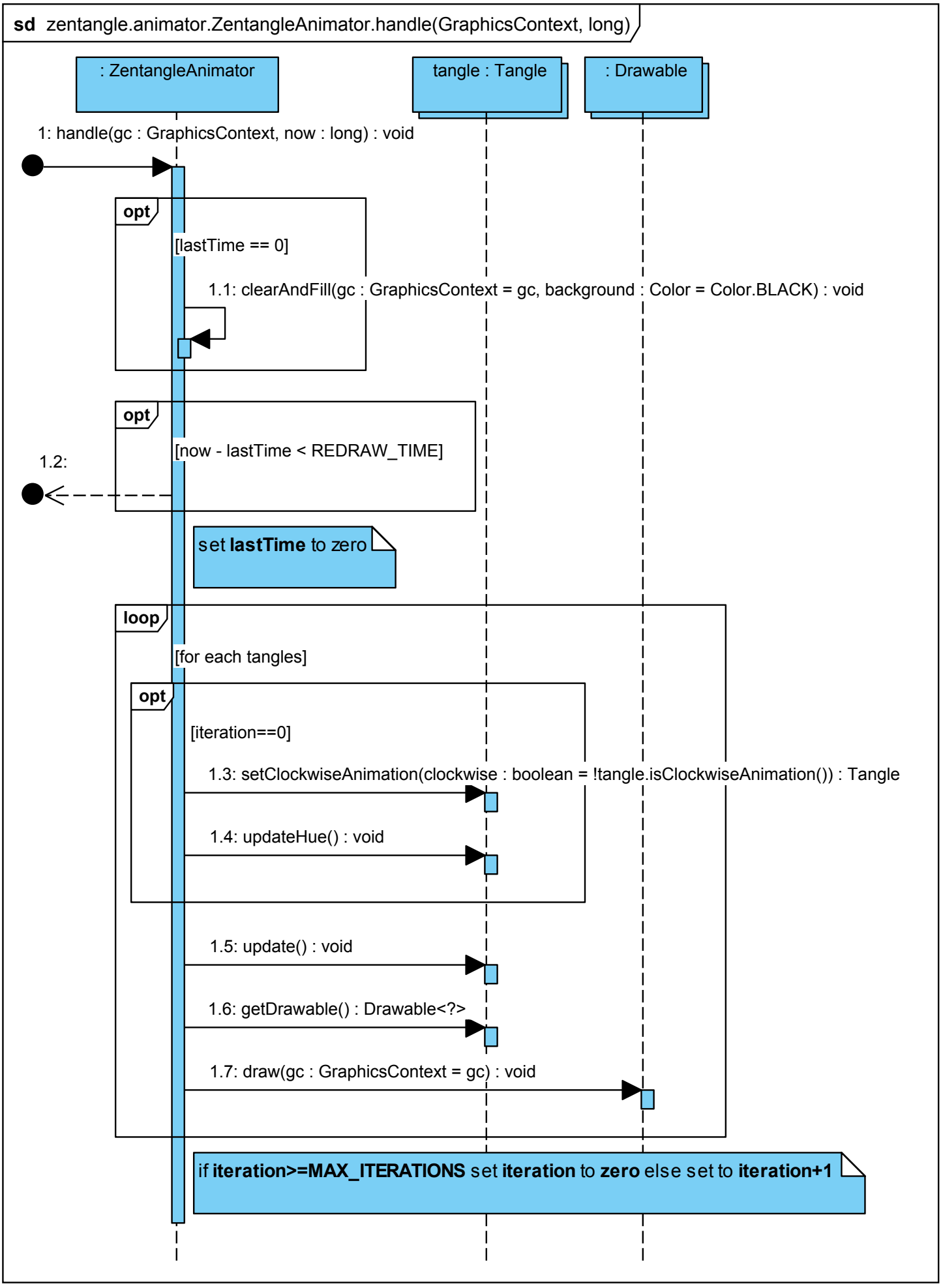
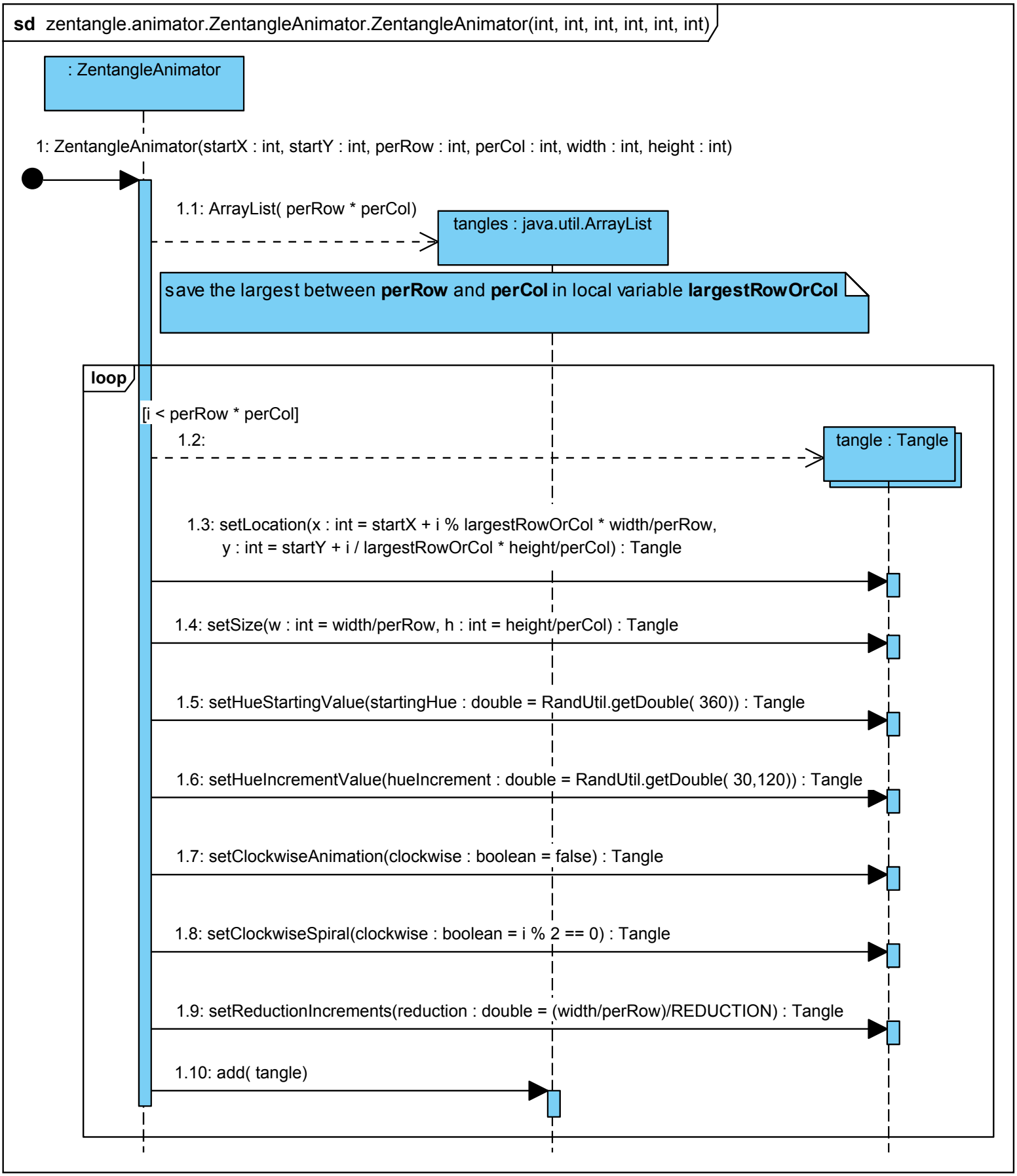

sd zentangle.animator.ZentangleAnimator.handle(GraphicsContext, long)
: ZentangleAnimator
tangle : Tangle
: Drawable
1: handle(gc : GraphicsContext, now : long) : void
opt
[lastTime == 0]
1.1: clearAndFill(gc : GraphicsContext = gc, background : Color = Color.BLACK) : void
opt
[now - lastTime < REDRAW_TIME]
1.2:
set lastTime to zero
loop
[for each tangles]
opt
[iteration==0]
1.3: setClockwiseAnimation(clockwise : boolean = !tangle.isClockwiseAnimation()) : Tangle
1.4: updateHue() : void
1.5: update() : void
1.6: getDrawable() : Drawable<?>
1.7: draw(gc : GraphicsContext = gc) : void
if iteration>=MAX_ITERATIONS set iteration to zero else set to iteration+1


sd zentangle.animator.ZentangleAnimator.ZentangleAnimator(int, int, int, int, int, int)
: ZentangleAnimator
1: ZentangleAnimator(startX : int, startY : int, perRow : int, perCol : int, width : int, height : int)
1.1: ArrayList(perRow * perCol)
tangles : java.util.ArrayList
save the largest between perRow and perCol in local variable largestRowOrCol
loop
[i < perRow * perCol]
1.2:
tangle : Tangle
1.3: setLocation(x : int = startX + i % largestRowOrCol * width/perRow,
y : int = startY + i / largestRowOrCol * height/perCol) : Tangle
1.4: setSize(w : int = width/perRow, h : int = height/perCol) : Tangle
1.5: setHueStartingValue(startingHue : double = RandUtil.getDouble(360)) : Tangle
1.6: setHueIncrementValue(hueIncrement : double = RandUtil.getDouble(30,120)) : Tangle
1.7: setClockwiseAnimation(clockwise : boolean = false) : Tangle
1.8: setClockwiseSpiral(clockwise : boolean = i % 2 == 0) : Tangle
1.9: setReductionIncrements(reduction : double = (width/perRow)/REDUCTION) : Tangle
1.10: add(tangle)