Main

#include<iostream>

#include "triangle.h"

using namespace std;

int main()

{

Triangle T1;

Triangle T2(3.0, 4.0, 5.0);

Triangle T3(T2);

cout << "\n";

T1.printInfo();

cout << "\n";

T3.printInfo();

cout << "\n";

cout << "Changing the value of fields\n\n";

T1.getOne() = 6.0;

T1.getTwo() = 8.0;

T1.getThree() = 10.0;

T1.printInfo();

T3.setOne(9.0);

T3.setTwo(12.0);

T3.setThree(15.0);

T3.printInfo();

double val1;

cout << "\nEnter new value a: ";

cin >> val1;

T3.setOne(val1);

T3.printInfo();

double val2;

cout << "\nEnter new value b: ";

cin >> val2;

T3.setTwo(val2);

T3.printInfo();

double val3;

cout << "\nEnter new value c: ";

cin >> val3;

T3.setThree(val3);

T3.printInfo();

Triangle T4;

cout << "\nEnter the values of the sides of the triangle:\n";

cin >> T4;

cout << "This is a triangle with values\n" << T4 << "\n";

cout << "\n";

return 0;

}

Triangle

#include "triangle.h"

Triangle::Triangle() : one(), two(), three()

{

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

}

Triangle::Triangle(const double& a, const double& b, const double& c) : one(a), two(b), three(c)

{

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

}

Triangle::Triangle(const Triangle& T) : one(T.one), two(T.two), three(T.three)

{

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

}

Triangle::~Triangle()

{

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

}

double Triangle::getOne() const

{

return this->one;

}

double Triangle::getTwo() const

{

return this->two;

}

double Triangle::getThree() const

{

return this->three;

}

void Triangle::printInfo() const

{

cout << "Triangle info\n";

cout << "a = " << one << ";" << " b = " << two << ";" << " c = " << three << "." << "\n";

}

double& Triangle::getOne()

{

return one;

}

double& Triangle::getTwo()

{

return two;

}

double& Triangle::getThree()

{

return three;

}

void Triangle::setOne(const double& oneValue)

{

if (oneValue <= 0)

{

cout << "Error! Value " << oneValue << "the side of the triangle cannot have\n";

return;

}

one = oneValue;

}

void Triangle::setTwo(const double& twoValue)

{

if (twoValue <= 0)

{

cout << "Error! Value " << twoValue << "the side of the triangle cannot have\n";

return;

}

two = twoValue;

}

void Triangle::setThree(const double& threeValue)

{

if (threeValue <= 0)

{

cout << "Error! Value " << threeValue << " side of the triangle cannot have\n";

return;

}

three = threeValue;

}

void Triangle::readFrom(istream& in)

{

in >> one >> two >> three;

}

void Triangle::writeTo(ostream& out)

{

out << one << " " << two << " " << three << " ";

}

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

{ //через setter

//double a; in >> a; T.setOne(a);

//double b; in >> b; T.setTwo(b);

//double c; in >> c; T.setThree(c);

//через getter

in >> T.getOne()>> T.getTwo()>>T.getThree();

////через method

//T.readFrom(in);

return in;

}

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

{ //через getter

out << T.getOne() << " " << T.getTwo() << " " << T.getThree();

////через method

//T.writeTo(out);

return out;

}

Triangle h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Triangle

{

private:

double one;

double two;

double three;

public:

Triangle();

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

Triangle(const Triangle& T);

~Triangle();

void printInfo() const;

double getOne() const;

double getTwo() const;

double getThree() const;

double& getOne();

double& getTwo();

double& getThree();

void setOne(const double& oneValue);

void setTwo(const double& twoValue);

void setThree(const double& threeValue);

void readFrom(istream& in);

void writeTo(ostream& out);

};

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

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