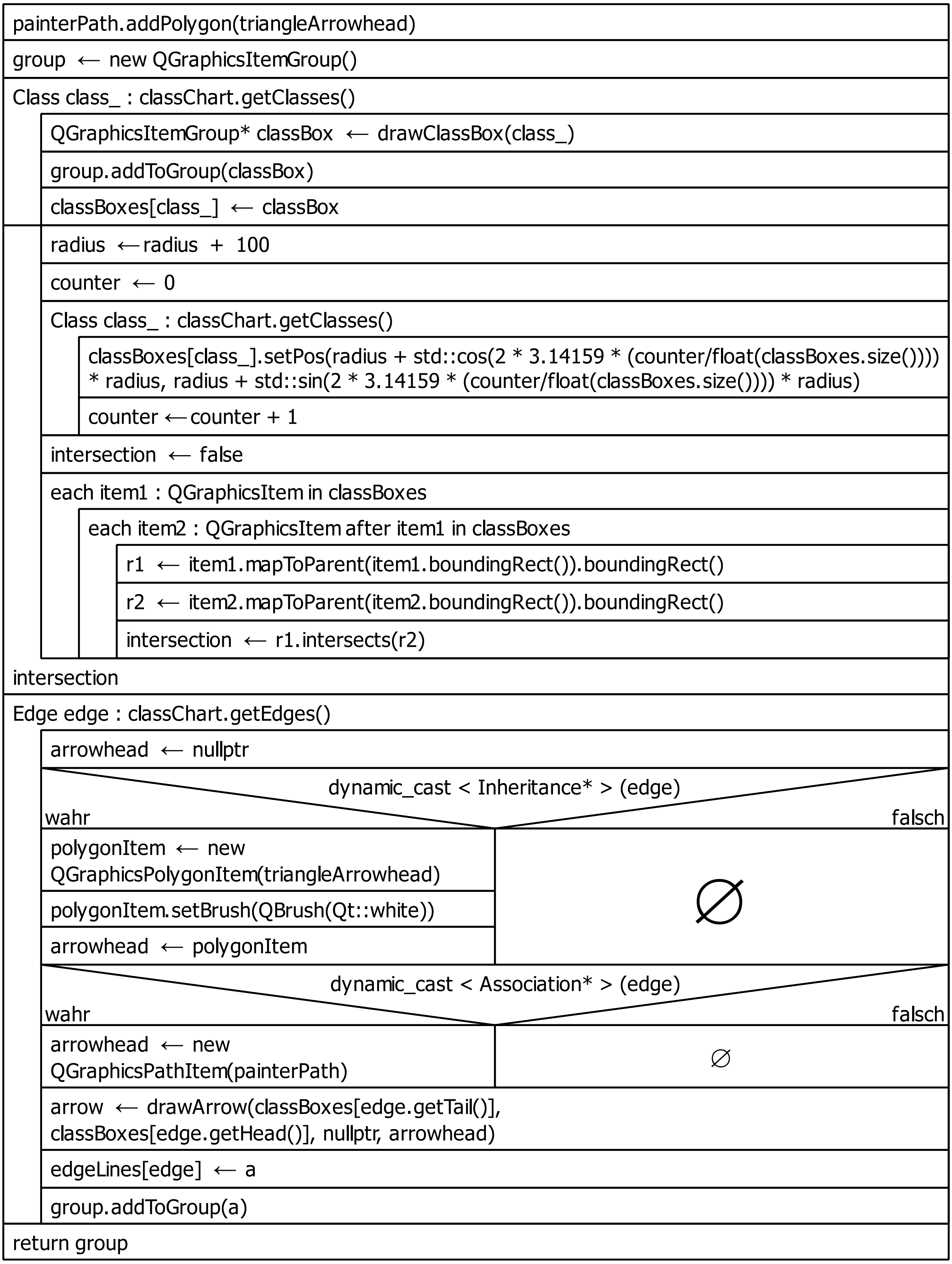
drawClassChart(classChart: ClassChart)

lokale/globale Variablen und Attribute:
group: QGraphicsItemGroup*
classBoxes: map<Class*,QGraphicsItemGroup*>
radius,counter: GZ
intersection: Boolean
edgeLines: map<Edge*,QGraphicsItemGroup*>
arrow: QGraphicsLineItem
painterPath: QPainterPath
arrowhead: QGraphicsItem
polygonItem: QGraphicsPolygonItem
Rect: r1,r2



```
painterPath.addPolygon(triangleArrowhead)
group ← new QGraphicsItemGroup()
Class class_ : classChart.getClasses()
    QGraphicsItemGroup* classBox ← drawClassBox(class_)
    group.addToGroup(classBox)
    classBoxes[class_] ← classBox
    radius ←radius + 100
    counter ← 0
    Class class_ : classChart.getClasses()
        classBoxes[class_].setPos(radius + std::cos(2 * 3.14159 * (counter/float(classBoxes.size())))
        * radius, radius + std::sin(2 * 3.14159 * (counter/float(classBoxes.size()))) * radius)
        counter ←counter + 1
    intersection ← false
    each item1 : QGraphicsItem in classBoxes
        each item2 : QGraphicsItem after item1 in classBoxes
            r1 ← item1.mapToParent(item1.boundingRect()).boundingRect()
            r2 ← item2.mapToParent(item2.boundingRect()).boundingRect()
            intersection ← r1.intersects(r2)
intersection
Edge edge : classChart.getEdges()
    arrowhead ← nullptr
    dynamic_cast < Inheritance* > (edge)
    wahr
        polygonItem ← new
        QGraphicsPolygonItem(triangleArrowhead)
        polygonItem.setBrush(QBrush(Qt::white))
        arrowhead ← polygonItem
    falsch
        ∅
    dynamic_cast < Association* > (edge)
    wahr
        arrowhead ← new
        QGraphicsPathItem(painterPath)
    falsch
        ∅
    arrow ← drawArrow(classBoxes[edge.getTail()],
    classBoxes[edge.getHead()], nullptr, arrowhead)
    edgeLines[edge] ← a
    group.addToGroup(a)
return group
```