**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №5

Дисциплина: Объектно-ориентированное программирование

по теме Классы, виды отношений. Наследование.

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверил: Буханов Д.Г.

**Белгород 2019**

Вариант 1

Задание 1

Графический редактор.

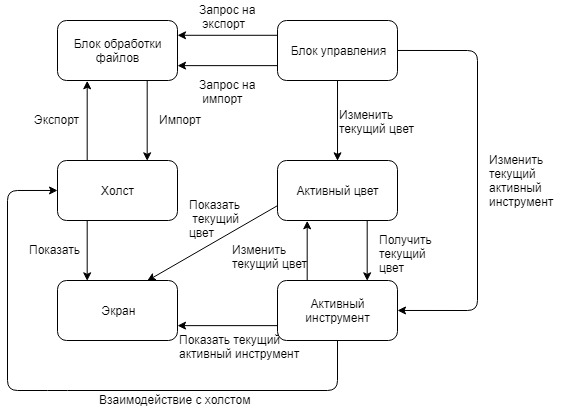
Задание 2

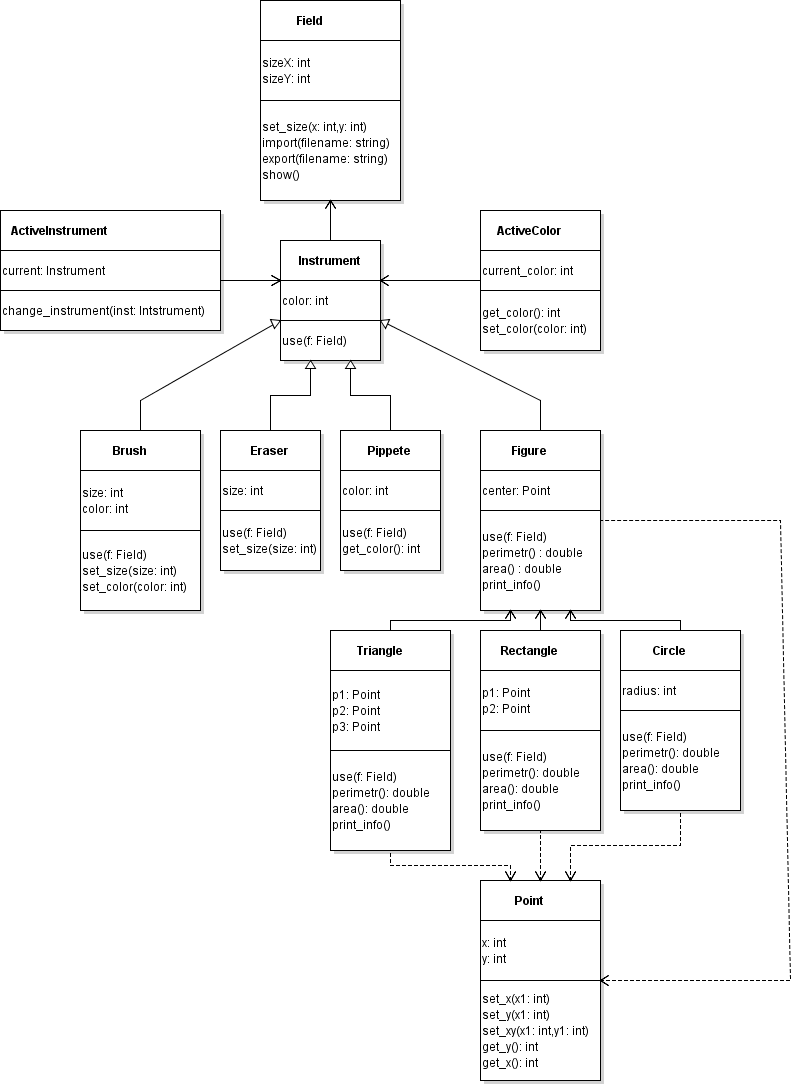
1. Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран.

2. Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник) со своими методами вычисления площади и периметра.

3. Создать массив n фигур и вывести полную информацию о фигурах на экран.

Задание 1





Задание 2.

#include <iostream>

#include <math.h>

using namespace std;

class Point{

int x,y;

public:

Point(int x1,int y1){

this->x=x1;

this->y=y1;

}

void set\_x(int x1){this->x=x1;};

void set\_y(int y1){this->y=y1;};

void set\_xy(int x1,int y1){

this->set\_x(x1);

this->set\_y(y1);

};

int get\_x(){

return this->x;

};

int get\_y(){

return this->y;

};

};

class Figure{

Point center;

public:

Figure(Point t):center(t){};

virtual double perimetr()=0;

virtual double area()=0;

virtual void print\_info()=0;

virtual ~Figure(){};

};

class Rectangle : public Figure {

Point p1,p2; // диагональные точки.

public:

Rectangle(Point a,Point b,Point center):Figure(center),p1(a),p2(b){};

double perimetr();

double area();

void print\_info();

};

class Triangle : public Figure {

Point p1,p2,p3; //координаты вершин треугольника

public:

Triangle(Point a,Point b, Point c,Point center):Figure(center),p1(a),p2(b),p3(c){};

double perimetr();

double area();

void print\_info();

};

class Circle : public Figure {

int radius;

public:

Circle(int r,Point center):Figure(center),radius(r){};

double perimetr();

double area();

void print\_info();

};

int point\_distance(Point a,Point b){

int x=b.get\_x()-a.get\_x();

int y=b.get\_y()-a.get\_y();

return (sqrt(x\*x+y\*y));

};

//реализация методов класса Rectangle

double Rectangle::perimetr() {

Point temp(this->p1.get\_x(),this->p2.get\_y());

return 2\*(point\_distance(this->p1,temp)+point\_distance(temp,this->p2));

};

double Rectangle::area() {

Point temp(this->p1.get\_x(),this->p2.get\_y());

return point\_distance(this->p1,temp)\*point\_distance(temp,this->p2);

};

void Rectangle::print\_info() {

std::cout <<"Информация о прямоугольнике: "<<std::endl;

std::cout << "Периметр = " << this->perimetr() <<std::endl;

std::cout << "Площадь = " << this->area()<< std::endl<< std::endl;

};

//реализация методов класса Triangle

double Triangle::perimetr() {

return point\_distance(p1,p2)+point\_distance(p2,p3)+point\_distance(p1,p3);

};

double Triangle::area() {

int a=point\_distance(p1,p2);

int b=point\_distance(p2,p3);

int c=point\_distance(p1,p3);

int p=0.5\*(a+b+c);

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

};

void Triangle::print\_info() {

std::cout <<"Информация о треугольнике: "<<std::endl;

std::cout << "Периметр = " << this->perimetr() <<std::endl;

std::cout << "Площадь = " << this->area()<< std::endl<< std::endl;

};

//реализация методов класса Circle

double Circle::perimetr() {

return 2\*radius\*M\_PI;

};

double Circle::area() {

return M\_PI\*radius\*radius;

};

void Circle::print\_info() {

std::cout <<"Информация о круге: "<<std::endl;

std::cout << "Радиус = " << this->radius <<std::endl;

std::cout << "Периметр = " << this->perimetr() <<std::endl;

std::cout << "Площадь = " << this->area()<< std::endl<< std::endl;

};

int main() {

char temp\_type;

int x1,x2,x3,y1,y2,y3,y4,x4,n;

cout <<"Введите кол-во фигур :";

cin >> n;

Figure\* arr[n];

int i = 0;

while(i<n){

cout <<"Выберите тип вводимой фигуры (1-прямоугольник, 2-круг, 3-треугольник): ";

cin >> temp\_type;

switch (temp\_type) {

case '1':

cout << "Введите координаты фигуры центра: ";

cin >> x1 >> y1;

cout <<"Введите координаты точки p1: ";

cin >> x2 >> y2;

cout <<"Введите координаты точки p2: ";

cin >> x3 >> y3;

arr[i]=new Rectangle(Point(y2,y2),Point(x3,y3),Point(x1,y1));

i++;

break;

case '2':

cout << "Введите координаты фигуры центра: ";

cin >> x1 >> y1;

cout <<"Введите радиус: ";

cin >> x2;

arr[i]=new Circle(x2,Point(x1,y1));

i++;

break;

case '3':

cout << "Введите координаты фигуры центра: ";

cin >> x1 >> y1;

cout <<"Введите координаты точки p1: ";

cin >> x2 >> y2;

cout <<"Введите координаты точки p2: ";

cin >> x3 >> y3;

cout <<"Введите координаты точки p3: ";

cin >> x4 >> y4;

arr[i]=new Triangle(Point(x2,y2),Point(x3,y3),Point(x4,y4),Point(x1,y1));

i++;

break;

default:

cout <<"\nНеправильный тип фигуры."<<endl;

break;

}

}

for(i=0;i<n;i++) {

arr[i]->print\_info();

delete arr[i];

}

return 0;

}