**Лабораторная работа №2**

**ИССЛЕДОВАНИЕ МЕХАНИЗМА ВИРТУАЛЬНЫХ ФУНКЦИЙ**

1. **Цель работы**

Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизмов наследования и виртуальных функций. Освоение особенностей отладки объектно-ориентированных программ.

1. **Вариант задания**

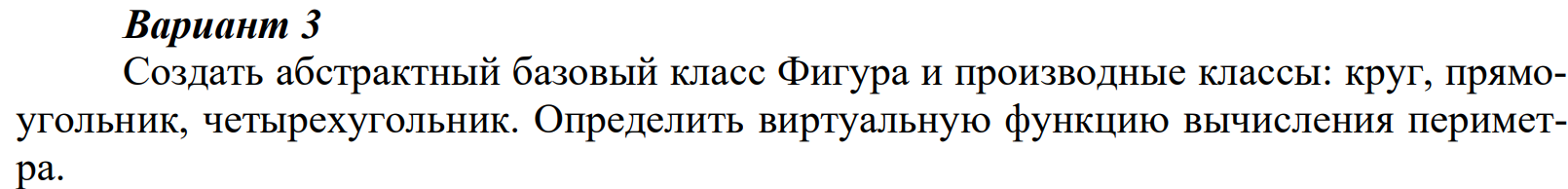
****

Рис 1 – Вариант задания

1. **Текст программы**

#include <iostream>

#define pi 3.1415926535

using namespace std;

class form

{

public:

virtual void perimetr() = 0;

form()

{

cout << "Конструктор"<< endl;

}

virtual ~form()

{

cout << "Деструктор Фигура"<< endl;

}

};

class circle :public form

{

public:

double a=0,r;

circle()

{

cout << "Конструктор круг" << endl;

cout<<"Введите радиус круга"<<endl;

cin >> r;

}

virtual void perimetr()

{

a = 2 \* pi \* r;

cout <<"Периметр круга = "<< a << endl;

}

~circle()

{

cout << "Деструктор Круг" << endl;

}

};

class rectangle :public form

{

public:

double b = 0,c,d;

rectangle()

{

cout << "Конструктор Прямоугольник" << endl;

cout << "Введите 1 сторону: " << endl;

cin >> c;

cout << "Введите 2 соседнюю сторону: " << endl;

cin >> d;

}

virtual void perimetr()

{

b = 2 \* (c + d);

cout << "Периметр прямоугольника = "<< b << endl;

}

~rectangle()

{

cout << "Деструктор Прямоугольник" << endl;

}

};

class quadrilateral:public form

{

public:

double e=0,f,g,h,i;

quadrilateral()

{

cout << "Конструктор Четырёхугольник" << endl;

cout << "Введите длину 1 ребра: " << endl;

cin >> f;

cout << "Введите длину 2 ребра: " << endl;

cin >> g;

cout << "Введите длину 3 ребра: " << endl;

cin >> h;

cout << "Введите длину 4 ребра: " << endl;

cin >> i;

}

virtual void perimetr()

{

e = f + g + h + i;

cout << "Периметр Четырёхугольника = "<< e << endl;

}

~quadrilateral()

{

cout << "Деструктор Четырёхугольник" << endl;

}

};

int main()

{

setlocale(LC\_ALL, "Russian");

form\* ob;---

ob = new circle;

ob->perimetr();

delete ob;

ob = new rectangle;

ob->perimetr();

delete ob;

ob = new quadrilateral;

ob->perimetr();

delete ob;

system("pause");

}

1. **Тестовые примеры**

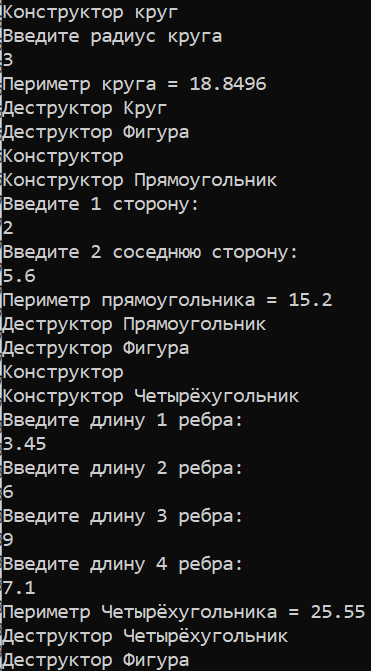
****

Рис 2 – Демонстрация работы программы

**Вывод**

Были приобретены практические навыки при написании объектно-ориентированных программ с использованием механизмов наследования и виртуальных функций. Освоены особенности отладки объектно-ориентированных программ.