

ГУАП

КАФЕДРА №43

ОТЧЁТ  
ЗАЩИЩЁН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
доцент  
должность, уч.степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
А.И. Попов  
инициалы, фамилия

## ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

### ОСВОЕНИЕ СРЕДЫ РАЗРАБОТКИ, ПОДКЛЮЧЕНИЯ ГРАФИЧЕСКОЙ БИБЛИОТЕКИ, СБОРКИ ПРИЛОЖЕНИЯ

по курсу:

Компьютерная графика

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр №

\_\_\_\_\_  
М411

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
И.К. Крылов  
инициалы, фамилия

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Задание на лабораторную работу .....	4
2 Реализация приложения .....	5
2.1 Выбор среды разработки и инструментов .....	5
2.2 Архитектура приложения .....	5
2.3 Используемые функции OpenGL .....	6
2.4 Конфигурация сборки .....	7
3 Листинг программы .....	8
4 Результаты работы программы .....	11
4.1 Сборка проекта .....	11
4.2 Демонстрация работы программы .....	11
ЗАКЛЮЧЕНИЕ .....	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	16

## ВВЕДЕНИЕ

Компьютерная графика является одной из важнейших областей современных информационных технологий. Графические приложения используются в науке, технике, медицине, развлечениях и во многих других сферах деятельности человека.

OpenGL (Open Graphics Library) представляет собой кросс-платформенный программный интерфейс для разработки приложений, использующих двумерную и трёхмерную компьютерную графику. Библиотека OpenGL широко применяется в промышленности и научных исследованиях благодаря своей универсальности, производительности и доступности.

Цель работы: освоить среду разработки и процесс подключения графической библиотеки OpenGL для выполнения лабораторных работ по дисциплине «Компьютерная графика».

Задачи работы:

- Настроить среду разработки Visual Studio с поддержкой CMake
- Подключить библиотеки OpenGL и GLU
- Создать приложение с вызовами графической библиотеки OpenGL
- Реализовать отрисовку базовых графических примитивов
- Скомпилировать и запустить приложение
- Продемонстрировать работу базовых функций OpenGL

Среда программирования: Visual Studio 2022 с системой сборки CMake и графической библиотекой OpenGL.

## 1 Задание на лабораторную работу

Запустить среду разработки, скомпилировать любое приложение с вызовами библиотеки OpenGL, при необходимости настроить работу с библиотеками и заголовочными файлами.

Рекомендуемые среды программирования:

- Microsoft Visual Studio (2005 Standard Edition и выше)
- Qt Creator с MinGW и OpenGL

Необходимые библиотеки:

- OpenGL (основная графическая библиотека)
- GLU (OpenGL Utility Library)
- GLUT или FreeGLUT (опционально)

Требования к выполнению:

1. Использовать одну из рекомендованных сред разработки
2. Подключить необходимые библиотеки OpenGL и GLU
3. Создать работающее приложение с вызовами OpenGL
4. Убедиться в корректной компиляции и запуске программы
5. Продемонстрировать работу графического приложения

## 2 Реализация приложения

### 2.1 Выбор среды разработки и инструментов

Для выполнения лабораторной работы была выбрана среда Microsoft Visual Studio 2022 Community Edition с системой сборки CMake.

Преимущества выбранной конфигурации:

- Полная поддержка стандарта C++17
- Встроенная поддержка CMake для кросс-платформенной разработки
- Современные инструменты отладки и анализа кода
- OpenGL встроен в Windows (библиотеки `opengl32.dll` и `glu32.dll`)

### 2.2 Архитектура приложения

Разработанное приложение демонстрирует базовые возможности OpenGL: создание окна, инициализацию контекста OpenGL и отрисовку графических примитивов.

Основные компоненты программы:

**Инициализация графической подсистемы.** Используется Windows API (WinAPI) для создания графического окна с поддержкой OpenGL. Функция

`CreateGLWindow()`

выполняет регистрацию класса окна, настройку формата пикселей (`PIXELFORMATDESCRIPTOR`

) и создание контекста рендеринга через функции

`wglCreateContext()`

и

`wglMakeCurrent()`

.

**Настройка OpenGL контекста.** Функция

`InitGL()`

инициализирует параметры рендеринга: включает плавное закрашивание (`GL_SMOOTH`

), настраивает тест глубины (

`GL_DEPTH_TEST`

) и устанавливает цвет фона. Функция

`ResizeGLScene()`

настраивает viewport и перспективную проекцию с углом обзора 45° через

`gluPerspective()`

.

**Отрисовка графической сцены.** Функция

`DrawGLScene()`

выполняет рендеринг двух примитивов: треугольника с градиентной заливкой (вершины окрашены в красный, зелёный и синий цвета) и квадрата голубого цвета.

Используются матричные преобразования (

`glTranslatef()`

) для позиционирования объектов в пространстве.

**Цикл обработки событий.** Главная функция

`WinMain()`

реализует стандартный цикл обработки сообщений Windows с использованием

`PeekMessage()`

. В каждой итерации цикла происходит отрисовка сцены и обмен буферов (

`SwapBuffers()`

) для реализации двойной буферизации.

**Освобождение ресурсов.** Функция

`KillGLWindow()`

корректно освобождает контекст рендеринга, device context и уничтожает окно при завершении работы приложения.

## 2.3 Используемые функции OpenGL

В таблице 1 представлены основные функции библиотеки OpenGL, использованные в программе.

Таблица 1 — Функции OpenGL, использованные в программе

Продолжение таблицы 1

Функция	Назначение
glShadeModel()	Установка режима закрашивания (плавное)
glClearColor()	Установка цвета фона (чёрный: 0.0, 0.0, 0.0)
glClear()	Очистка буферов цвета и глубины
glViewport()	Установка области вывода
glLoadIdentity()	Сброс текущей матрицы
glTranslatef()	Применение переноса объекта
glBegin() / glEnd()	Начало и завершение отрисовки примитивов
glColor3f()	Установка цвета вершины (RGB)
glVertex3f()	Задание координат вершины в 3D
gluPerspective()	Установка параметров перспективы

## 2.4 Конфигурация сборки

Файл

CMakeLists.txt

содержит настройки для сборки проекта. В нём указывается минимальная версия CMake (3.20), имя проекта, стандарт языка C++ (C++17), исходные файлы и библиотеки для линковки (

opengl32

и

glu32

).

### **3 Листинг программы**

Полный исходный код приложения представлен в листинге 1 ниже.



```

/*
 * Лабораторная работа №1 - Базовое OpenGL приложение
 * Демонстрация создания окна Win32 и отрисовки примитивов с помощью OpenGL
 */

#include <windows.h>
#include <gl/GL.h>
#include <gl/GLU.h>

// Глобальные переменные
HGLRC hRC = NULL;      // Rendering Context
HDC hDC = NULL;        // Device Context
HWND hWnd = NULL;      // Handle окна
HINSTANCE hInstance;    // Instance приложения

bool keys[256];         // Массив для хранения состояния клавиш
bool active = TRUE;     // Флаг активности окна
bool fullscreen = FALSE; // Режим полного экрана

// Прототипы функций
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

/*
 * Функция изменения размера окна OpenGL
 */
Lvoid ResizeGLScene(GLsizei width, GLsizei height)
{
    if (height == 0)
        height = 1;

    glViewport(0, 0, width, height);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // Установка перспективы
    gluPerspective(45.0f, (GLfloat)width / (GLfloat)height, 0.1f, 100.0f);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

```

Код содержит подробные комментарии на русском языке, поясняющие назначение каждой функции и основных блоков программы.

## 4 Результаты работы программы

### 4.1 Сборка проекта

Для сборки проекта выполнены следующие команды в командной строке:

```
# Конфигурация проекта для Visual Studio 2022
cmake -S . -B build -G "Visual Studio 17 2022" -A Win32

# Сборка в конфигурации Release
cmake --build build --config Release
```

Результаты компиляции:

— Исполняемый файл:

build\Release\Lab1\_OpenGL.exe

— Размер файла: 10 KB

— Время сборки: 3 секунды

— Ошибок компиляции: 0

— Предупреждений: 0

### 4.2 Демонстрация работы программы

При запуске исполняемого файла открывается графическое окно с заголовком «Лабораторная работа №1 - OpenGL». В окне отображаются два графических примитива: цветной треугольник и голубой квадрат.

Характеристики визуализации:

— Размер окна: 800×600 пикселей

— Цвет фона: чёрный (RGB: 0.0, 0.0, 0.0)

— Объект 1: треугольник с градиентной заливкой

— Вершины окрашены в красный (верхняя), зелёный (нижняя левая) и синий (нижняя правая) цвета

— Положение: слева от центра экрана

— Объект 2: квадрат голубого цвета

— Цвет: RGB (0.5, 0.5, 1.0)

— Положение: справа от центра экрана

Управление программой:

- **ESC** — закрытие приложения
- **Кнопка X** — стандартное закрытие окна

Программа использует двойную буферизацию для предотвращения мерцания изображения при отрисовке.



## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы №1 была достигнута поставленная цель — освоена среда разработки и процесс подключения графической библиотеки OpenGL для разработки графических приложений.

Основные результаты работы:

**Настройка среды разработки.** Успешно настроена среда Microsoft Visual Studio 2022 с системой сборки CMake для работы с графической библиотекой OpenGL. Изучены принципы конфигурирования проектов с использованием современных инструментов автоматизации сборки.

**Подключение библиотек OpenGL.** Корректно подключены системные библиотеки

opengl32.dll

и

glu32.dll

через механизм линковки в CMake. Изучена структура библиотек OpenGL и их взаимосвязь с операционной системой Windows.

**Разработка графического приложения.** Создано работающее приложение, демонстрирующее базовые возможности OpenGL: инициализацию контекста рендеринга, настройку viewport и проекции, отрисовку графических примитивов (треугольника и квадрата) с использованием цветовой интерполяции.

**Компиляция и тестирование.** Проект успешно скомпилирован без ошибок и предупреждений. Создан исполняемый файл размером 10 КВ. Приложение корректно работает в операционной системе Windows, создаёт графическое окно и отображает примитивы.

**Освоение API OpenGL.** В процессе выполнения работы изучены и практически применены основные функции библиотеки OpenGL для настройки контекста, управления матрицами, отрисовки примитивов и работы с цветом.

Полученные навыки:

- Работа с системой сборки CMake и конфигурирование проектов
- Подключение и использование внешних графических библиотек
- Работа с Windows API для создания графических окон
- Инициализация и настройка контекста рендеринга OpenGL
- Отрисовка графических примитивов и применение матричных преобразований
- Реализация цикла рендеринга с использованием двойной буферизации

Программа полностью соответствует требованиям лабораторной работы и готова для использования в качестве основы для выполнения последующих лабораторных работ по компьютерной графике. Полученный опыт позволяет перейти к изучению более сложных возможностей OpenGL: работе с текстурами, освещением, трёхмерными моделями и шейдерами.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Эйнджел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. М.: Вильямс, 2001. 592 с.
2. Хирн Д., Бейкер М. П. Компьютерная графика и стандарт OpenGL. 3-е изд. М.: Вильямс, 2005. 1168 с.
3. Райт Р. С., Липчак Б. OpenGL: суперкнига. 3-е изд. СПб.: Вильямс, 2006. 1040 с.
4. Шикин Е. В., Боресков А. В. Компьютерная графика. Динамика, реалистические изображения. М.: ДИАЛОГ-МИФИ, 1996. 288 с.
5. CMake Documentation [Электронный ресурс]. URL: <https://cmake.org/documentation/> (дата обращения: 09.01.2026).
6. OpenGL Reference Pages [Электронный ресурс]. URL: <https://registry.khronos.org/OpenGL-Refpages/> (дата обращения: 09.01.2026).
7. Microsoft Visual Studio Documentation [Электронный ресурс]. URL: <https://docs.microsoft.com/visualstudio/> (дата обращения: 09.01.2026).