# Project design

Gcalc
map<string, shared_ptr<Graph> >
1. parse_to_vector
2. syntax_checker
3. tokenizer
4. read

GRAPH
map<string, set<string> >
Graph contractor&copy
operator=
Operator+
Operator^
Operator-
Operator*
operator!
print

Tokens
Type
Name
std::shared_ptr<Graph> ptr
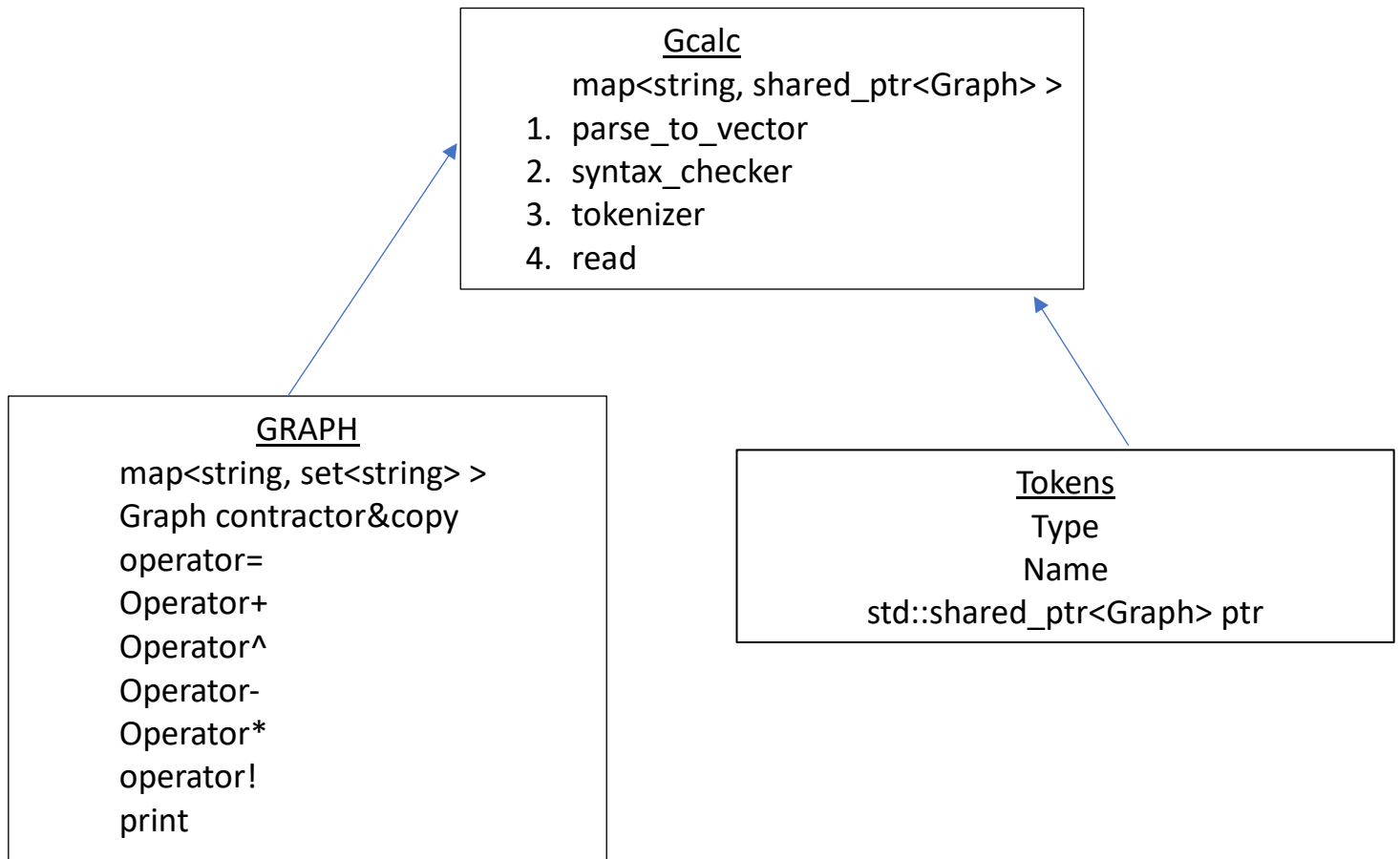
Explanation:
Gcalc receives from the user the mode of witch the program will preform. The string converts from string input to vector<string> (parse_to_vector function) – all spaces except in file name are eliminated. From there the string passes through a syntax checker to check if the input complies to the requirement of the project. Each string in the vector goes into an object of token type. The token class can describe function tokens, regex tokens or graph tokens. In the tokenizer function graph literals are converting into graph tokens. In addition the load function is executed in this stage.

Now we have a vector<shared_ptr<Token> > which passes to the read function. In that stage if the user entered a function like who/reset/delete it is executed. If the user entered an assignment operator or print/save – the input expression is evaluated from left to right (brackets are evaluated in a recursive way) using the operators in the Graph class . If no exception was thrown in the prosses  the value is assigned to the output graph (or saved/printed) . if the user entered an assignment operator then the output is added to the map_graph (map<string, shared_ptr<Graph> >).

The class Graph has a member of type map<string, set<string> >. In Witch each key represents an vertex in the graph, and each value of the key is a set of srtings that represents the vertex destinations of all the edges form witch the key is the source of.