ГУАП

КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель E.О. Шумова

должность, уч. степень, звание подпись, дата инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

«НАСЛЕДОВАНИЕ КЛАССОВ, БАЗОВЫЙ КЛАСС, ПРОИЗВОДНЫЙ КЛАСС»

по курсу: ОБЪЕКТНО ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

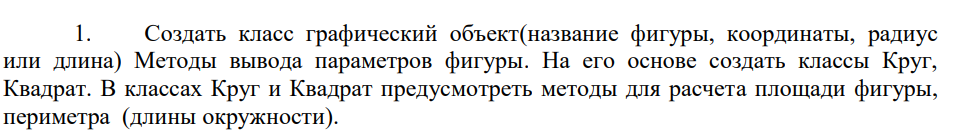
СТУДЕНТ ГР. № 4134К Иванов И.В.

подпись, дата инициалы, фамилия

Санкт-Петербург 2022

**Цель работы:** Изучить механизм создания нового класса на основе уже существующего, варианты доступа к элементам базового класса из производного.

**Условие для варианта:** 1



**Листинг программы:**

#include <iostream>

#include <string>

#include <math.h>

#define \_USE\_MATH\_DEFINES

using namespace std;

class Figure {

protected:

string name;

int start\_x, start\_y;

double radius;

public:

Figure(string name, int x, int y) {

this->name = name;

start\_x = x;

start\_y = y;

}

string getName();

int getXchrod();

int getYchord();

void show();

};

string Figure::getName() { return name; }

int Figure::getXchrod() { return start\_x; }

int Figure::getYchord() { return start\_y; }

void Figure::show() {

cout << "Figure name: " << name << "\n";

cout << "start X: " << start\_x << "\n";

cout << "start Y: " << start\_y << "\n";

}

class Circle :public Figure {

double radius1;

public:

void rad(double radius1) {

this ->radius=radius1;

}

Circle(string name, int x, int y, double r)

: Figure(name, x, y), radius1(r) {

};

double getArea();

double getLength();

void show();

};

double Circle::getArea() {

double pi = 22.0 / 7.0;

return pi \* (radius1 \* radius1);

}

double Circle::getLength() {

double pi = 22.0 / 7.0;

return 2 \* pi \* radius1;

}

void Circle::show() {

Figure::show();

cout << "Радиус: " << radius1 << endl;

}

class Square :public Figure {

double length1;

public:

void rad(double length11) {

this->radius = length1;

}

Square(string name, int x, int y, double len)

: Figure(name, x, y), length1(len) {

};

double getArea();

double getPerimeter();

void show();

};

double Square::getArea() {

return length1 \* length1;

}

double Square::getPerimeter() {

return 4 \* length1;

}

void Square::show() {

Figure::show();

cout << "Длина стороны: " << length1 << endl;

}

int main() {

setlocale(LC\_ALL, "rus");

system("color F0");

Circle c("Круг", 0, 0, 4.25);

Square sq("Квадрат", 0, 0, 7);

c.show();

cout << "Площадь круга: " << c.getArea() << endl;

cout << "Длина окружности: " << c.getLength() << "\n\n";

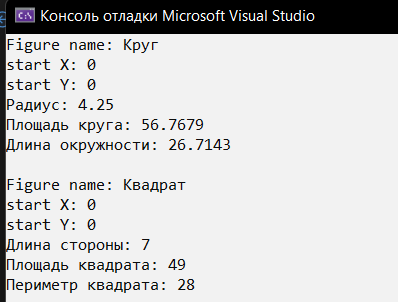
sq.show();

cout << "Площадь квадрата: " << sq.getArea() << endl;

cout << "Периметр квадрата: " << sq.getPerimeter() << endl;

}

**Результаты работы программы:**



**Вывод:**

В ходе лабораторной работы я изучил механизм создания нового класса на основе уже существующего, а так же варианты доступа к элементам базового класса из производного.