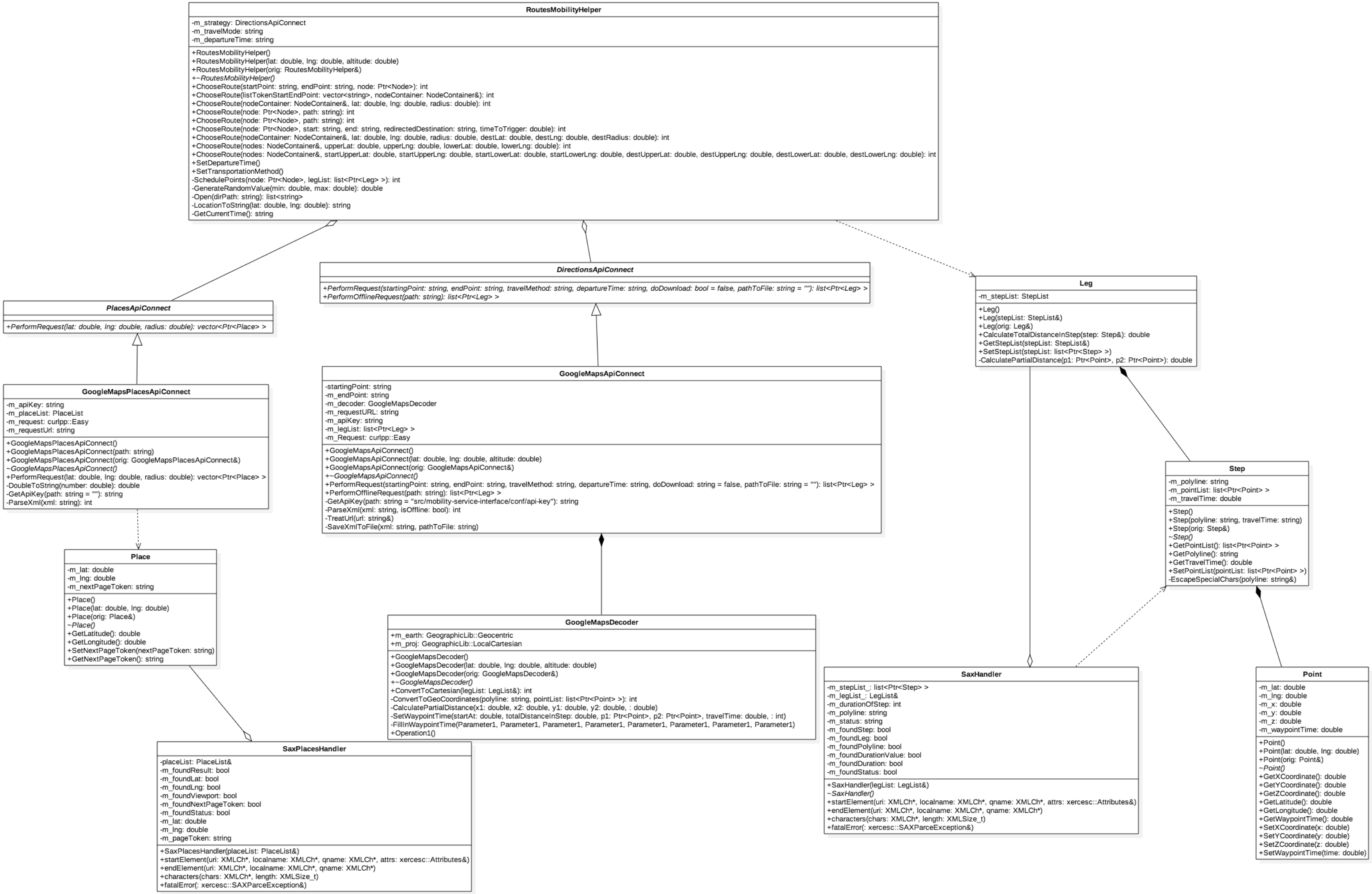
RoutesMobilityHelper
-m_strategy: DirectionsApiConnect
-m_travelMode: string
-m_departureTime: string
+RoutesMobilityHelper()
+RoutesMobilityHelper(lat: double, lng: double, altitude: double)
+RoutesMobilityHelper(orig: RoutesMobilityHelper&)
+~RoutesMobilityHelper()
+ChooseRoute(startPoint: string, endPoint: string, node: Ptr<Node>): int
+ChooseRoute(listTokenStartEndPoint: vector<string>, nodeContainer: NodeContainer&): int
+ChooseRoute(nodeContainer: NodeContainer&, lat: double, lng: double, radius: double): int
+ChooseRoute(node: Ptr<Node>, path: string): int
+ChooseRoute(node: Ptr<Node>, path: string): int
+ChooseRoute(node: Ptr<Node>, start: string, end: string, redirectedDestination: string, timeToTrigger: double): int
+ChooseRoute(nodeContainer: NodeContainer&, lat: double, lng: double, radius: double, destLat: double, destLng: double, destRadius: double): int
+ChooseRoute(nodes: NodeContainer&, upperLat: double, upperLng: double, lowerLat: double, lowerLng: double): int
+ChooseRoute(nodes: NodeContainer&, startUpperLat: double, startUpperLng: double, startLowerLat: double, startLowerLng: double, destUpperLat: double, destUpperLng: double, destLowerLat: double, destLowerLng: double): int
+SetDepartureTime()
+SetTransportationMethod()
-SchedulePoints(node: Ptr<Node>, legList: list<Ptr<Leg> >): int
-GenerateRandomValue(min: double, max: double): double
-Open(dirPath: string): list<string>
-LocationToString(lat: double, lng: double): string
-GetCurrentTime(): string
PlacesApiConnect
+PerformRequest(lat: double, lng: double, radius: double): vector<Ptr<Place> >
DirectionsApiConnect
+PerformRequest(startingPoint: string, endPoint: string, travelMethod: string, departureTime: string, doDownload: bool = false, pathToFile: string = ""): list<Ptr<Leg> >
+PerformOfflineRequest(path: string): list<Ptr<Leg> >
Leg
-m_stepList: StepList
+Leg()
+Leg(stepList: StepList&)
+Leg(orig: Leg&)
+CalculateTotalDistanceInStep(step: Step&): double
+GetStepList(stepList: StepList&)
+SetStepList(stepList: list<Ptr<Step> >)
-CalculatePartialDistance(p1: Ptr<Point>, p2: Ptr<Point>): double
GoogleMapsPlacesApiConnect
-m_apiKey: string
-m_placeList: PlaceList
-m_request: curlpp::Easy
-m_requestUrl: string
+GoogleMapsPlacesApiConnect()
+GoogleMapsPlacesApiConnect(path: string)
+GoogleMapsPlacesApiConnect(orig: GoogleMapsPlacesApiConnect&)
~GoogleMapsPlacesApiConnect()
+PerformRequest(lat: double, lng: double, radius: double): vector<Ptr<Place> >
-DoubleToString(number: double): double
-GetApiKey(path: string = ""): string
-ParseXml(xml: string): int
GoogleMapsApiConnect
-startingPoint: string
-m_endPoint: string
-m_decoder: GoogleMapsDecoder
-m_requestURL: string
-m_apiKey: string
-m_legList: list<Ptr<Leg> >
-m_Request: curlpp::Easy
+GoogleMapsApiConnect()
+GoogleMapsApiConnect(lat: double, lng: double, altitude: double)
+GoogleMapsApiConnect(orig: GoogleMapsApiConnect&)
+~GoogleMapsApiConnect()
+PerformRequest(startingPoint: string, endPoint: string, travelMethod: string, departureTime: string, doDownload: string = false, pathToFile: string = ""): list<Ptr<Leg> >
+PerformOfflineRequest(path: string): list<Ptr<Leg> >
-GetApiKey(path: string = "src/mobility-service-interface/conf/api-key"): string
-ParseXml(xml: string, isOffline: bool): int
-TreatUrl(url: string&)
-SaveXmlToFile(xml: string, pathToFile: string)
Step
-m_polyline: string
-m_pointList: list<Ptr<Point> >
-m_travelTime: double
+Step()
+Step(polyline: string, travelTime: string)
+Step(orig: Step&)
~Step()
+GetPointList(): list<Ptr<Point> >
+GetPolyline(): string
+GetTravelTime(): double
+SetPointList(pointList: list<Ptr<Point> >)
-EscapeSpecialChars(polyline: string&)
Place
-m_lat: double
-m_lng: double
-m_nextPageToken: string
+Place()
+Place(lat: double, lng: double)
+Place(orig: Place&)
~Place()
+GetLatitude(): double
+GetLongitude(): double
+SetNextPageToken(nextPageToken: string)
+GetNextPageToken(): string
GoogleMapsDecoder
+m_earth: GeographicLib::Geocentric
+m_proj: GeographicLib::LocalCartesian
+GoogleMapsDecoder()
+GoogleMapsDecoder(lat: double, lng: double, altitude: double)
+GoogleMapsDecoder(orig: GoogleMapsDecoder&)
+~GoogleMapsDecoder()
+ConvertToCartesian(legList: LegList&): int
-ConvertToGeoCoordinates(polyline: string, pointList: list<Ptr<Point> >): int
-CalculatePartialDistance(x1: double, x2: double, y1: double, y2: double, : double)
-SetWaypointTime(startAt: double, totalDistanceInStep: double, p1: Ptr<Point>, p2: Ptr<Point>, travelTime: double, : int)
-FillInWaypointTime(Parameter1, Parameter1, Parameter1, Parameter1, Parameter1, Parameter1, Parameter1, Parameter1)
+Operation1()
SaxHandler
-m_stepList_: list<Ptr<Step> >
-m_legList_: LegList&
-m_durationOfStep: int
-m_polyline: string
-m_status: string
-m_foundStep: bool
-m_foundLeg: bool
-m_foundPolyline: bool
-m_foundDurationValue: bool
-m_foundDuration: bool
-m_foundStatus: bool
+SaxHandler(legList: LegList&)
~SaxHandler()
+startElement(uri: XMLCh*, localname: XMLCh*, qname: XMLCh*, attrs: xercesc::Attributes&)
+endElement(uri: XMLCh*, localname: XMLCh*, qname: XMLCh*)
+characters(chars: XMLCh*, length: XMLSize_t)
+fatalError(: xercesc::SAXParceException&)
Point
-m_lat: double
-m_lng: double
-m_x: double
-m_y: double
-m_z: double
-m_waypointTime: double
+Point()
+Point(lat: double, lng: double)
+Point(orig: Point&)
~Point()
+GetXCoordinate(): double
+GetYCoordinate(): double
+GetZCoordinate(): double
+GetLatitude(): double
+GetLongitude(): double
+GetWaypointTime(): double
+SetXCoordinate(x: double)
+SetYCoordinate(y: double)
+SetZCoordinate(z: double)
+SetWaypointTime(time: double)
SaxPlacesHandler
-placeList: PlaceList&
-m_foundResult: bool
-m_foundLat: bool
-m_foundLng: bool
-m_foundViewport: bool
-m_foundNextPageToken: bool
-m_foundStatus: bool
-m_lat: double
-m_lng: double
-m_pageToken: string
+SaxPlacesHandler(placeList: PlaceList&)
+startElement(uri: XMLCh*, localname: XMLCh*, qname: XMLCh*, attrs: xercesc::Attributes&)
+endElement(uri: XMLCh*, localname: XMLCh*, qname: XMLCh*)
+characters(chars: XMLCh*, length: XMLSize_t)
+fatalError(: xercesc::SAXParceException&)