```
/*************************************************************************
                      GraphGenerator  -  description
                             -------------------
    begin                : 23/11/2015
    copyright            : (C) 2015 by Edern Haumont & Nicolas Six
*************************************************************************/

//-- Realisation of the class GraphGenerator (file GraphGenerator.cpp) ---

//---------------------------------------------------------------- INCLUDE

//------------------------------------------------------- Personal include
#include "GraphGenerator.h"
#include "config.h"

// don't use: "using namespace std;" to keep clear that we use std and
// not any other library and by the same way keeping ready to use an other
// library than the std.

//----------------------------------------------------------------- PUBLIC

//--------------------------------------------------------- Public methods

// Constructor
GraphGenerator::GraphGenerator(const std::string &filePath)
{
    // Algorithm :
    // creating a new one graph file, deleting the older one if the file already exist
    graphFileStream.open(filePath,std::ios::out | std::ios::trunc);
    if(!graphFileStream)
    {
        std::cerr << "erreur lors de l'ouverture du fichier: " << filePath << std::endl;
    }
    else
    {
        graphFileStream << "digraph {" << std::endl;
    }
}
// Destructor
GraphGenerator::~GraphGenerator()
{
    if(graphFileStream)
    {
        graphFileStream << "}" << std::endl;
        graphFileStream.close();
    }
}

int GraphGenerator::addNodeToGraph(const std::string &nodeName)
{
    if(!graphFileStream)
    {
        std::cerr << "erreur sur le fichier en écriture" << std::endl;
        return FILE_ERROR;
    }
    else
    {
        graphFileStream << "       " << transformToNodeName(nodeName) << " [label=\"" << nodeName <<
"\"];" << std::endl;
    }
    return 0;
}
int GraphGenerator::addLinkToGraph(const std::string &nodeNameFrom, const std::string &nodeNameTo,
const std::string &linkLabel)
{
    addNodeToGraph(nodeNameFrom);
    addNodeToGraph(nodeNameTo);
    if(!graphFileStream)
    {
        std::cerr << "erreur sur le fichier en écriture" << std::endl;
        return FILE_ERROR;
    }
    else
    {
        graphFileStream << "       " << transformToNodeName(nodeNameTo) << " -> ";
        graphFileStream << transformToNodeName(nodeNameFrom) << " [label=" << linkLabel << "];" <<
std::endl;
    }
    return 0;
}

//----------------------------------------------------------- Private methods

std::string GraphGenerator::transformToNodeName(const std::string &nonUsableName) const
{
    std::string ret ="";
    bool add;
    for (unsigned i=0; i<nonUsableName.length(); i++)
    {
        add = true;
        for(unsigned n=0 ; n<INVALID_CHAR.length() ; n++)
        {
            if(nonUsableName.at(i)==INVALID_CHAR.at(n))
            {
                add = false;
                break;
            }
        }
        if(add)
        {
            ret.push_back(nonUsableName.at(i));
        }
        else
        {
            ret.push_back(nonUsableName.at(i)%26+'a');
        }
    }
    return ret;
}
```