МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №16

по дисциплине Основы программирования:

«Классы»

Работу выполнил

студент гр. 4141 В.С. Сыворотнев

Санкт-Петербург

2022

***Цель лабораторной работы:*** *изучение структуры, свойств и видов объектов; изучение способов доступа к полям и правил вызова методов объектов; получение навыков объектно-ориентированного программирования на языке C/C++.*

***Задание на программирование:*** *используя технологию объектно-ориентированного программирования разработать программу, реализующую движущийся графический объект в соответствии с индивидуальным заданием с размещением объекта в динамической памяти.*

***Вариант 31:***

*Движение закрашенного эллипса по треугольному контуру с изменением цвета при изменении направления движения.*

**Код программы**

#include <windows.h>

#include <iostream>

#define SLEEP\_DURATION 16

POINT operator-(const POINT& a, const POINT& b)

{

    return { a.x - b.x, a.y - b.y };

}

POINT operator+(const POINT& a, const POINT& b)

{

    return { a.x + b.x, a.y + b.y };

}

POINT operator\*(const POINT& a, const double b)

{

    return { LONG((double)a.x \* b), (LONG)((double)a.y \* b) };

}

class DrawingEllipse

{

private:

    RECT size;

    POINT center;

    POINT(\*moveFunc)(time\_t);

public:

    DrawingEllipse(RECT size, POINT(\*moveFunc)(time\_t))

        : size(size)

        , moveFunc(moveFunc)

    {

        center = moveFunc(0);

    };

    void tick(time\_t time)

    {

        center = moveFunc(time);

    }

    RECT getFrameBounds() const

    {

        return {

            size.left + center.x,

            size.top + center.y,

            size.right + center.x,

            size.bottom + center.y

        };

    }

};

POINT moveFunc(time\_t time)

{

    static const POINT trianglePoints[] =

    {

        {0, 0},

        {600, 0},

        {0, 600}

    };

    static const POINT deltaPoints[] =

    {

        trianglePoints[1] - trianglePoints[0],

        trianglePoints[2] - trianglePoints[1],

        trianglePoints[0] - trianglePoints[2]

    };

    time\_t period = time % 1000;

    if (period < 300)

        return trianglePoints[0] + deltaPoints[0] \* (period / 300.);

    else if (period < 600)

        return trianglePoints[1] + deltaPoints[1] \* ((period - 300) / 300.);

    else

        return trianglePoints[2] + deltaPoints[2] \* ((period - 600) / 300.);

}

COLORREF colorFunc(time\_t time)

{

    // three colors loop

    time\_t period = time % 1000;

    if (period < 300)

        return RGB(0xFF, 0x00, 0x00);

    else if (period < 600)

        return RGB(0x00, 0xFF, 0x00);

    else

        return RGB(0x00, 0x00, 0xFF);

}

int main()

{

    setlocale(LC\_ALL, "Russian");

    size\_t width = 50, height = 50;

    std::cout << "Введите ширину и высоту эллипса: ";

    do {

        std::cin >> width >> height;

    } while (std::cin.fail());

    std::cin.ignore();

    std::cin.clear();

    auto figure = new DrawingEllipse{ {0, 0, (LONG)width, (LONG)height}, moveFunc };

    time\_t time = 0;

    HWND hConsole = GetConsoleWindow();

    HDC hdc = GetDC(hConsole);

    HBRUSH hbrBlack = (HBRUSH)GetStockObject(BLACK\_BRUSH);

    HPEN hPen = CreatePen(PS\_SOLID, 7, RGB(0xFF, 0xFF, 0xFF));

    SelectObject(hdc, hPen);

    for (;;)

    {

        RECT rcConsole;

        GetClientRect(hConsole, &rcConsole);

        FillRect(hdc, &rcConsole, hbrBlack);

        HBRUSH hbrFig = CreateSolidBrush(colorFunc(time));

        SelectObject(hdc, hbrFig);

        figure->tick(time);

        auto [left, top, right, bottom] = figure->getFrameBounds();

        Ellipse(hdc, left, top, right, bottom);

        time += SLEEP\_DURATION;

        Sleep(SLEEP\_DURATION);

        DeleteObject(hbrFig);

    }

    delete figure;

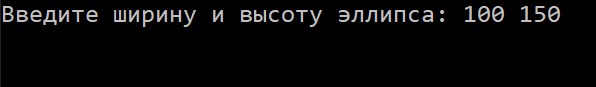
    DeleteObject(hPen);

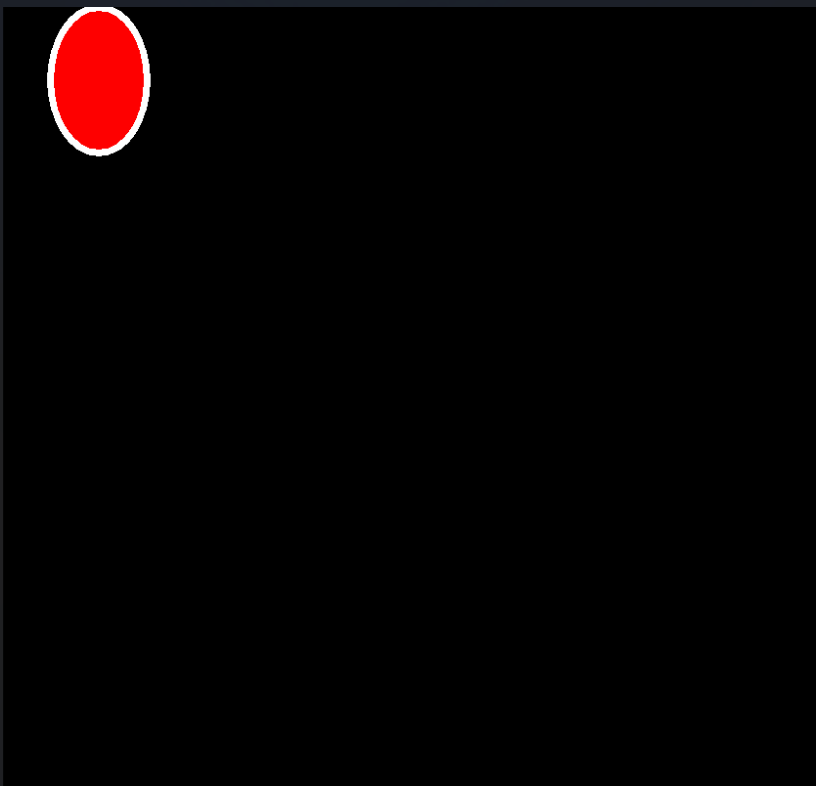
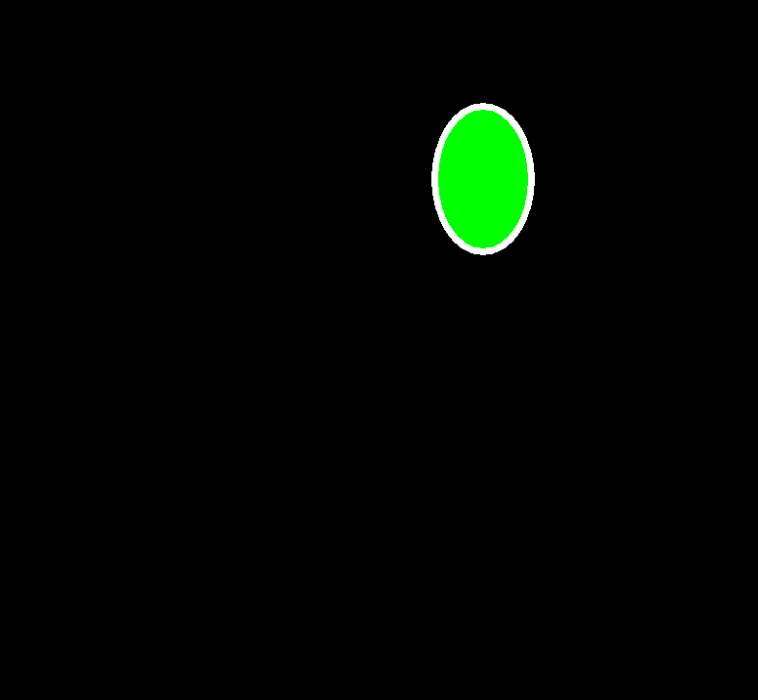
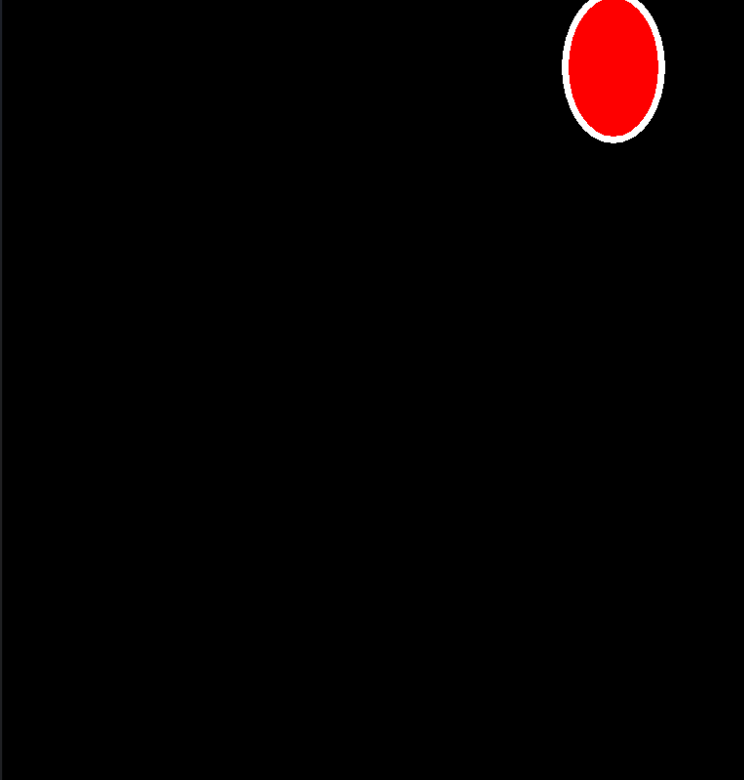
    ReleaseDC(hConsole, hdc);

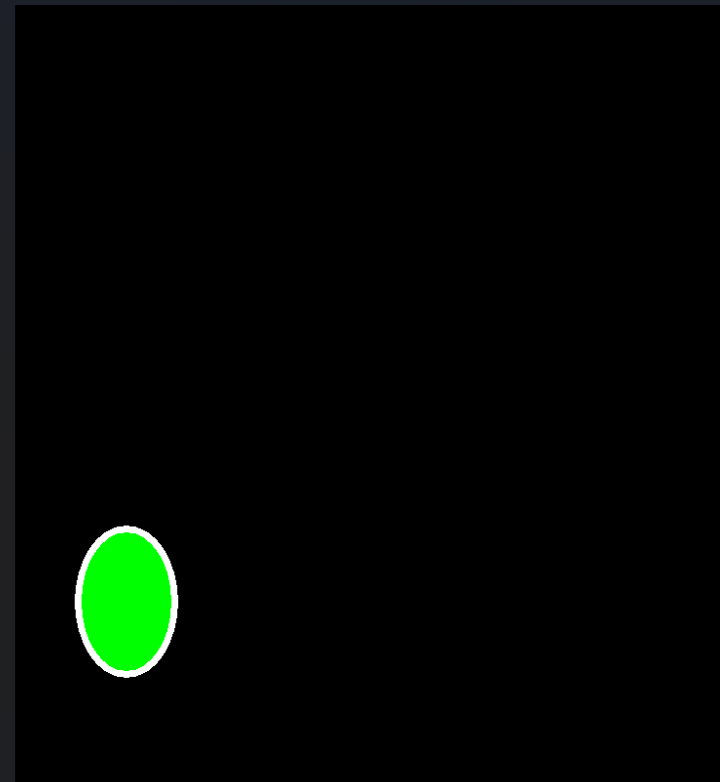
    return 0;

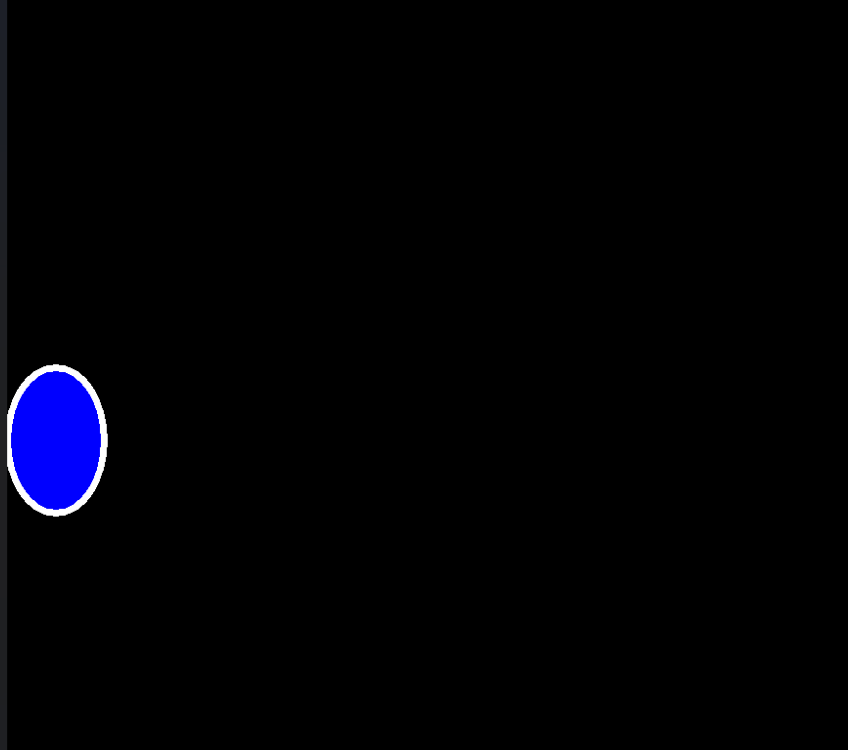
}

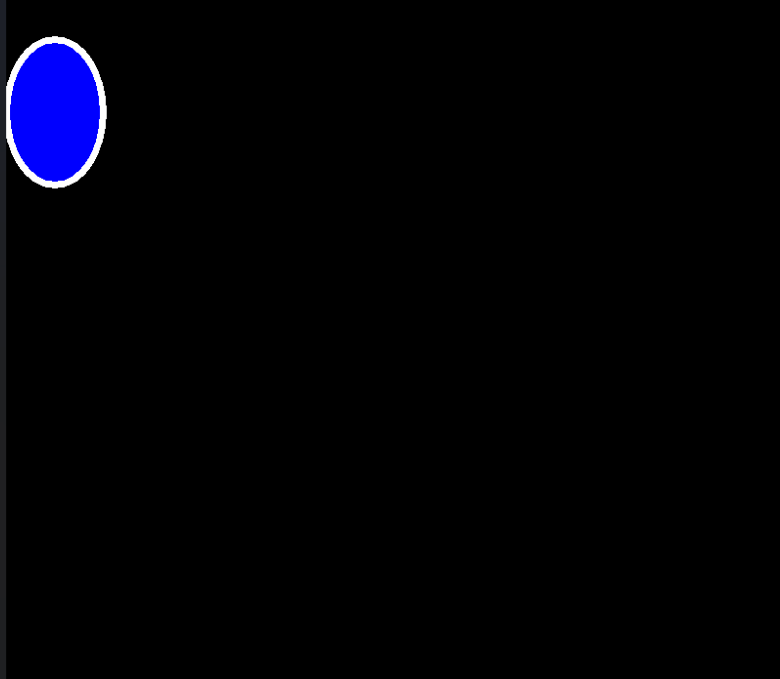
**Результаты выполнения программы:**









**Вывод:** я изучил структуры, свойства и виды объектов; изучил способы доступа к полям и правила вызова методов объектов; получил навыки объектно-ориентированного программирования на языке C/C++.