#ifndef \_CSVPARSER\_HPP\_

# define \_CSVPARSER\_HPP\_

# include <stdexcept>

# include <string>

# include <vector>

# include <list>

# include <sstream>

namespace csv

{

class Error : public std::runtime\_error

{

public:

Error(const std::string &msg):

std::runtime\_error(std::string("CSVparser : ").append(msg))

{

}

};

class Row

{

public:

Row(const std::vector<std::string> &);

~Row(void);

public:

unsigned int size(void) const;

void push(const std::string &);

bool set(const std::string &, const std::string &);

private:

const std::vector<std::string> \_header;

std::vector<std::string> \_values;

public:

template<typename T>

const T getValue(unsigned int pos) const

{

if (pos < \_values.size())

{

T res;

std::stringstream ss;

ss << \_values[pos];

ss >> res;

return res;

}

throw Error("can't return this value (doesn't exist)");

}

const std::string operator[](unsigned int) const;

const std::string operator[](const std::string &valueName) const;

friend std::ostream& operator<<(std::ostream& os, const Row &row);

friend std::ofstream& operator<<(std::ofstream& os, const Row &row);

};

enum DataType {

eFILE = 0,

ePURE = 1

};

class Parser

{

public:

Parser(const std::string &, const DataType &type = eFILE, char sep = ',');

~Parser(void);

public:

Row &getRow(unsigned int row) const;

unsigned int rowCount(void) const;

unsigned int columnCount(void) const;

std::vector<std::string> getHeader(void) const;

const std::string getHeaderElement(unsigned int pos) const;

const std::string &getFileName(void) const;

public:

bool deleteRow(unsigned int row);

bool addRow(unsigned int pos, const std::vector<std::string> &);

void sync(void) const;

protected:

void parseHeader(void);

void parseContent(void);

private:

std::string \_file;

const DataType \_type;

const char \_sep;

std::vector<std::string> \_originalFile;

std::vector<std::string> \_header;

std::vector<Row \*> \_content;

public:

Row &operator[](unsigned int row) const;

};

}

#endif /\*!\_CSVPARSER\_HPP\_\*/