Балтийский государственный технический университет «ВОЕНМЕХ»

им. Д. Ф. Устинова

Кафедра И5

«Информационные системы и программная инженерия»

Практическое задание № 2

по дисциплине «Программирование на ЯВУ»

на тему «Классы: основные понятия и определения»

Выполнил:

Студент Дубровский В.И.

Группа И582

Преподаватель:

Спирин Д.О.

Санкт-Петербург

2019

**Вариант 8**

Цель работы – изучить понятие класса, механизмы работы с классами, научиться обеспечивать вызов методов и обращение к полям классов, изменять видимость компонент в определении класса с использованием спецификаторов доступа, перегружать операции и использовать дружественные функции.

Задание:

Класс «Треугольник». Поля: длины сторон. Методы: вычисление площади, радиусов вписанной и описанной окружностей, определение типа (остроугольный, прямоугольный, тупоугольный), перегрузка операции сравнения на равенство «==».

Текст программы:

main.cpp

#include "Triangle.h"

#include <iostream>

#include <conio.h>

#include <locale.h>

#include <vector>

using namespace std;

void menu\_vector(void);

void menu\_triangle(void);

void clear\_enter(void);

int main()

{

setlocale(LC\_ALL, "Rus");

vector<Triangle> list;

bool quit = true;

int c = -1;

int i;

do {

menu\_vector();

while ((c < '0' || c > '6') && (c = \_getch()));

if (c == '0') break;

system("cls");

switch (c)

{

case '1': {

for (i=0; i<list.size(); i++)

{

cout << i+1 << ".\t" << list[i] << endl;

}

if (i == 0) cout << "Empty!" << endl;

break;

}

case '2': {

Triangle temp;

bool flag=true;

while (!temp.check()) {

while (!(cin >> temp)) {

cout << "Incorrect enter!" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

list.push\_back(temp);

break;

}

case '3': {

if (list.size() == 0) {

cout << "Empty";

break;

}

int num=-1;

cout << "Enter num of triangle: ";

while (!(cin >> num) || (num > list.size() || num < 1)) {

cout << "Incorrect num" << endl;

cout << "Enter num of triangle: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

auto iter = list.begin();

advance(iter, num-1);

list.erase(iter);

break;

}

case '4': {

if (list.size() < 2) {

cout << "Not enough triangles";

break;

}

int num = -1;

cout << "Enter num of first triangle: ";

while (!(cin >> num) || (num > list.size() || num < 1)) {

cout << "Incorrect num" << endl;

cout << "Enter num of triangle: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

num--;

int num2 = -1;

cout << "Enter num of second triangle: ";

while (!(cin >> num2) || (num2 > list.size() || num2 < 1)) {

cout << "Incorrect num" << endl;

cout << "Enter num of triangle: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

num2--;

if (list[num] == list[num2]) cout << "Equal" << endl;

else cout << "Not equal" << endl;

break;

}

case '5': {

int num;

cout << "Enter num of triangle: ";

while (!(cin >> num) || (num > list.size() || num < 1)) {

cout << "Incorrect num" << endl;

cout << "Enter num of triangle: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

num--;

do {

cout << list[num];

menu\_triangle();

c = -1;

while ((c = \_getch()) && (c < '1' || c>'4') && (c!=13));

if (c == 13) break;

system("cls");

switch (c)

{

case '1':

cout << list[num].area() << endl;

break;

case '2':

cout << list[num].Rin() << endl;

break;

case '3':

cout << list[num].Rout() << endl;

break;

case '4':

cout << list[num].type().c\_str() << endl;

}

} while (true);

break;

}

case '6':

double a, b, c;

cout << "Enter a, b, c" << endl;

do {

while (!(cin >> a >> b >> c))

{

cout << "Incorrect num" << endl;

cout << "Enter a, b, c" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

Triangle temp(a, b, c);

if (temp.check()) {

list.push\_back(temp);

break;

}

else {

cout << "Incorrect triangle" << endl;

}

} while (true);

}

clear\_enter();

c = -1;

} while (true);

return 0;

}

void menu\_vector(void)

{

cout << "1. Show list of triangles" << "\n"

<< "2. Add triangle in list with \"<<\"" << "\n"

<< "3. Delete triangle from list" << "\n"

<< "4. Compare on equal of two triangles" << "\n"

<< "5. Operations with triangle" << "\n"

<< "6. Add with param" << "\n"

<< "0. Exit " << endl;

}

void menu\_triangle(void)

{

cout << "1. Compute area of triangle" << "\n"

<< "2. Find radius of incircle" << "\n"

<< "3. Find radius of circumcircle" << "\n"

<< "4. Type of triangle" << endl;

}

void clear\_enter(void)

{

while (\_getch() != 13);

system("cls");

}

Triangle.h

#pragma once

#include <iostream>

#include <math.h>

using namespace std;

class Triangle

{

private:

/\*struct sides {

double a;

double b;

double c;

sides(double a=0, double b=0, double c=0) : a(a), b(b), c(c)

{};

}sides;\*/

double a;

double b;

double c;

public:

Triangle(double a = 0, double b = 0, double c = 0) : a(a), b(b), c(c)

{

cout << "Вызов конструктора по умолчанию" << endl;

};

Triangle(double a, double b, double c) : a(a), b(b), c(c)

{

cout << "Вызов конструктора с параметрами" << endl;

};

Triangle(const Triangle &copy) : a(copy.a), b(copy.b), c(copy.c)

{

cout << "Вызов конструктора копирования" << endl;

};

~Triangle()

{

cout << "Вызов деструктора" << endl;

};

double area(void);

double Rin(void);

double Rout(void);

const bool check(void);

string type(void);

friend istream &operator >> (istream &stream, Triangle &p);

friend ostream &operator << (ostream &stream, const Triangle &p);

const bool &operator == (const Triangle &right);

friend const bool &operator == (const Triangle &left, const Triangle &right);

};

Triangle.cpp

#include "Triangle.h"

double Triangle::area(void)

{

cout << "Площадь треугольника" << endl;

double p = (a + b + c) / 2;

return sqrt(p\*(p - a)\*(p - b)\*(p - c));

}

double Triangle::Rin(void)

{

cout << "Радиус вписанной окружности" << endl;

return (area() \* 2) / (a + b + c);

}

double Triangle::Rout(void)

{

cout << "Радиус описанной окружности" << endl;

return a\*b\*c / (4 \* area());

}

const bool Triangle::check(void)

{

cout << "Вызов проверки, существует ли такой треугольник." << endl;

if (a <= 0 || b <= 0 || c <= 0) return false;

return a + b > c && b + c > a && c + a > b;

}

std::string Triangle::type(void)

{

cout << "Определение типа треугольника" << endl;

if (a\*a == b\*b + c\*c || b\*b == c\*c + a\*a ||

c\*c == a\*a + b\*b) return "Right triangle";

else if (a\*a > b\*b + c\*c || b\*b > c\*c + a\*a ||

c\*c > a\*a + b\*b) return "Obtuse triangle";

else return "Acute triangle";

}

const bool & Triangle::operator==(const Triangle & right)

{

return this->a == right.a && this->b == right.b && this->c == right.c;

}

/////////////////////

////friend func/////

///////////////////

istream &operator >> (istream &stream, Triangle &p)

{

cout << "a = "; stream >> p.a;

cout << "b = "; stream >> p.b;

cout << "c = "; stream >> p.c;

return stream;

}

ostream &operator << (ostream &stream, const Triangle &p)

{

stream << "a = " << p.a << "\t"

<< "b = " << p.b << "\t"

<< "c = " << p.c << endl;

return stream;

}

const bool &operator==(const Triangle &left, const Triangle &right)

{

return left.a == right.a && left.b == right.b && left.c == left.c;

}

Результат работы программы:

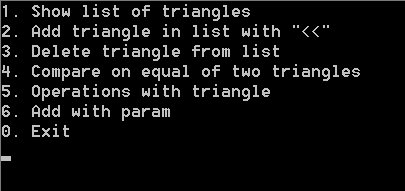


Рисунок 1.

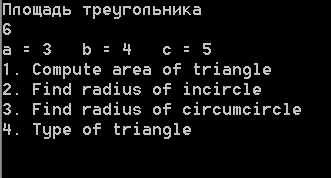


Рисунок 2.