#include <iostream>

#include <assert.h>

#include <sstream>

#include "complex.h"

#include "matrix.h"

#include "polynomial.h"

#include <string>

#include "fraction.h"

void complex\_calc() {

static char L = 'E';

while (1) {

Complex a, b, c;

std::cout << "Enter first Number \n";

std::cin >> a;

std::cout << "Enter second Number \n";

std::cin >> b;

std::cout << "Enter Action('-','+','/','\*','E') :";

std::cin >> L;

if (L == 'E' || L == 'e')

{

break;

}

std::cout << std::endl;

switch (L)

{

case '+':c = a + b; break;

case '-':c = a - b; break;

case '\*':c = a \* b; break;

case '/':c = a / b; break;

default:throw std::invalid\_argument("unknown symbol");

}

std::cout << c << "\n";

}

}

template<typename T>void matrix\_calc() {

static char L = 'E';

while (1) {

matrix<T> a, b, ans\_out;

T ans\_out\_T;

std::cout << "Enter first Number \n";

std::cin >> a;

std::cout << "\n" << a << "\n";

std::cout << "Enter second Number \n";

std::cin >> b;

std::cout << "\n" << b << "\n";

std::cout << "Enter Action('-','+','/','\*','r','E','d') :\n";

std::cin >> L;

if (L == 'E' || L == 'e')

{

break;

}

std::cout << std::endl;

switch (L)

{

case '+':ans\_out = a + b; break;

case '-':ans\_out = a - b; break;

case '\*':ans\_out = a \* b; break;

case '/':ans\_out = a / b; break;

case 'd':ans\_out\_T = (a.determinant()); break;

case 'r':ans\_out = a.inverse\_M(); break;

default:throw std::invalid\_argument("unknown symbol");

}

if (L == 'd') {

std::cout << ans\_out\_T << "\n";

}

else

{

std::cout << ans\_out << "\n";

}

}

}

void main\_calc\_menu() {

std::cout << "Enter calc type \n";

std::cout << "mfpint-matrix\_calc<fraction<polynomial<int>>>()\n";

std::cout << "mfpdouble-matrix\_calc<fraction<polynomial<double>>>()\n";

std::cout << "mpint-matrix\_calc<polynomial<int>>()\n";

std::cout << "mpdouble-matrix\_calc<polynomial<double>>()\n";

std::cout << "mdouble-matrix\_calc<double>()\n";

std::cout << "mint-matrix\_calc<int>()\n";

std::cout << "cmpl-complex\_calc()\n";

std::cout << "\ncalc type:";

std::string type;

std::cin >> type;

if (type == "complex\_calc()" || type == "cmpl")complex\_calc();

else if (type == "matrix\_calc<fraction<polynomial<int>>>()" || type == "mfpint" || type == "matfrpolint") matrix\_calc<fraction<polynomial<int>>>();

else if (type == "matrix\_calc<fraction<polynomial<double>>>()" || type == "mfpdouble" || type == "matfrpoldouble") matrix\_calc<fraction<polynomial<double>>>();

else if (type == "matrix\_calc<polynomial<int>>()" || type == "mpint" || type == "matpolint") matrix\_calc<polynomial<int>>();

else if (type == "matrix\_calc<polynomial<double>>()" || type == "mpdouble" || type == "matpoldouble") matrix\_calc<polynomial<double>>();

else if (type == "matrix\_calc<double>()" || type == "mdouble" || type == "matdouble") matrix\_calc<double>();

else if (type == "matrix\_calc<int>()" || type == "mint" || type == "matint") matrix\_calc<int>();

else throw std::invalid\_argument("unknown calculator type or an error in the name");

}

int main() {

try

{

std::string type;

restart:

main\_calc\_menu();

std::cout << "Do you want to get out?(yes/no)\n:";

std::cin >> type;

if (type == "yes" || type == "y" || !(type == "no" || type == "n"))std::exit(0);

if (type == "no" || type == "n")goto restart;

}

catch (std::exception ex)

{

std::cout << "exeption!!What:" << ex.what();

}

catch (...)

{

std::cout << "unknown error";

}

system("pause");

return 0;

}