**Минобрнауки России**

**Федеральное государственное образовательное учреждение высшего образования «Юго-Западный государственный университет»**

**Факультет фундаментальной и прикладной информатики**

**кафедра Программной инженерии**

По дисциплине «Компьютерная графика»

**ЛАБОРАТОРНАЯ РАБОТА №1**

КОМПЬЮТЕРНАЯ ГРАФИКА: ВВЕДЕНИЕ В ОБРАБОТКУ ИЗОБРАЖЕНИЙ

Выполнили: ст. гр. ПО-33б

Долженков А. А.

Иванов В. Н.

Проверил: Ефремов В. В.

Курск, 2025

# Вариант 4

# 1. Формулировка задачи

Необходимо разработать программу на языке C++ для обработки растрового изображения в формате PPM. Программа должна:  
1. Считать изображение из файла.  
2. Разместить точку цвета (255,127,127) в верхний левый угол, точку цвета (127,255,127) в центр верхней строки, точку цвета (127,127,255) в нижний левый угол.  
3. Сохранить результат в новый файл.  
4. Определить изменения в изображении и описать цвета точек.

Ссылка на открытый гит репозиторий https://github.com/diwelt /lr1-graph-

# 2. Программа

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include <sstream>

#include <cmath>

using namespace std;

struct RGB {

    unsigned char r, g, b;

};

struct Point {

    int x, y;

};

class Image1

{

public:

    int width, height;

    std::vector<RGB> pixels;

    Image1(const string& filepath)

    {

        ifstream stream(filepath, ios::binary);

        if (!stream) {

            cout << "Cant open file" << endl;

            return;

        }

        stream.ignore(100, '\n');

        stream >> width >> height;

        stream.ignore(100, '\n');

        stream.ignore(100, '\n');

        pixels.resize(width \* height);

        stream.read(reinterpret\_cast<char\*>(pixels.data()), pixels.size() \* sizeof(RGB));

    }

    void Replace\_Pixel(int x, int y, unsigned char r, unsigned char g, unsigned char b)

    {

        pixels[y \* width + x] = { r, g, b };

    }

    bool Save\_image(const string& filepath)

    {

        ofstream stream(filepath, ios::binary);

        if (!stream) {

            cout << "Cant make file" << endl;

            return false;

        }

        stream << "P6\n" << width << " " << height << "\n255\n";

        stream.write(reinterpret\_cast<const char\*>(pixels.data()), pixels.size() \* sizeof(RGB));

        return true;

    }

};

int main(int argc, char \*argv[])

{

    string input = "input.ppm";

    string output = "output.ppm";

    if(argc==2){

        input = argv[1];

    }else if(argc==3){

        input = argv[1];

        output = argv[2];

    }

    Image1 img(input);

    img.Replace\_Pixel(0,0,255,127,127);

    img.Replace\_Pixel(img.width/2,0,127,255,127);

    img.Replace\_Pixel(0,img.height-1,127,127,255);

    img.Save\_image(output);

}

# 3. Форма программы

Программа выполнена на языке C++ и запускается в консоли. Вызов осуществляется из командной строки с указанием входного и выходного файла:



# 4. Исходное и преобразованное изображения

Исходное изображение

A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A black square grid with many squares

AI-generated content may be incorrect.  
Преобразованное изображение – то же фото, но:

A screenshot of a cell phone

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A black and white pixelated background

AI-generated content may be incorrect.

- в левом верхнем углу появилась оранжевая точка (255,127,127);  
- в центре верхнем верхней строки появилась зеленая точка (127,255,127);  
- в нижнем левом углу появилась синяя точка (127,127,255).

# 5. Ответы на контрольные вопросы

1. Графика — раздел компьютерной науки, занимающийся представлением и обработкой изображений.

2. Оптическое изображение — результат взаимодействия света с объектами, фиксируемый глазом или камерой.

3. Сцена — совокупность объектов и их характеристик для визуализации.

4. Визуализация — преобразование данных в изображение.

5. Визуализация невозможного нужна для моделирования недоступных явлений.

6. Рендеринг — процесс построения изображения по цифровому описанию.

7. 3D-графика на 2D-мониторе — проекция трёхмерных объектов на экран.

8. Движущееся изображение получается последовательным выводом кадров (анимация).

9. По назначению графика бывает: научная, инженерная, художественная, игровая.

10. Растровая графика = пиксели; векторная графика = примитивы.

11. 4D-объекты отображаются проекцией на 3D, затем на 2D.

12. Цвет кодируется моделями (RGB, CMYK, HSV и др.).

13. Стереоизображения создают иллюзию глубины, но не содержат полной 3D-модели.

14. Фрактал — самоподобная геометрическая фигура.

15. Разрешение — количество пикселей на единицу длины.

16. Разрешающая способность измеряется в dpi (dots per inch).