版本M.0.1

——2020年11月22日

代码介绍：

此版本主要对底层使用的数据结构进行了更换，将WangpaiStack换成了STL中的stack。更换的目的是为了此项目能和现代技术接轨。

* 因将WangpaiStack换成了STL中的stack，导致要进行的函数改变有：

1. WangpaiList::isemp() ⇨ empty()
2. WangpaiStack::getop() ⇨ top()
3. WangpaiStack::pop() ⇨ top()+pop()
4. WangpaiList::count() ⇨ size()
5. WangpaiList:: operator[] (...) ⇨ stack\_auxCla::index (...)
6. WangpaiList:: search(...) ⇨ stack\_auxCla::find (...)
7. WangpaiList:: deleteall() ⇨ stack\_auxCla:: empty\_stack (...)
8. WangpaiList:: operator+= (...) ⇨ stack\_auxCla:: stackCat(...)

* 改成stack之后，底层的元素序号规定为靠近栈顶的序号值小，栈顶的序号值为0（而之前栈顶的序号值规定为1）
* 因为使用了stack，而加入了stack\_auxCla类用于弥补STL中的stack没有的一些功能。stack\_auxCla类的方法如下：

1. template <typename DataType>

static vector<DataType> stack2vector(std::stack<DataType> this\_stack)

1. template <typename DataType>

static DataType index(const std::stack<DataType>& this\_stack, int seqnum)

1. template <typename DataType>

static int find(const std::stack<DataType>& this\_stack, const DataType& data)

1. template <typename DataType>

static int empty\_stack(std::stack<DataType>& this\_stack)

1. template <typename DataType>

static std::stack<DataType>& stackCat(std::stack<DataType>& this\_stack, const std::stack<DataType>& other\_stack)

关系项依赖图：

无

推荐编译器：VS 2013

其它源文件：arithoper\_M.0.1.cpp

头文件：stack\_auxiliary.h

arithoper\_M.0.1.h

截止至2020年11月22日，在核心的代码中，

总的代码行数为：（按前面从上到下列出的文件顺序给出）

574+110+42=726

主程序示例：

#include "arithoper\_M.0.0.1.h"

int main()

{

using namespace arithoper;

evalexpre();

system("pause");

return 0;

}

程序屏幕输入示例：

234234.623\*6345-234/1234+234\*(254-45.23542)=

一个屏幕输出结果的片段：

略