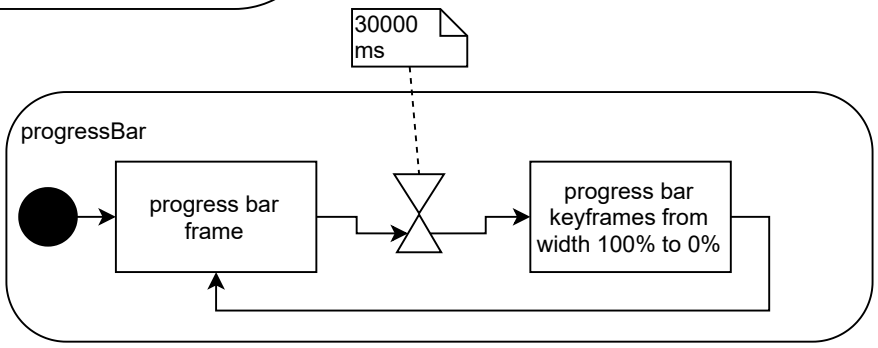
ACTIVITY-DIAGRAM MEADOW



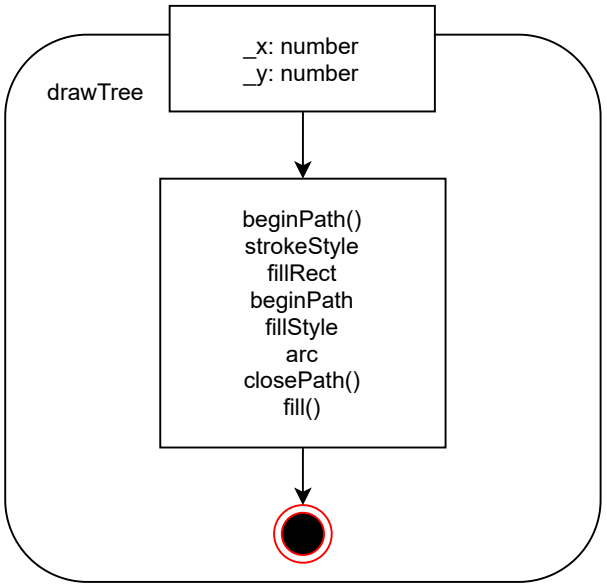

AD Main
handleLoad
canvas = get canvas-Element
crc2 = canvas.getContext("2d")
createBackground
imgData = crc2.getImageData
(0,0, canvas.width, canvas.height)
height > ...
i++
let t: SuperclassFlower
= new Tulip and nectarLevel = 1
let p: SuperclassFlower
= new Poppy and nectarLevel = 1
let s: SuperclassFlower
= new Sunflower and nectarLevel = 1
set Intervall
nectarLevel -= 0.1
i > z
i++
moreBees
i > ...
i++
let oneBigCloud: SuperclassCloud
= new BigCloud
let oneSmallCloud: SuperclassCloud
= new SmallCloud
animate()
20ms
animate
animate
crc2.putImageData
(backgroundImage, 0, 0)
draw flowers
>=
allBees.lenght
let b: SuperclassBee = allBees[i]
update()
i++
animate()
>=
allClouds.lenght
bigClouds && smallClouds
of allClouds moveForward()
and drawCloud()
i++
[clouds[i].x
< 1300]
x = canvas.width
- 2000
iterate
through the
array
the clouds
will come
again, if they
are outside
the canvas
moreBees
_event: Event
switch case
let random number between 0 and 1
[0]
[1]
new NormalBee
new HoneyBee
allBees.push
allBees.push

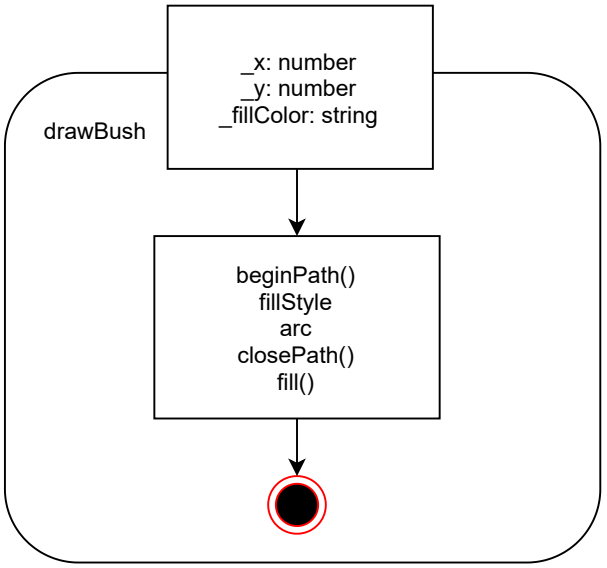
AD Background

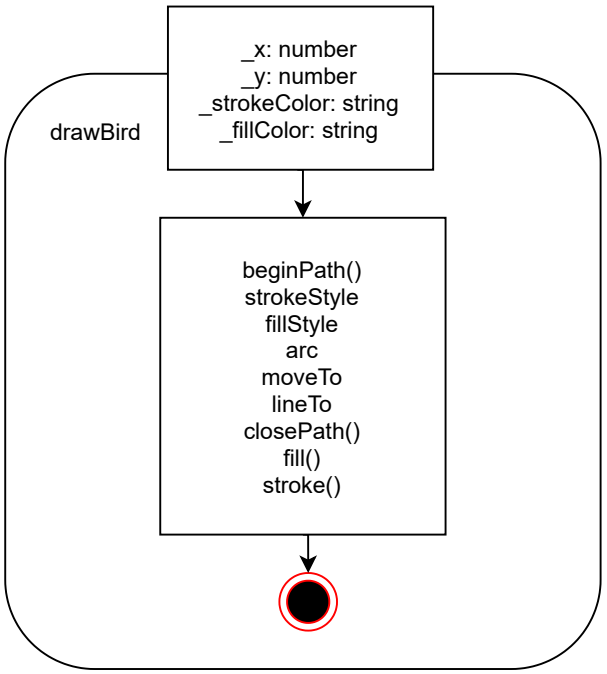


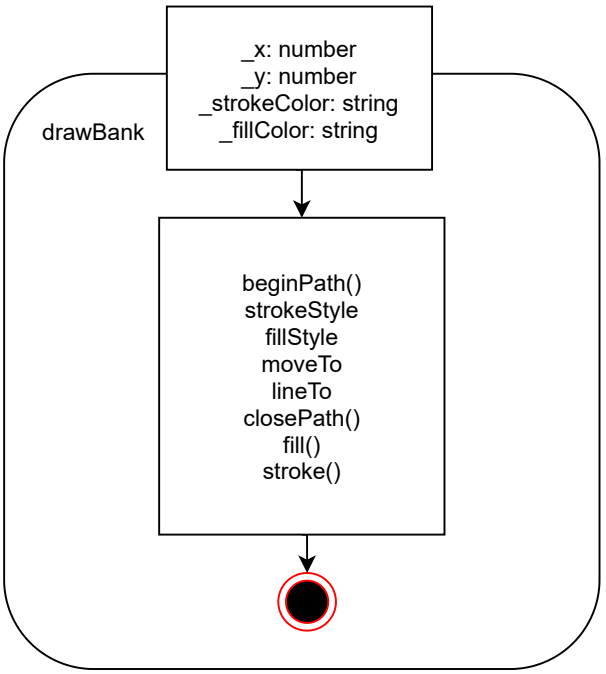


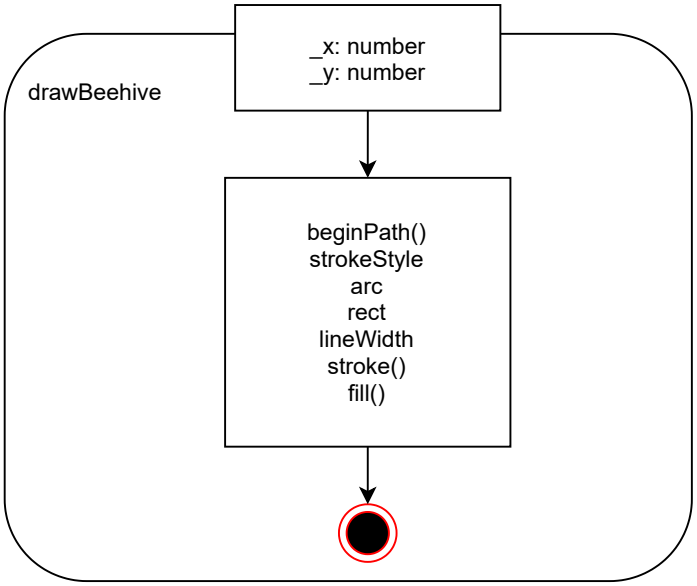
AD Background



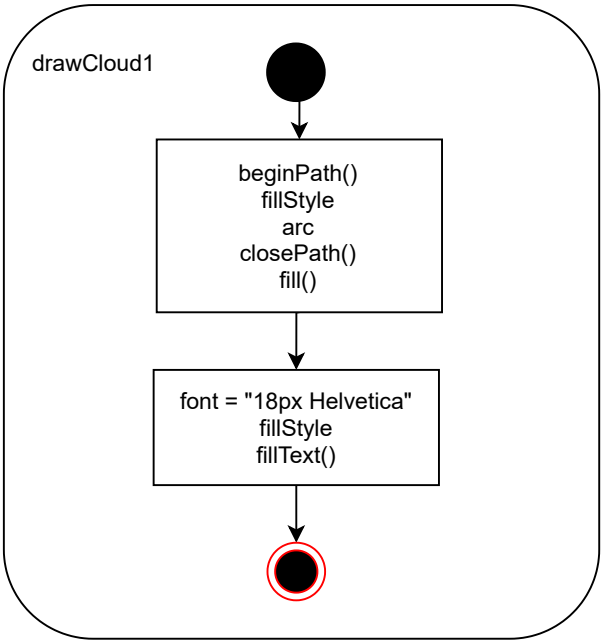


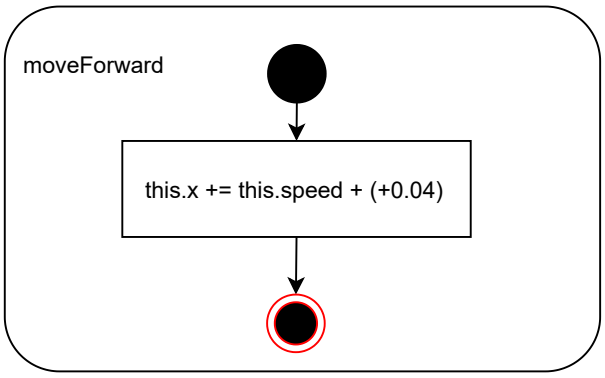




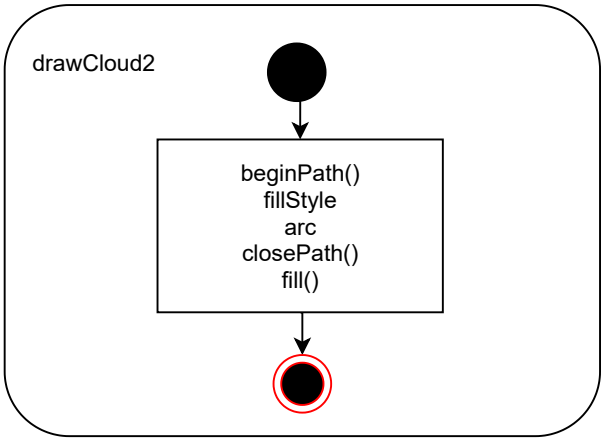


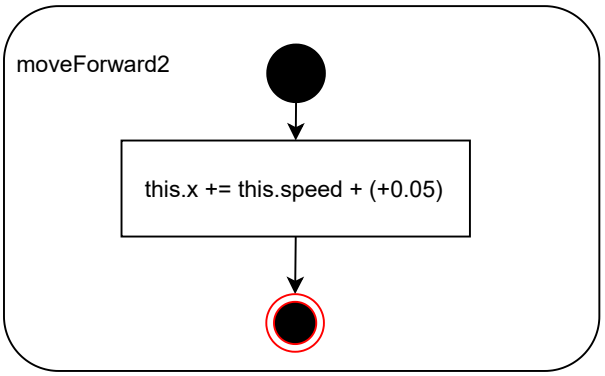
AD Subclass BigCloud



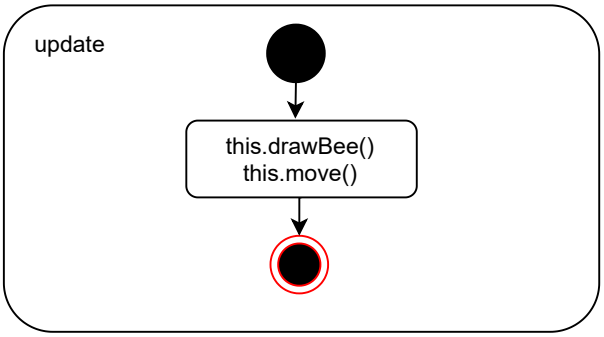


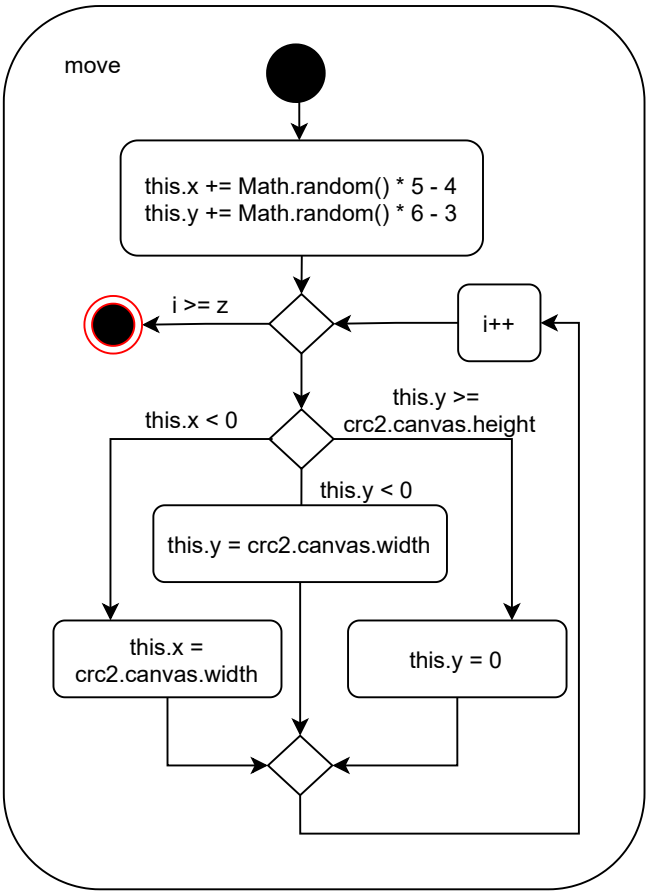
AD Subclass SmallCloud

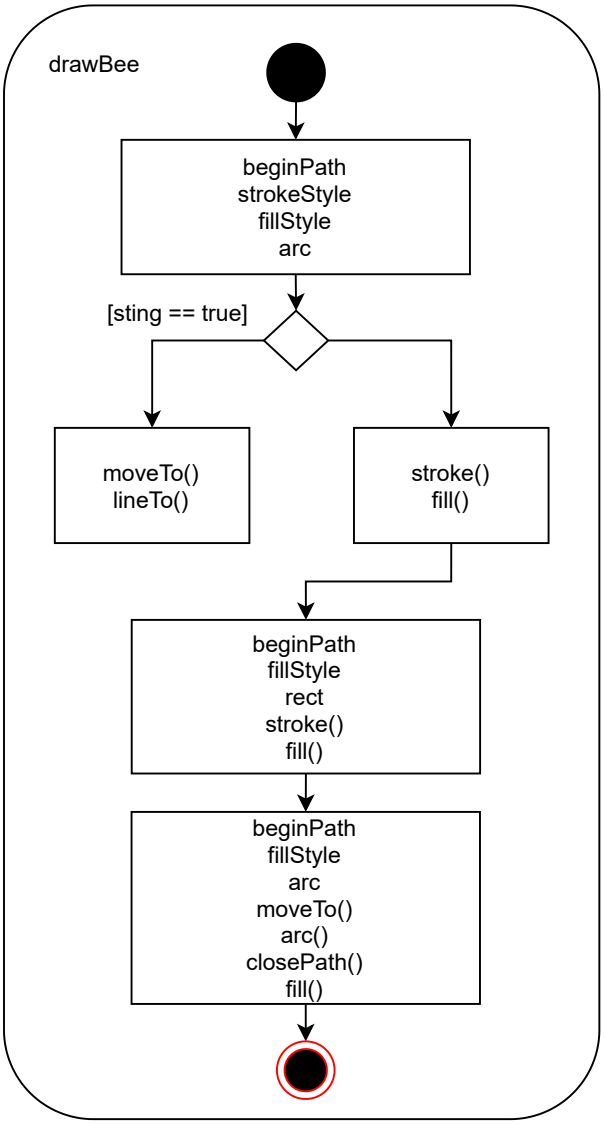


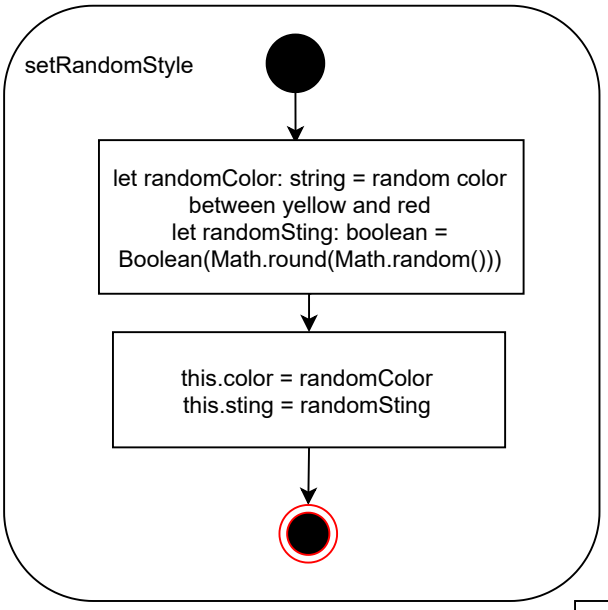


AD SuperclassBee

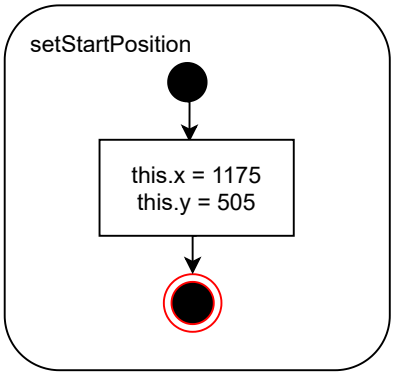


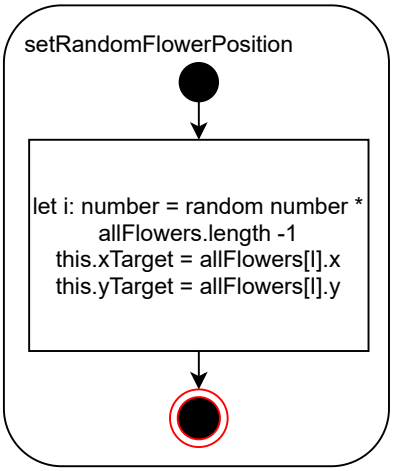


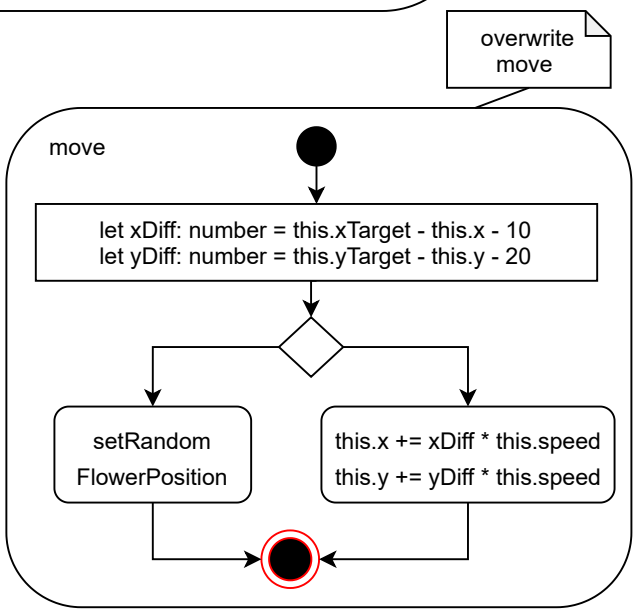




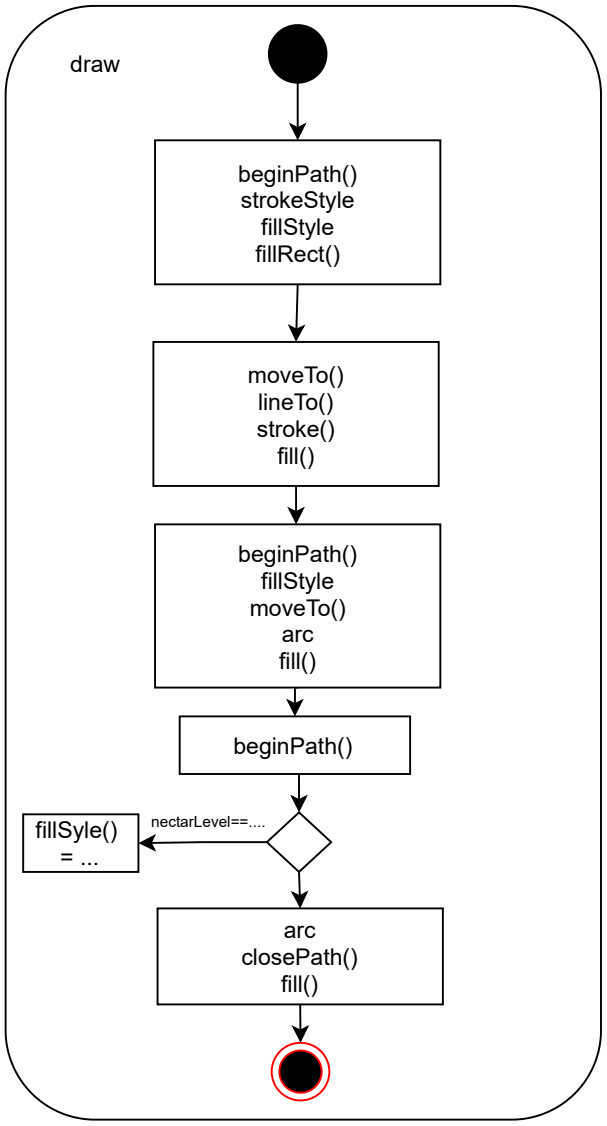
AD Subclass HoneyBee



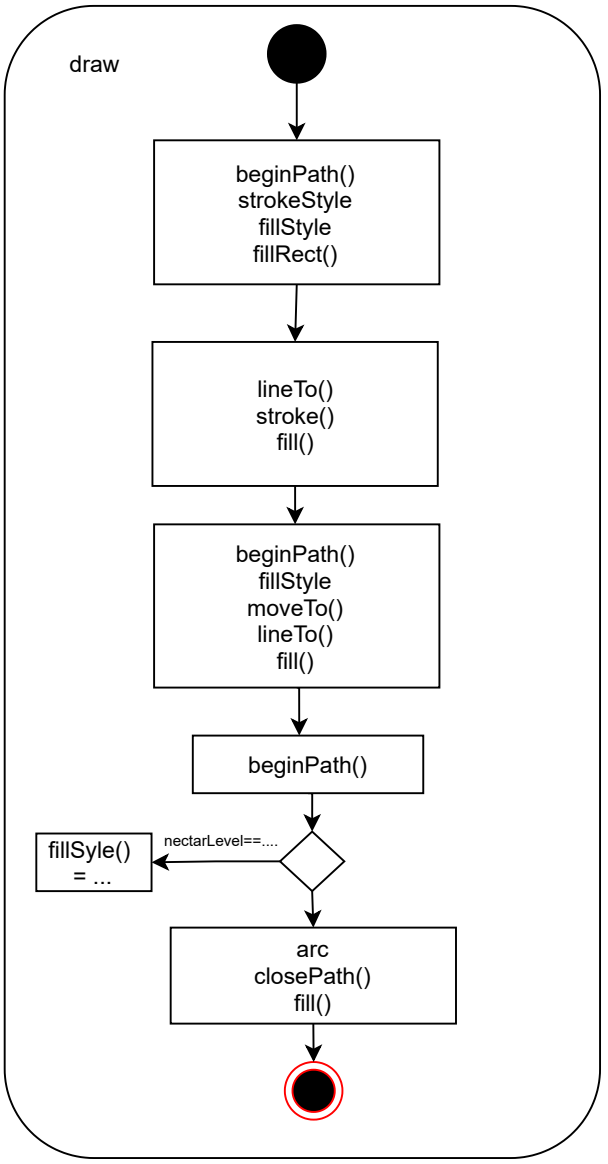
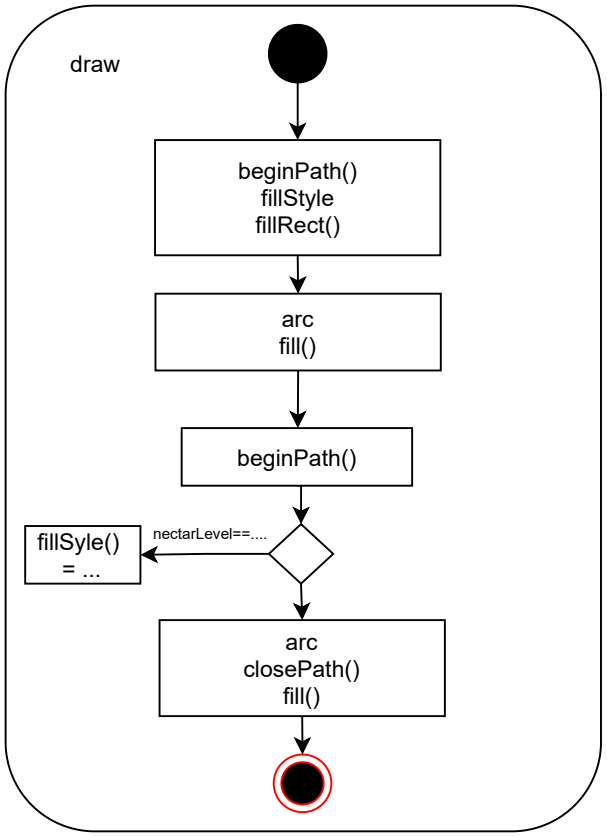




AD Subclass Poppy



AD Subclass Sunflower



AD Subclass Tulip