

Министерство науки и высшего образования РФ
ФГАОУ ВО «Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина»

ИРИТ-РТФ

Центр ускоренного обучения

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

по дисциплине «Прикладное программирование»

Тема: *Знакомство с механизмом наследования C++, виртуальными функциями, абстрактными классами*

Студент группы РИЗ-200028у:

И. С. Арсентьев

Преподаватель:

О. Л. Чагаева,

ст. преподаватель

Екатеринбург 2022

СОДЕРЖАНИЕ

1 Постановка задачи.....	3
2 Описание работы.....	4
2.1 . Выполнение поставленных задач.	4
3 Выводы по лабораторной работе	10

1 Постановка задачи

Цель: познакомиться с механизмом наследования C++, виртуальными функциями, абстрактными классами

2 Описание работы

2.1 . Выполнение поставленных задач.

1) Основной класс Lab7_1.cpp

```
#include <iostream>
//RU
#include <locale.h>
#include <windows.h>
//
#include "graphics.h"
#include <stdio.h>

class Point // базовый класс
{
public:
    //переменные класса - координаты и цвет
    int x, y;
    WORD cvet;
    //конструктор с параметрами
    Point(int xn, int yn, WORD color) : x(xn), y(yn),
    cvet(color)
    {
        x = xn; y = yn; cvet = color;
    }
    // виртуальные функции
    virtual void Show();
    virtual void Hide();
    // методы
    void Locat(int&, int&);
    void Fly(int cost);
};

class Krug : public Point // производный класс от Point
{
public:
    // параметр класса - радиус
    WORD radius;
    // наследование конструктора с параметром
    Krug(int xn, int yn, WORD color, WORD r) : Point(xn, yn,
    color), radius(r)
    {
        Point(xn, yn, color);
        radius = r;
        Show();
    }
    // виртуальные функции
    virtual void Show();
    virtual void Hide();
};

class Ring : public Krug // производный класс от Krug
{
public:
    // параметр класса - ширина
    WORD width;
    // наследование конструктора с параметром
    Ring(int xn, int yn, WORD color, WORD r, WORD wid) :
    Krug(xn, yn, color, r), width(wid)
```

```

        {
            width = wid;
            Krug(xn, yn, color, r);
        }
        //виртуальные функции
        virtual void Show();
};

int main()
{
    //для работы с русским языком
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    //
    std::cout << "Объекты перемещаются по экрану при нажатии
любой клавиши кроме ESC \n";
    std::cout << "Координаты объектов задаются генератором
случайных чисел \n";
    std::cout << "При нажатии ESC программа выполняется
дальше... \n";
    // инициализация графического окна
    initwindow(500, 500);
    std::cout << "Графическое окно активно! Сделайте его
активным для работы... \n";
    setbkcolor(0); // установка цвета фона
    // создаем объекты с начальными параметрами
    Krug TestKrug(150, 40, 15, 10);
    Ring TestRing(450, 80, 90, 50, 10);
    // перемещение объектов с выводом их координат
    std::cout << "--> Перемещение 1-го объекта! \n";
    TestKrug.Fly(100);
    std::cout << "--> 1-й объект стоп! \n";
    std::cout << "--> Перемещение 2-го объекта! \n";
    TestRing.Fly(60);
    std::cout << "--> 2-й объект стоп! \n";
    std::cout << "--> Перемещение 1-го объекта! \n";
    TestKrug.Fly(60);
    std::cout << "--> 2-й объект стоп! \n";
    getch();
    std::cout << "--> Скрыть объекты! \n";
    // прячем объекты в конце
    TestKrug.Hide();
    TestRing.Hide();
    getch();
    // закрываем графическое окно
    closegraph();
    std::cout << "Графическое окно закрыто! \n";
    system("pause");
}

void Point::Show()
{
    // виртуальные функции
}

void Point::Hide()
{
    // виртуальные функции
}

```

```

void Point::Locat(int& x1, int& y1)
{    // запрос координат внутри класса
    x1 = x; y1 = y;
}
void Point::Fly(int cost)
{
    int xx, yy;
    Show();
    //генератор случайных чисел
    srand(time(0)); // автоматическая рандомизация r=1 +
rand() % 10; где 1 - начало, 10 - ширина диапазона
    int in;
    do
    {
        Locat(xx, yy); // запрос текущих координат, в 1-й раз
из конструктора
        do //перемещение 1
        {
            xx = xx + round(((rand() % 2) - 0.5) * cost); //
приращение координат
        } while ((xx < 0) || (xx > getmaxx())); //
отслеживание координат < области окна
        do //перемещение 2
        {
            yy = yy + round(((rand() % 2) - 0.5) * cost);
// приращение координат
        } while ((yy < 0) || (yy > getmaxy())); //
отслеживание координат < области окна
        // создание эффекта моргания/перемещения
        Hide();
        x = xx; y = yy; // обновляем координаты класса
        Show();
        std::cout << "Координаты объекта: x=" << xx << " y =
" << yy << "\n";
        delay(300);
        in = getch();
    } while (in!=27);
}
void Krug::Show()
{
    setcolor(cvet); // установка цвета
    setfillstyle(1, cvet); // цвет заливки и текстура
    pieslice(x, y, 0, 320, radius); // сектор круговой
диаграммы на 0-320
}
void Krug::Hide()
{
    setcolor(getbkcolor()); // установка цвета - цвет фона
    setfillstyle(1, getbkcolor()); // цвет заливки и текстура
    pieslice(x, y, 0, 320, radius); // сектор круговой
диаграммы на 0-320
}
void Ring::Show()
{
    Krug::Show(); //вызываем функцию прородителя класса
    setfillstyle(10, getbkcolor()); // цвет заливки и
текстура
    pieslice(x, y, 0, 320, radius-width); // сектор круговой
диаграммы на 0-320
}

```

2.Класс с дополненным методом

```
#include "Circle.h"
void Circle::Show()
{   framework->draw_circle(x,y,color,radius, angle);
}/// <summary>
/// Устанавливает рандомный размер "рта" круга
/// </summary>
void Circle::ChangeEndAngle()
{
    angle = rand() % 120 + 240;
}
void Circle::Hide()
{
    framework->draw_circle(x, y, framework->get_bk(), radius,
360);
}

#include "Point.h"
#include <cstdlib>
#include <windows.h>
#include <winuser.h>

const unsigned short MSB = 0x8000;
bool listenKeyPress(short p_key);

void Point::Locate(int* x1, int* y1)
{
    *x1 = x;
    *y1 = y;
}
/// <summary>
/// Устанавливает рандомный цвет из 15 доступных, кроме чёрного
/// </summary>
void Point::ChangeColor()
{
    color = rand() % 15 + 1;
}
int Point::GetMaxX()
{
```

```

        return framework->get_x();
    }
    int Point::GetMaxY()
    {
        return framework->get_y();
    }
    void Point::Fly(int cost)
    {
        int xx, yy;
        Locate(&xx, &yy);
        int state = 0;
        Show();
        do
        {
            /*do
            {
                xx = xx + round((rand() % 10 - 5) * cost / 10);
            } while (!(xx > 0 && xx < GetMaxX()));
            do
            {
                yy = yy + round((rand() % 10 - 5) * cost / 10);
            } while (!(yy > 0 && yy < GetMaxY()));*/
            int n_xx = xx + round((rand() % 10 - 5) * cost / 10);
            int n_yy = yy + round((rand() % 10 - 5) * cost / 10);
            if (n_xx > 0 && n_xx < GetMaxX() && n_yy > 0 && n_yy <
GetMaxY())
            {
                xx = n_xx;
                yy = n_yy;
                Hide();
                x = xx;
                y = yy;
                ChangeColor();
                ChangeEndAngle();
                Show();
                Sleep(300);
            }
        } while (!listenKeyPress(VK_RETURN));
    }
}

```



```

bool listenKeyPress(short p_key)
{
    return (GetAsyncKeyState(p_key) & MSB);
}

#include "Ring.h"
void Ring::Show()
{
    framework->draw_ring(x,y,color,radius,width, angle,
innerAngle);
}

void Ring::ChangeEndAngle()
{
    angle = rand() % 120 + 240;
    innerAngle = rand() % 220 + 140;
}
#include <iostream>
#include "Ring.h"
int main()
{
    Framework* framework = new Framework(700, 700, 100);
    Circle* test_circle = new Circle(framework, 150, 40, 50, 1);
    Circle* test_ring = new Ring(framework, 450, 80, 90, 50, 1);

    //framework->set_bk(80);

    test_circle->Fly(100);
    test_ring->Fly(60);
    test_circle->Fly(60);
    std::cin.get();
    test_ring->Hide();
    test_circle->Hide();
    std::cin.get();
    delete framework;
}

```

3 Выводы по лабораторной работе

В рамках выполнения лабораторной работы был дополнен набор методов одного из классов функциями, которые выполняют сужение-расширение вырезанного сектора на фигурах и смену цветов фигур одновременно с перемещением.