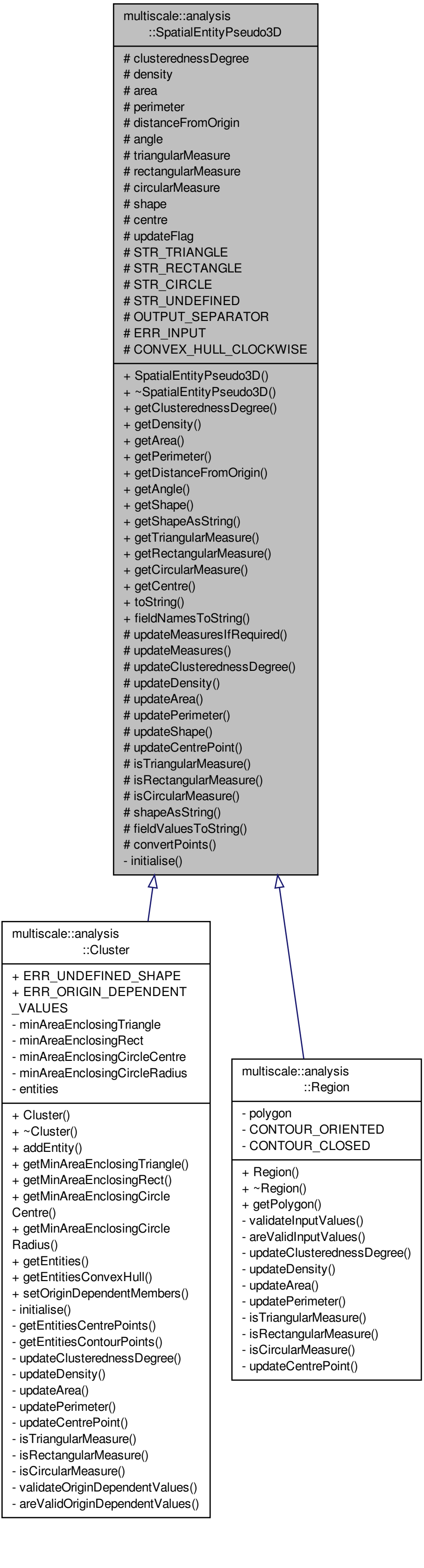
multiscale::analysis
::SpatialEntityPseudo3D
clusterednessDegree
density
area
perimeter
distanceFromOrigin
angle
triangularMeasure
rectangularMeasure
circularMeasure
shape
centre
updateFlag
STR_TRIANGLE
STR_RECTANGLE
STR_CIRCLE
STR_UNDEFINED
OUTPUT_SEPARATOR
ERR_INPUT
CONVEX_HULL_CLOCKWISE
+ SpatialEntityPseudo3D()
+ ~SpatialEntityPseudo3D()
+ getClusterednessDegree()
+ getDensity()
+ getArea()
+ getPerimeter()
+ getDistanceFromOrigin()
+ getAngle()
+ getShape()
+ getShapeAsString()
+ getTriangularMeasure()
+ getRectangularMeasure()
+ getCircularMeasure()
+ getCentre()
+ toString()
+ fieldNamesToString()
# updateMeasuresIfRequired()
# updateMeasures()
# updateClusterednessDegree()
# updateDensity()
# updateArea()
# updatePerimeter()
# updateShape()
# updateCentrePoint()
# isTriangularMeasure()
# isRectangularMeasure()
# isCircularMeasure()
# shapeAsString()
# fieldValuesToString()
# convertPoints()
- initialise()
multiscale::analysis
::Cluster
+ ERR_UNDEFINED_SHAPE
+ ERR_ORIGIN_DEPENDENT
_VALUES
- minAreaEnclosingTriangle
- minAreaEnclosingRect
- minAreaEnclosingCircleCentre
- minAreaEnclosingCircleRadius
- entities
+ Cluster()
+ ~Cluster()
+ addEntity()
+ getMinAreaEnclosingTriangle()
+ getMinAreaEnclosingRect()
+ getMinAreaEnclosingCircle
Centre()
+ getMinAreaEnclosingCircle
Radius()
+ getEntities()
+ getEntitiesConvexHull()
+ setOriginDependentMembers()
- initialise()
- getEntitiesCentrePoints()
- getEntitiesContourPoints()
- updateClusterednessDegree()
- updateDensity()
- updateArea()
- updatePerimeter()
- updateCentrePoint()
- isTriangularMeasure()
- isRectangularMeasure()
- isCircularMeasure()
- validateOriginDependentValues()
- areValidOriginDependentValues()
multiscale::analysis
::Region
- polygon
- CONTOUR_ORIENTED
- CONTOUR_CLOSED
+ Region()
+ ~Region()
+ getPolygon()
- validateInputValues()
- areValidInputValues()
- updateClusterednessDegree()
- updateDensity()
- updateArea()
- updatePerimeter()
- isTriangularMeasure()
- isRectangularMeasure()
- isCircularMeasure()
- updateCentrePoint()