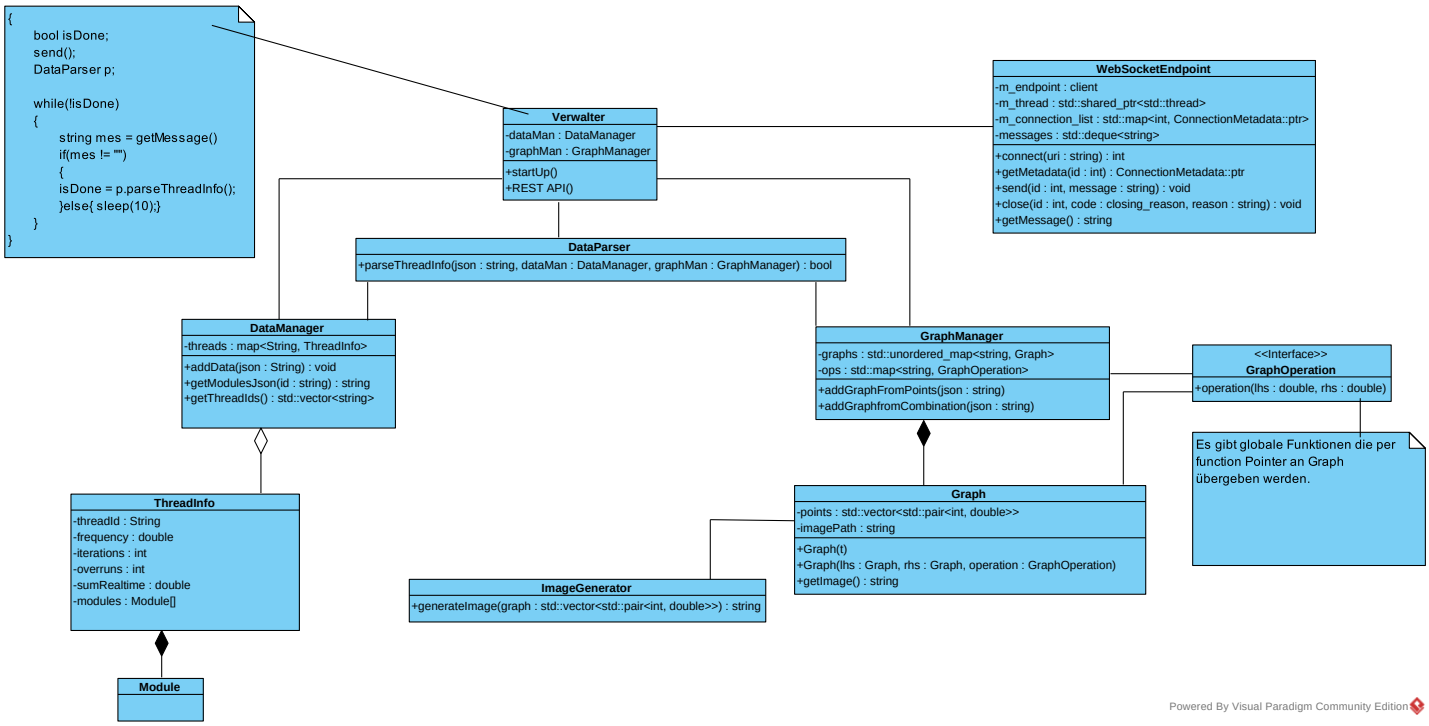
{
bool isDone;
send();
DataParser p;
while(!isDone)
{
string mes = getMessage()
if(mes != "")
{
isDone = p.parseThreadInfo();
}else{ sleep(10);}
}
}
Verwalter
-dataMan : DataManager
-graphMan : GraphManager
+startUp()
+REST API()
WebSocketEndpoint
-m_endpoint : client
-m_thread : std::shared_ptr<std::thread>
-m_connection_list : std::map<int, ConnectionMetadata::ptr>
-messages : std::deque<string>
+connect(uri : string) : int
+getMetadata(id : int) : ConnectionMetadata::ptr
+send(id : int, message : string) : void
+close(id : int, code : closing_reason, reason : string) : void
+getMessage() : string
DataParser
+parseThreadInfo(json : string, dataMan : DataManager, graphMan : GraphManager) : bool
DataManager
-threads : map<String, ThreadInfo>
+addData(json : String) : void
+getModulesJson(id : string) : string
+getThreadIds() : std::vector<string>
GraphManager
-graphs : std::unordered_map<string, Graph>
-ops : std::map<string, GraphOperation>
+addGraphFromPoints(json : string)
+addGraphfromCombination(json : string)
<<Interface>>
GraphOperation
+operation(lhs : double, rhs : double)
Es gibt globale Funktionen die per function Pointer an Graph übergeben werden.
ThreadInfo
-threadId : String
-frequency : double
-iterations : int
-overruns : int
-sumRealtime : double
-modules : Module[]
Graph
-points : std::vector<std::pair<int, double>>
-imagePath : string
+Graph(t)
+Graph(lhs : Graph, rhs : Graph, operation : GraphOperation)
+getImage() : string
ImageGenerator
+generateImage(graph : std::vector<std::pair<int, double>>) : string
Module
Powered By Visual Paradigm Community Edition