Triangle h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Triangle

{

private:

double A;

double B;

double C;

public:

Triangle();

Triangle(const double& a, const double& b, const double& c);

Triangle(const Triangle& T);

~Triangle();

double getA() const;

double getB() const;

double getC() const;

void printInfo() const;

double& getA();

double& getB();

double& getC();

void setA(const double& a);

void setB(const double& b);

void setC(const double& c);

double Perimeter() const;

double Square() const;

//void readFrom(istream& in);

//void writeTo(ostream& out) const;

//bool operator<(const Triangle& T)const;

};

istream& operator>>(istream& in, Triangle& T);

ostream& operator<<(ostream& out, const Triangle& T);

void printTriangleArray(Triangle\* arr, int n);

//double totalTrSquare(Triangle\* fileArray, int n);

Main

#include<iostream>

#include <fstream>

#include "triangle.h"

using namespace std;

int main()

{

Triangle T1;

Triangle T2(3, 4, 5);

Triangle T3(T2);

cout << "\n";

cout << "Triangle info\n";

cout << "a = " << T2.getA() << "\nb = " << T2.getB() << "\nc = " << T2.getC() << "\n";

T1.printInfo();

cout << "\nVia getters\n";

T1.getA() = 6;

T1.getB() = 8;

T1.getC() = 10;

T1.printInfo();

cout << "\n";

T3.setA(9);

T3.setB(12);

T3.setC(15);

T3.printInfo();

double a1;

cout << "\nPlease, enter the first new side of the triangle: ";

cin >> a1;

T3.setA(a1);

T3.printInfo();

double a2;

cout << "\nPlease, enter the second new side of the triangle: ";

cin >> a2;

T3.setA(a2);

T3.printInfo();

double a3;

cout << "\nPlease, enter the third new side of the triangle: ";

cin >> a3;

T3.setA(a3);

T3.printInfo();

Triangle T4;

cout << "\nPlease, enter triangle info:\n";

cin >> T4;

cout << "\nHere is your triangle" << T4 << "\n";

double p;

p = T4.Perimeter();

cout << "Perimeter = " << p << "\n";

double s;

s = T4.Square();

cout << "Square = " << s << "\n";

//cout << "Comparison\n\n";

ifstream iFile("triangles\_info.txt");

int size; iFile >> size;

Triangle\* fileArray = new Triangle[size];

for (int i = 0; i < size; ++i)

{

iFile >> fileArray[i];

}

iFile.close();

cout << "\nTriangles from file\n";

printTriangleArray(fileArray, size);

//double totalSquare = totalTrSquare(Triangle \* fileArray, n);

//cout << "Total square: " << totalTrSquare << "\n";

cout << "\n";

delete[]fileArray;

return 0;

}

Triangle

#include "triangle.h"

Triangle::Triangle() : A(), B(), C()

{

//cout << "Default Triangle\n";

}

Triangle::Triangle(const double& a, const double& b, const double& c) : A(a), B(b), C(c)

{

//cout << "Parameters Triangle\n";

}

Triangle::Triangle(const Triangle& T) : A(T.A), B(T.B), C(T.C)

{

//cout << "Copy Triangle\n";

}

Triangle::~Triangle()

{

//cout << "Destroy Triangle\n"<< A <<"\n";

}

double Triangle::getA() const

{

return this->A;

}

double Triangle::getB() const

{

return this->B;

}

double Triangle::getC() const

{

return this->C;

}

void Triangle::printInfo() const

{

cout << "Triangle info: ";

//cout << A << " " << B << " " << C << " \n";

cout << "\na = " << A << "\nb = " << B << "\nc = " << C << "\n";

}

double &Triangle::getA()

{

return A;

}

double& Triangle::getB()

{

return B;

}

double& Triangle::getC()

{

return C;

}

void Triangle::setA(const double& a)

{

if (a <= 0)

{

cout << "Error! The side of the triangle cannot be " << a << "\n";

return;

}

A = a;

}

void Triangle::setB(const double& b)

{

if (b <= 0)

{

cout << "Error! The side of the triangle cannot be " << b << "\n";

return;

}

B = b;

}

void Triangle::setC(const double& c)

{

if (c <= 0)

{

cout << "Error! The side of the triangle cannot be " << c << "\n";

return;

}

C = c;

}

istream& operator>>(istream& in, Triangle& T)

{

in >> T.getA() >> T.getB() >>T.getC();

//T.readFrom(in);

return in;

}

ostream& operator<<(ostream& out, const Triangle& T)

{

out << "\na= " <<T.getA() << "\nb= " << T.getB() << "\nc= " << T.getC();

//T.writeTo(out);

return out;

}

double Triangle::Perimeter() const

{

double p = A + B + C;

return p;

}

double Triangle::Square() const

{

double p = (A + B + C)/2;

double s = sqrt(p \* (p - A) \* (p - B) \* (p - C));

return s;

}

//bool Triangle::operator<(const Triangle& T)const

//{

// return this->Perimeter() < T.Perimeter();

//}

//double totalTrSquare(Triangle\* fileArray, int n)

//{

// double sum = 0;

// for (int i = 0; i < n; ++i)

// {

// sum += fileArray[i].Square;

// }

//return sum;

//}

triangles\_info

4

6 8 10

12 16 20

4 3 2

15 20 25