**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 3**

Тема: Механизмы наследования в С++

Студент: Воронов Кирилл Михайлович

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. **Постановка задачи**

Изучение механизмов работы с наследованием в С++

Разработать классы согласно варианту задания, классы должны наследоваться от базового класса Figure. Фигуры являются фигурами вращения. Все классы должны поддерживать набор общих методов:

1. Вычисление геометрического центра фигуры;

2. Вывод в стандартный поток вывода std::cout координат вершин фигуры;

3. Вычисление площади фигуры.

*Вариант 27*

Прямоугольник Ромб Трапеция

1. **Описание программы**

Общий класс Figure имеет три виртуальные функции: square - площадь фигуры, center - центр фигуры и print - печать координат фигуры. Создадим три подкласса для каждой из фигуры.

1. Класс Pramougol

Этот класс предназначен для прямоугольника. Будем хранить левую нижнюю координату t (отдельно создадим класс координат с переменными х,у) и длины двух сторон а, b. Реализуем функции нахождения площади, центра и печати координат.

2.Класс Romb

Этот класс предназначен для ромба. Будем хранить нижнюю координату и длины двух диагоналей. Также реализуем функции из общего класса.

3. Класс Trapecia

Этот класс предназначен для трапеции. Будем хранить её нижнюю левую координату, длины оснований и боковой стороны. Определим выше описанные функции.

1. **Набор тестов и результаты их выполнения**

**Тест 1**

Введите координату нижней левой точки прямоугольника и длины сторон, прилегающих к ней, начиная с горизонтальной

-9 -5 a -6

Введены неверные данные

**Тест 2**

Введите координату нижней левой точки прямоугольника и длины сторон, прилегающих к ней, начиная с горизонтальной

1 1 6 5

Площадь - 30

Центр - (4,3.5)

Координаты

(1,6) (7,6)

(1,1) (7,1)

Введите координату нижней точки ромба и длину его диагоналей, начиная с горизонтальной

2 1 5 6

Площадь - 15

Центр - (2,4)

Координаты

(2,7)

(-0.5,4) (4.5,4)

(2,1)

Введите координату нижней левой точки трапеции, наибольшее основание, наименьшее основание и длину боковой стороны

8 7 5 3 6

Площадь - 23.6643

Центр - (10.5,9.95804)

Координаты

(9,12.9161) (12,12.9161)

(8,7) (13,7)

Общая площадь - 68.6643

**Тест 3**

Введите координату нижней левой точки прямоугольника и длины сторон, прилегающих к ней, начиная с горизонтальной

-5 -6 12 7

Площадь - 84

Центр - (1,-2.5)

Координаты

(-5,1) (7,1)

(-5,-6) (7,-6)

Введите координату нижней точки ромба и длину его диагоналей, начиная с горизонтальной

-9 6.45 21.4 54.1

Площадь - 578.87

Центр - (-9,33.5)

Координаты

(-9,60.55)

(-19.7,33.5) (1.7,33.5)

(-9,6.45)

Введите координату нижней левой точки трапеции, наибольшее основание, наименьшее основание и длину боковой стороны

-7.41 14.4 59.789 12.5 68.8925

Площадь - 2338.84

Центр - (22.4845,46.754)

Координаты

(16.2345,79.1079) (28.7345,79.1079)

(-7.41,14.4) (52.379,14.4)

Общая площадь - 3001.71

**Тест 4**

Введите координату нижней левой точки прямоугольника и длины сторон, прилегающих к ней, начиная с горизонтальной

-987.4545 698.451 171.54 225.569

Площадь - 38694.1

Центр - (-901.685,811.236)

Координаты

(-987.455,924.02) (-815.915,924.02)

(-987.455,698.451) (-815.915,698.451)

Введите координату нижней точки ромба и длину его диагоналей, начиная с горизонтальной

789.456 -3.5 698.2 21.1

Площадь - 7366.01

Центр - (789.456,7.05)

Координаты

(789.456,17.6)

(440.356,7.05) (1138.56,7.05)

(789.456,-3.5)

Введите координату нижней левой точки трапеции, наибольшее основание, наименьшее основание и длину боковой стороны

-98.5 -963.545 1000.455 32 9999.999

Площадь - 5.15622e+06

Центр - (401.728,4030.59)

Координаты

(385.728,9024.72) (417.728,9024.72)

(-98.5,-963.545) (901.955,-963.545)

Общая площадь - 5.20228e+06

1. **Листинг программы**

/\*Воронов К.М.

Разработать классы согласно варианту задания, классы должны наследоваться

от базового класса Figure. Фигуры являются фигурами вращения.

Все классы должны поддерживать набор общих методов:

1.Вычисление геометрического центра фигуры;

2.Вывод в стандартный поток вывода std::cout координат вершин фигуры;

3.Вычисление площади фигуры.

Вариант 27

Прямоугольник ромб трапеция

\*/

#include "iostream"

#include <vector>

#include <valarray>

using namespace std;

class Coord{

public:

double x,y;

Coord(): x(0),y(0){};

Coord(double a, double b): x(a),y(b){};

friend ostream & operator<<(ostream &out,const Coord &a);

~Coord(){};

};

ostream & operator<<(ostream &out,const Coord &a){

out<<"("<<a.x<<","<<a.y<<")";

return out;

}

class Figure{

public:

virtual double square() = 0;

virtual Coord centre() = 0;

virtual void print()=0;

};

class Pramougol: public Figure{

public:

Coord t;

double a,b;

Pramougol():a(0),b(0),t(){};

Pramougol(double x,double y,double a1, double b1): a(a1), b(b1), t(x,y){};

double square() override{

return a\*b;

}

Coord centre() override {

Coord t2(t.x+a,t.y);

Coord t3(t.x+a,t.y+b);

Coord t4(t.x,t.y+b);

t2.x=(t.x+t2.x+t3.x+t4.x)/4;

t2.y=(t.y+t2.y+t3.y+t4.y)/4;

return (t2);

}

void print() override {

Coord t2(t.x+a,t.y);

Coord t3(t.x+a,t.y+b);

Coord t4(t.x,t.y+b);

cout << t4 << " "<< t3 << endl<<endl<<t<<" "<<t2<<endl;

}

};

class Romb: public Figure{

public:

Coord t;

double a,b;

Romb():t(),a(0),b(0){};

Romb(double x,double y,double a1, double b1): a(a1),b(b1),t(x,y){};

double square() override{

return a\*b/2;

}

Coord centre() override {

Coord t2(t.x,t.y+b);

Coord t3(t.x+a/2,t.y+b/2);

Coord t4(t.x-a/2,t.y+b/2);

t2.x=(t.x+t2.x+t3.x+t4.x)/4;

t2.y=(t.y+t2.y+t3.y+t4.y)/4;

return t2;

}

void print() override {

Coord t2(t.x,t.y+b);

Coord t3(t.x+a/2,t.y+b/2);

Coord t4(t.x-a/2,t.y+b/2);

cout<<" "<<t2<<endl<<endl<<t4<<" "<<t3<<endl<<endl <<" "<<t<<endl;

}

};

class Trapecia: public Figure{

public:

Coord t;

double a,b,c;

Trapecia():a(0),b(0),c(0),t(){};

Trapecia(double x,double y,double a1, double b1,double c1): a(a1), b(b1),c(c1), t(x,y){};

double square() override{

double g = (a - b)/2;

g=sqrt(c\*c-g\*g);

return((a+b)\*g/2);

}

Coord centre() override {

double g1 = (a - b)/2;

double g2=sqrt(c\*c-g1\*g1);

Coord t2(t.x+a,t.y);

Coord t3(t.x+g1,t.y+g2);

Coord t4(t2.x-g1,t.y+g2);

t2.x=(t.x+t2.x+t3.x+t4.x)/4;

t2.y=(t.y+t2.y+t3.y+t4.y)/4;

return t2;

}

void print() override {

double g1 = (a - b)/2;

double g2=sqrt(c\*c-g1\*g1);

Coord t2(t.x+a,t.y);

Coord t3(t.x+g1,t.y+g2);

Coord t4(t2.x-g1,t.y+g2);

cout<<" "<<t3<<" "<<t4<<endl<<endl<<t<<" "<<t2<<endl;

}

};

int main(){

vector<Figure\*> f(3);

double x,y,a,b,c,pl;

Coord cen;

printf("Введите координату нижней левой точки прямоугольника и длины сторон, прилегающих к ней, начиная с горизонтальной\n");

cin>>x>>y>>a>>b;

if((!cin)||(a<0)||(b<0)){

printf("Введены неверные данные\n");

return 1;

}

f[0] = new Pramougol(x,y,a,b);

pl=f[0]->square();

cout<<"Площадь - "<<pl<<endl;

cen=f[0]->centre();

cout<<"Центр - "<<cen<<endl;

cout<<"Координаты"<<endl;

f[0]->print();

printf("Введите координату нижней точки ромба и длину его диагоналей, начиная с горизонтальной\n");

cin>>x>>y>>a>>b;

if((!cin)||(a<0)||(b<0)){

printf("Введены неверные данные\n");

return 1;

}

f[1] = new Romb(x,y,a,b);

pl=f[1]->square();

cout<<"Площадь - "<<pl<<endl;

cen=f[1]->centre();

cout<<"Центр - "<<cen<<endl;

cout<<"Координаты"<<endl;

f[1]->print();

printf("Введите координату нижней левой точки трапеции, наибольшее основание, наименьшее основание и длину боковой стороны\n");

cin>>x>>y>>a>>b>>c;

if(!cin){

printf("Введены неверные данные\n");

return 1;

}

double g1 = (a - b)/2;

double g2=sqrt(c\*c-g1\*g1);

if((a<0)||(b<0)||(c<0)||(g1+c<g2)||(g2+g1<c)||(g2+c<g1)||(g1<=0)||(g2<=0)){

printf("Введены неверные данные\n");

return 1;

}

f[2] = new Trapecia(x,y,a,b,c);

pl=f[2]->square();

cout<<"Площадь - "<<pl<<endl;

cen=f[2]->centre();

cout<<"Центр - "<<cen<<endl;

cout<<"Координаты"<<endl;

f[2]->print();

cout<<"Общая площадь - "<<f[0]->square()+f[1]->square()+f[2]->square()<<endl;

}

1. **Вывод**

Я изучил механизмы наследования в С++ и вспомнил школьную геометрию.

**Список литературы**

# Наследование в C++[Электронный ресурс]

URL:https://habr.com/ru/post/445948/(дата обращения 16.10.2020).

1. Виртуальный метод[Электронный ресурс]

URL:<https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4> (дата обращения 16.10.2020).