САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Практическая работа №10

«Реализация отношений между классами»

Выполнил: Мороз И.О.

Группа К3120

Проверил: Осипов Н.А.

Санкт-Петербург

2018 г.

# Упражнение 1.

1. Был создан класс IdCard. Была описана логика работы с классом IdCard, а также его поля

#include "IdCard.h"

IdCard::IdCard(int n)

{

number = n;

category = "Не установлена";

}

IdCard::IdCard()

{

number = 0;

category = "Не установлена";

}

IdCard::IdCard(int n, string cat)

{

number = n;

category = cat;

}

void IdCard::setNumber(int newNumber)

{

number = newNumber;

}

int IdCard::getNumber()

{

return number;

}

void IdCard::setCategory(string cat)

{

category = cat;

}

string IdCard::getCategory()

{

return category;

}

1. В классе Student был объявлен указатель iCard и методы управления доступом к данному полю, был изменён конструктор

Student::Student(string name, string last\_name, IdCard \*id)

{

Student::set\_name(name);

Student::set\_last\_name(last\_name);

Student::setIdCard(id);

}

//…

void Student::setIdCard(IdCard \*c)

{

iCard = c;

}

IdCard Student::getIdCard()

{

return \*iCard;

}

1. В методе main был создан объект IdCard. Значения его полей были введены с клавиатуры, а позже выведены в консоль

cout << "Card number: ";

cin >> cardNum;

cout << "Card category: ";

cin >> cardCat;

IdCard \*idc = new IdCard(cardNum, cardCat);

//…

cout << "IdCard: " << student02->getIdCard().getNumber() << endl;

cout << "Category: " << student02->getIdCard().getCategory() << endl;

delete student02;

delete idc;

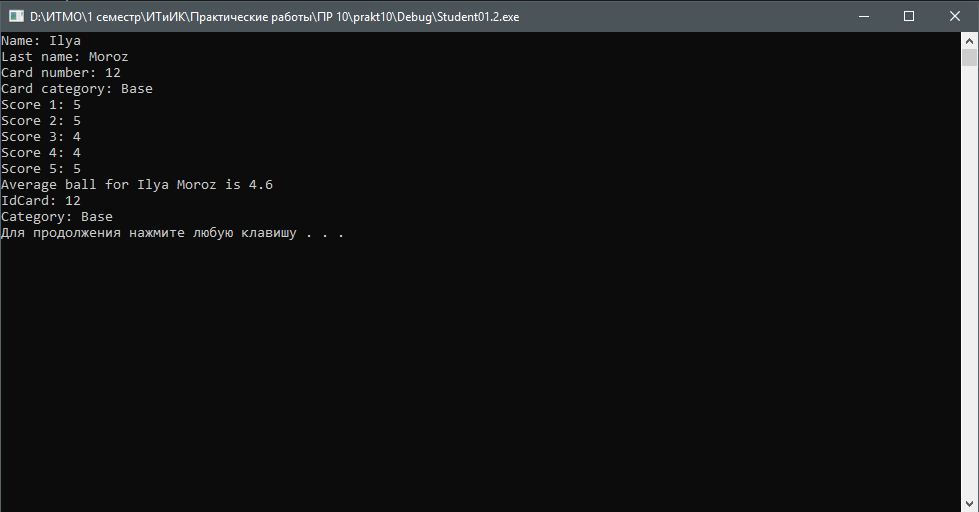


Рисунок 1 – результат работы программы

# Упражнение 2.

1. Был создан класс Dot, определяемый двумя координатами и двумя функциями

#include "Dot.h"

#include <math.h>

Dot::Dot()

{

x = 0; y = 0;

}

Dot::Dot(double x, double y)

{

this->x = x;

this->y = y;

}

double Dot::distanceTo(Dot point)

{

return sqrt(pow((point.x - x), 2) + pow(point.y - y, 2));

}

void Dot::SetCoord(double x, double y)

{

this->x = x;

this->y = y;

}

1. Был создан класс Triangle. В нём были созданы три переменные класса Dot – вершины треугольника. В классе Triangle были определены методы для вывода сторон в консоль, получения периметра и площади треугольника

#include "Triangle.h"

#include "Dot.h"

#include <iostream>

Triangle::Triangle(Dot dot1, Dot dot2, Dot dot3)

{

this->dot1 = dot1;

this->dot2 = dot2;

this->dot3 = dot3;

}

void Triangle::ShowSides()

{

std::cout << "Сторона a: " << dot1.distanceTo(dot2) << "\n";

std::cout << "Сторона b: " << dot2.distanceTo(dot3) << "\n";

std::cout << "Сторона c: " << dot3.distanceTo(dot1) << "\n";

}

double Triangle::GetPerim()

{

return dot1.distanceTo(dot2) + dot2.distanceTo(dot3) + dot3.distanceTo(dot1);

}

double Triangle::GetArea()

{

double p = (dot1.distanceTo(dot2) + dot2.distanceTo(dot3) + dot3.distanceTo(dot1)) / 2;

return sqrt(p\*(p - dot1.distanceTo(dot2))\*(p - dot2.distanceTo(dot3))\*(p - dot3.distanceTo(dot1)));

}

1. В методе main был создан объект класса Triangle. Для него был реализован ввод трёх вершин треугольника. Были использованы методы для вывода сторон треугольника, получения периметра и площади треугольника

Dot dots[3];

cout << "Введите координаты вершин треугольника\n";

for (int i = 0; i < 3; i++)

{

double x, y;

cout << "x" << i << " = ";

cin >> x;

cout << "y" << i << " = ";

cin >> y;

dots[i].SetCoord(x, y);

}

Triangle \*triag1 = new Triangle(dots[0], dots[1], dots[2]);

triag1->ShowSides();

cout << "Периметр треугольника: " << triag1->GetPerim() << "\n";

cout << "Площадь треугольника: " << triag1->GetArea() << "\n";

system("pause");

delete triag1;

return 0;

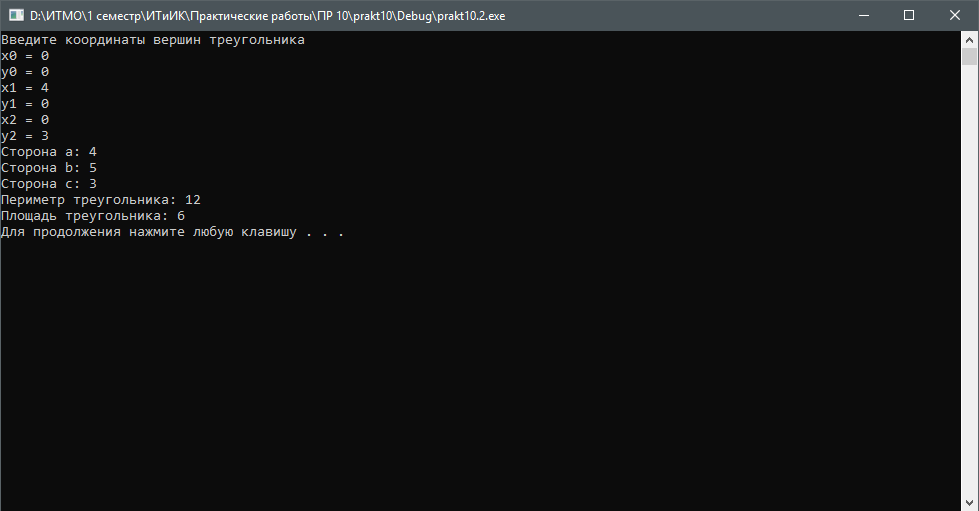


Рисунок 2 – пример работы программы

# Вывод

В ходе выполнения данного практического задания были закреплены и усовершенствованы навыки работы с классами, были рассмотрены отношения между классами. Для закрепления полученных знаний были выполнены упражнения.

# Контрольные вопросы