

Code structure diagram

Edge.h + Edge.cpp :

Edge is a class object with two members of type Vertex. Similarly to Vertex class it is used for validity checks to ease use in the main program, meaning it is also constructed with two valid and different Vertices and with correct Edge syntax.

Vertex.h + Vertex.cpp :

Vertex is a class object with a string member representing the name of the vertex.
All validity checks of the name are done by class functions and the constructor such that an object is constructed only if its name is valid, and errors are thrown accordingly.

Exceptions.h:

Contains all the exceptions used in all the files and code.
(all inherit from std::exception)

Exceptions

Edge

Vertex

Graph

CalcMemory

GCalc includes and utilizes all the shows files and classes to implement the program

GCalc

Gcalc.cpp :

The main program.

Graph.h + Graph.cpp :

Graph is the core class object where all graph specs are checked including syntax validity, Vertex and Edge duplication, and is only constructed with valid inputs and arguments.

This class is where all operators from section 2.3 are implemented.
A Graph object is defined by the Edges and Vertices contained in its two container members of type std::set.

CalcMemory.h + CalcMemory.cpp :

CalcMemory is the class object that serves as the "calculator memory" for the program where defined graph names are saved with Graph objects according to definition in a std::map container (key is the graph name and element is matching Graph object, and for lexicographical order the names are saved in the other container member of type std::set<string>).
All calculator operations (section 2.5) are implemented in the member functions.

In addition three files: graph.i , GraphP.h , GraphP.cpp are used for SWIG and the implementation of the Python library (GraphP.h includes the implementation of the 4.4 section functions and GraphO.cpp the implementation and nothing else).

Summary of Classes members:

Vertex : std::string name

Edge : Vertex v1 , Vertex v2

Graph : std::set<Vertex> vertices , std::set<Edge> edges

CalcMemory : std::set<std::string> graph_names , std::map<std::string,Graph> graphs_map