

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА АЭРОКОСМИЧЕСКИХ КОМПЬЮТЕРНЫХ И ПРОГРАММНЫХ СИСТЕМ

РУКОВОДИТЕЛЬ:

Кандидат техн. наук, доцент  
должность, уч. степень, звание

подпись, дата

Л.Н.Бариков  
инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №16

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: “КЛАССЫ”

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4143

подпись, дата

Е.Д.Тегай  
инициалы, фамилия

## Цель 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++.

## Индивидуальный вариант

Задание индивидуального варианта изображено на рисунке 1.

### Движение окружности по окружности.

Рисунок 1 – Задание варианта

## Текст программы

```
#define _USE_MATH_DEFINES
#include <cmath>
#include <conio.h>
#include <windows.h>
#include <iostream>

using namespace std;

class graph
{
protected:
    HWND hwnd; // дескриптор окна
public:
    HDC hdc;
    graph()
    {
        // функция, которая возвращает значение дескриптора приоритетного окна
        hwnd = GetForegroundWindow();
        // дескриптор контекста устройства
        hdc = GetDC(hwnd);
    }
};

class world :public graph
{
    POINT* arr;
    int x0;
    int y0;
public:
    float n;
    HPEN pBlack;
    HPEN pYellow;
    HPEN pRed;
    HBRUSH bGreen;
    HBRUSH bBlack;
public:
    world(int x0, int y0) //конструктор
    {
        // указатель, хранящий адрес конкретного объекта класса
        this->x0 = x0;
        this->y0 = y0;

        // создание перьев
        pBlack = CreatePen(PS_SOLID, 1, RGB(0, 0, 0));
        pYellow = CreatePen(PS_SOLID, 1, RGB(255, 255, 0));
        pRed = CreatePen(PS_SOLID, 1, RGB(255, 0, 0));
        // создание кистей
```

```

        bGreen = CreateSolidBrush(RGB(0, 255, 0));
        bBlack = CreateSolidBrush(RGB(0, 0, 0));

        double da = 0.15;
        double a = 0;
        int radCircle = x0 - 15;
        int radTraect = y0 - 15;

        n = 64;
        arr = new POINT[n];
        for (int i = 0; i < n; i++)
        {
            arr[i].x = x0 + radCircle * cos(a);
            arr[i].y = y0 + radTraect * sin(a);
            a += da;
        }
    }
    ~world() //деструктор
    {
        delete[] arr;
    }
    void clear()
    {
        SelectObject(hdc, bBlack);
        SelectObject(hdc, pBlack);
        Rectangle(hdc, 0, 0, x0 * 2, y0 * 2);
    }
    void path()
    {
        SelectObject(hdc, pYellow);
        Polyline(hdc, arr, n);
    }
    POINT getPoint(int n)
    {
        POINT p = { arr[n].x, arr[n].y };
        return p;
    }
};

class circle
{
    int step, ds;
    int a;
public:
    circle(int a) //конструктор
    {
        this->a = a;
        step = 0;
        ds = 1;
    }
    void show(world* wrd)
    {
        POINT p = wrd->getPoint(step);
        SelectObject(wrd->hdc, wrd->bGreen);
        SelectObject(wrd->hdc, wrd->pRed);
        Ellipse(wrd->hdc, p.x - a, p.y - a, p.x + a, p.y + a);
    }
    void go(world* wrd)
    {
        step += ds;
        if (step >= wrd->n) step = 0;
    }
};

int main()

```

```

{
    setlocale(LC_ALL, "Rus");
    int radCircle, radTraect;
    cout << "Пожалуйста, введите радиус окружности: " << endl;
    cin >> radCircle;
    cout << "Пожалуйста, введите радиус окружности, по которой будет двигаться
фигура:" << endl;
    cin >> radTraect;
    if (radCircle > 0 && radTraect > 0) {
        world* wrd = new world(radCircle + 310, radTraect + 310);
        circle* rc = new circle(radCircle);          /*создание объектов в динамической
                                                        памяти*/

        do
        {
            wrd->clear();
            wrd->path();
            rc->show(wrd);
            Sleep(250); // задержка в 250 мс
            rc->go(wrd);
        } while (!_kbhit()); //пока не нажата любая клавиша
        delete wrd;          //удаление объекта wrd из динамической памяти
        delete rc;           //удаление объекта rc из динамической памяти
        return 0;
    } cout << "Ошибка: извините, похоже, что данные введены некорректно ";
}

```

## Результаты работы программы

Результаты работы программы изображены на рисунках 2 - 4.

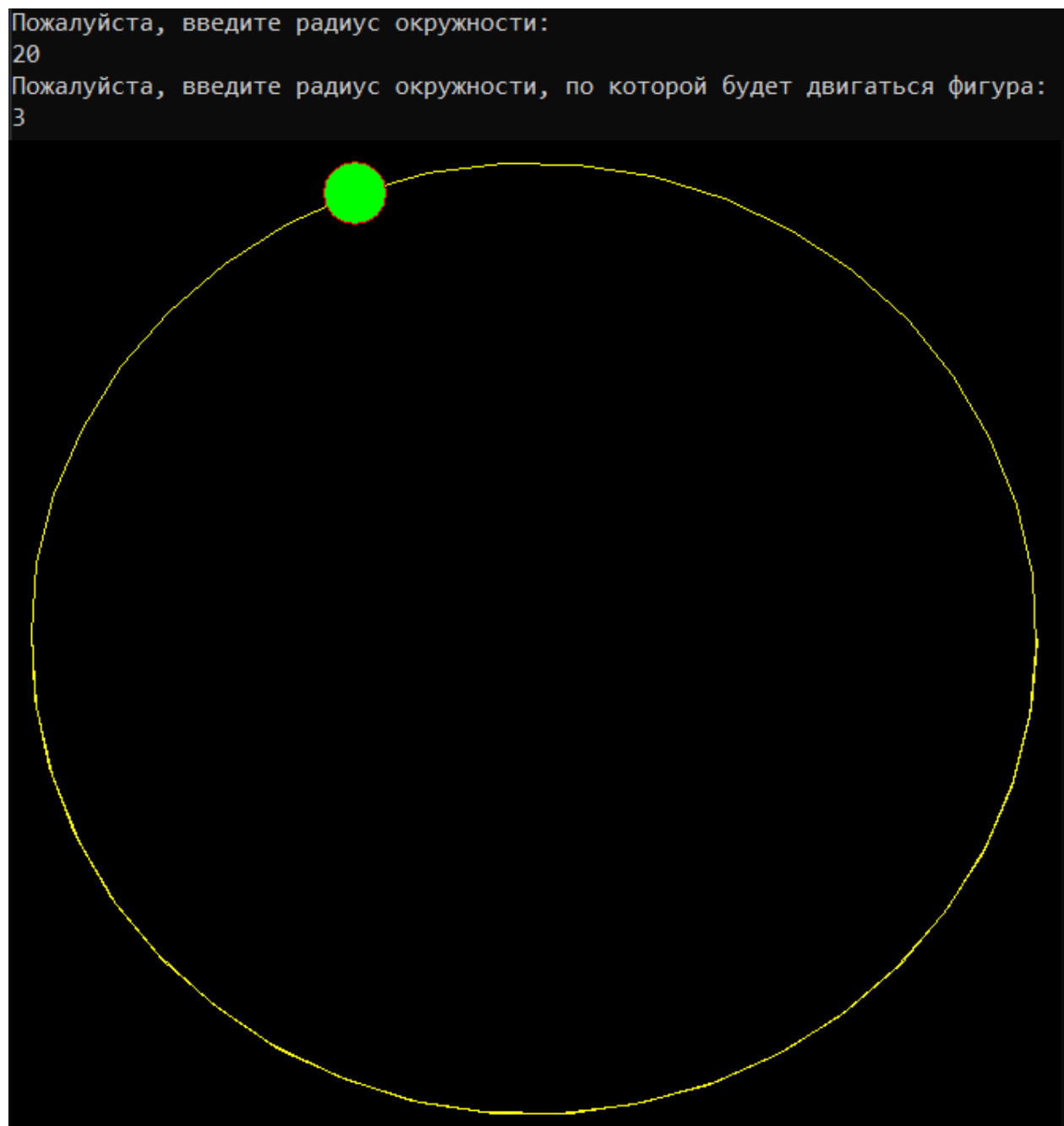


Рисунок 2 – Результат работы программы

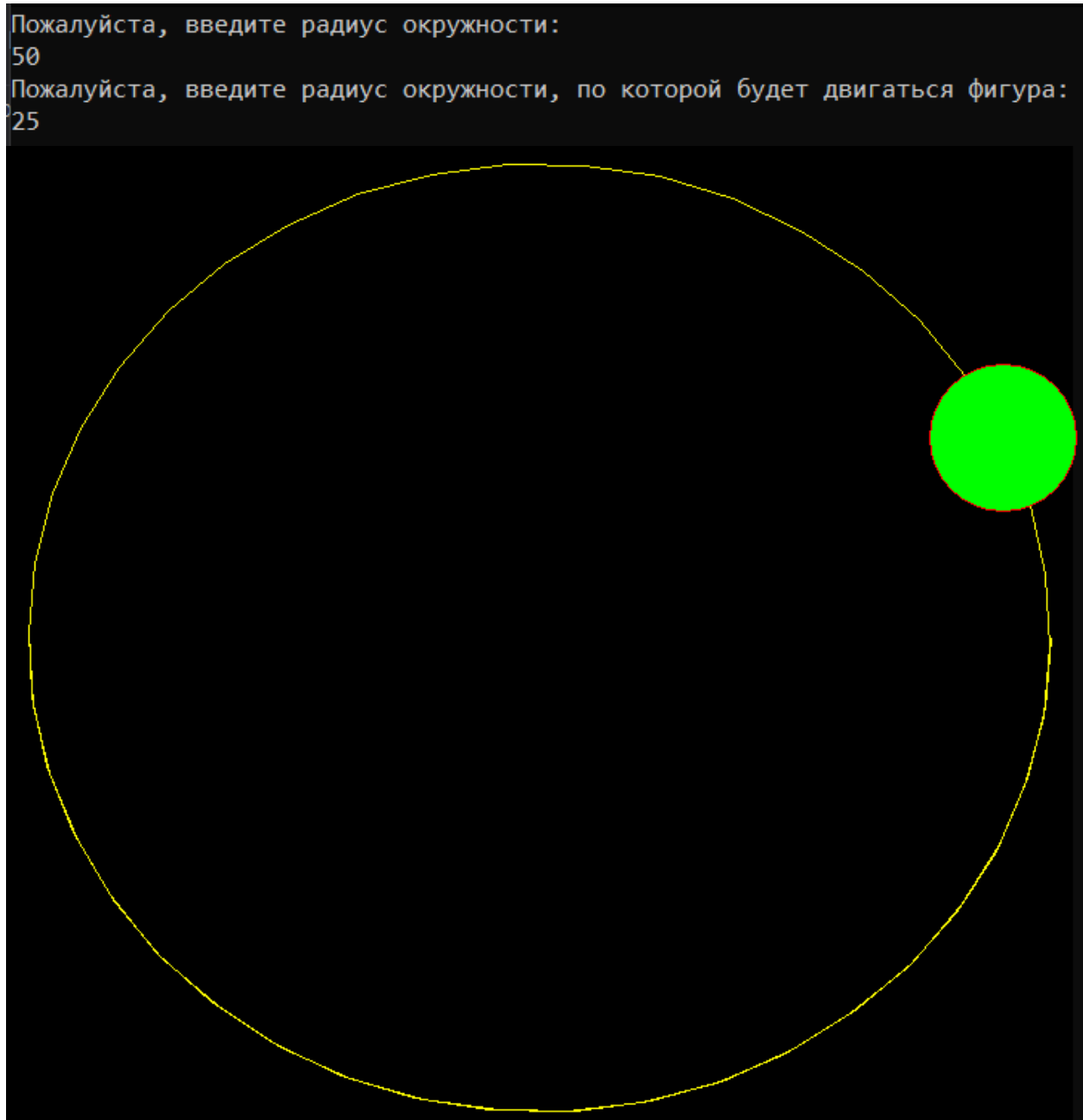


Рисунок 3 – Результат работы программы

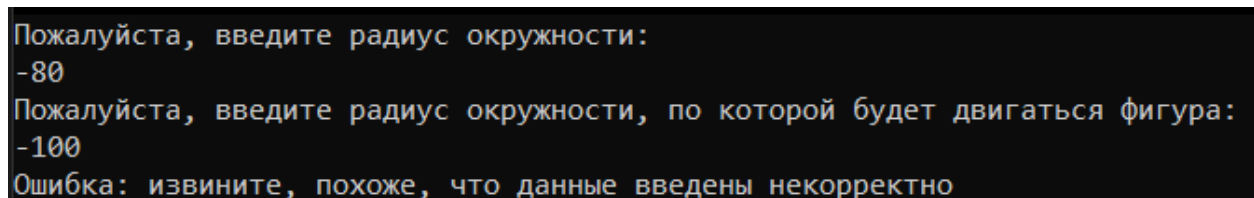


Рисунок 4 – Результат работы программы

## Вывод

В данной лабораторной работе были изучены структуры, свойства и виды объектов; изучены способов доступа к полям и правила вызова методов

объектов; получены навыки объектно-ориентированного программирования на языке C/C++.