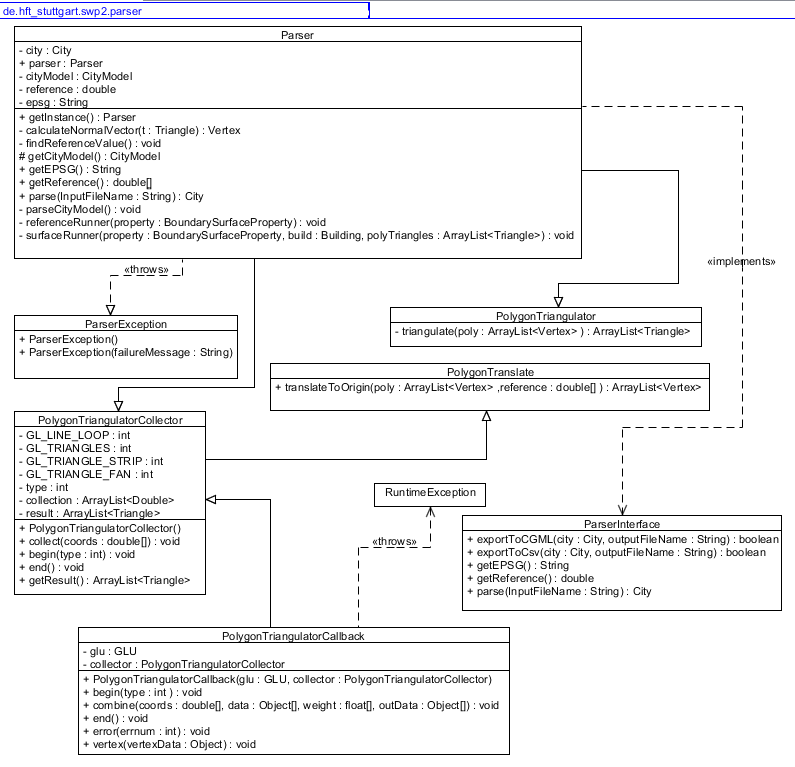
UML Klassendiagramm - Parser

Es wurde mit [UMLet](http://www.umlet.com/umlet_12_2/umlet_12.2.zip)  Version 12.2 erstellt.

Das Package besteht aus 7 Klassen und 1 Interface-Klasse. Die Hauptklasse ist die CGMLParser-Klasse. Es enthält 6 public Methoden und 2 private Methoden. Die Methode “parse“ liest die CityGML- Datei beziehungsweise liest die Gebäude sogenannten Buildings, die darin enthalten sind, transformiert deren Koordinaten zum Ursprung, wandelt dann die gewonnenen Polygone zur Vertices um und erstellt anschließend Building Objekt. Dieses Objekt enthält Building-id, Polygon und Vertices.

Es werden zusätzlich zu den Klassen im Parser-Klassendiagramm noch die Objekte aus dem

Package- Model benötigt:

* Building
* Vertex
* Triangle

Die Methode „exportToCGML“ exportiert Building-Objekte und speichert die Daten in einer neuen CityGML-Datei. Die gespeicherten Daten beinhalten Building-id und Building-Volume. Die Methode ist boolean und gibt somit true, wenn der Export erfolgreich verlauft, sonst false. Mit dem Attribut outputFileName kann man die exportierte Datei benennen. Die Methode wirft ParserException aus, wenn Fehler wie CityGMLWriteException, JAXBException auftreten.

Die Methode „exportToCsv“ funktioniert genauso wie die Methode „exportToCGML“ und gibt eine CSV-Datei mit Building-ID und Building-Volume. Es gibt true, wenn die Exportvorgang erfolgreich verläuft oder false, wenn etwas schief läuft.

Die Methode „getBuildingCoordinates“ sucht nach Buildings und deren Flächen beziehungsweise Koordinaten und anhand gefundener Polygone, speichert die x, y, z-Koordinaten aus jeder Oberflächen-Ecke, die sie findet in einem Array von Typ Double.

Die Methode „setBuildingCoordinates“ nützt „PolygonTriangulator“ um an die Polygone der Gebäude ran zukommen und nützt danach „PolygonTranslate“ um die Gebäude in die im Koordinatensystem auf die richtige Position zu verschieben. Die Koordinaten von triangulierten Polygonen speichert sie in ein ArrayList<Triangle>.

Mit der statischen Methode „translateToOrigin“ werden die Koordinatenachsen transformiert. Das heißt sie transformiert die komplexen Koordinaten so, dass die näher an den Ursprung des Koordinatensystems liegen. Das dient dazu um später die Koordinatendaten einfacher zu verarbeiten zu können.

Die public Methode „triangulate“ aus der Klasse PolygonTriangulator nutzt die JOGL-Bibliothek um aus den Polygonen aus dem Package Model Dreiecke zu erzeugen und speichert sie in ein ArryList<Triangle>. Die detaillierte Beschreibung des Algorithmus werde ich im nächsten Kapitel erläutern.

Parser-Paket enthält ein ParserInterface mit Methoden, „exportToCGML“, „exportToCSV“, „getEPSG“, „getReference“ und parse wie man aus dem UML-Diagramm oben entnehmen kann.

Die Methode „exportToCGML“ dient zur Exportierung der Gebäude in CityGML-Datei, die die Liste der Gebäude enthält. Durch die Parameterangabe outputFileName kann man bestimmen, wie die exportierte Datei später heißen soll. Die Methode gibt true, falls Exportvorgang korrekt verläuft, sonst false und wirft entweder NullPointerException oder JAXBException.

Ähnlich geht die Methode „exportToCSV“. Nur mit einer anderen Dateiendung beziehungsweise Dateiformat.

Die Methode „getEPSG“ gibt EPSG als String aus der GML-Datei und die Methode „getReference“ gibt die Referenzwerte aus, die für die Koordinaten-Transformationen später benötigt werden.