# User Interface:

*Input:* A text file with a description of a D-ary tree, as follows:

**Example for binary tree:**

Root (gate): child\_1, child\_2

Child\_1 (gate): child\_11, child\_12

Child\_2 (gate): child\_21, child\_22

Etc…

*Output:* A text file contains the description of a D-ary tree (describes the error resilient circuit).

# Classes List:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Extends** | **Data members** | **Methods** |
| Node |  | data  children\_pointers [D] (Node\*)  parent\_pointer (Node\*) | gets and sets  c’tor and d’tor |
| LogicGateNode | Node | Type (or/and) | Int calculate\_output(int x, int y)  gets and sets  c’tor and d’tor |
| PartyNode | Node | type (ALICE/BOB)  sub\_formula (LogicGateNode\*)  input (x/y) | gets and sets  c’tor and d’tor |
| DaryTree |  | root (Node\*)  fan\_in (int) | gets and sets  c’tor and d’tor  bool add\_node  bool delete\_node |
| FormulaTree | DaryTree | Int num\_of\_variables | C’tor(input file)  Bool calculate\_formula(Node\* z) |
| ProtocolTree | DaryTree |  | C’tor  Void apply\_KW(formulaTree\* root)  Void apply\_EGH()  Void apply\_reverse\_KW() |

# Main:

Purpose: an entry point for the application. Manage the conversion of the logic circuit into a resilient one. In addition, the main communicates with the user through the command line.

Inputs: none

Outputs: none

Calls:

* FormulaTree (create an instance of FormulaTree).
* ProtocolTree.apply\_KW(formulaTree\* root)
* ProtocolTree.apply\_EGH()
* ProtocolTree.apply\_reverse\_KW()

# DaryTree:

Purpose: stores a full D-ary tree.

Inputs: text file describes a D-ary tree structure.

Outputs: none

Calls: none

# FormulaTree:

Purpose: stores the **logic circuit** as a full D-ary tree. In contrast to the DaryTree which stores only a tree structure, the formulaTree describe a circuit as follows: each node has a gate type: and/or and a pointer to a sub-FormulaTree in order to calculate the output of the node for an input x/y.

Input: text file describes a D-ary tree structure with logic gates on each node.(description of logical circuit).

Output: none

Calls: none

# ProtocolTree:

Purpose: converts a formulaTree into an error resilient formula[[1]](#footnote-1).

Input: a FormulaTree

Output: a resilient FormulaTree

Calls: FormulaTree. calculate\_formula(Node\* z)

1. As described in the first report: <first_project_report_161216.docx> [↑](#footnote-ref-1)