**Задание 1. Система координат и примитивы OpenGL**

Цель выполнения задания: изучить на практике отображение простых геометрических фигур в пределах нормализованного объема видимости.

Изучаемые команды

Команды инициализации: auxInitWindowPosition, auxInitWindowSize, glutInitWindowPosition, glutInitWindowSize.

Команды настройки цветов фона и изображения: glClearColor, glClear, glColor.

Команды рисования геометрических примитивов: glVertex, glBegin – glEnd.

Команды задания стиля вычерчивания: glLineWidth, glEnable(GL\_LINE\_SMOOTH), glEnable(GL\_POINT\_SMOOTH).

Варианты заданий приведены в таблице 1.

Таблица 1

|  |  |  |
| --- | --- | --- |
| N варианта | Фигура | Геометрический примитив |
| 1 | Два раздельных треугольника | Отрезок (GL\_LINES), треугольник (GL\_TRIANGLES) |
| 2 | «Песочные часы» – два треугольника с общей вершиной | Связанные отрезки (GL\_LINE\_STRIP), треугольник (GL\_TRIANGLES) |
| 3 | Два связанных треугольника | Замкнутый контур (GL\_LINE\_LOOP) |
| 4 | Смежные треугольник и прямоугольник | Полигон (GL\_POLYGON) |
| 5 | Два смежных треугольника | Треугольник (GL\_TRIANGLES) |
| 6 | «Созвездие» из пяти разноцветных точек и «метеор» – вектор с крупной точкой | Точка (GL\_POINTS),отрезок (GL\_LINES) |
| 7 | Неправильный четырехугольник с точкой в вершине | Четырехугольник (GL\_QUADS), точка (GL\_POINTS) |
| 8 | Четырехугольник с диагональю | Связанные треугольники (GL\_TRIANGLE\_STRIP) |
| 9 | Четырехугольник с диагональю | Треугольник (GL\_TRIANGLES) |
| 10 | Два смежных четырехугольника | Связанные четырехугольники (GL\_QUAD\_STRIP) |
| 11 | Четырехугольник с диагональю | Связанные треугольники (GL\_TRIANGLE\_FAN) |
| 12 | Четырехугольник с диагональю | Связанные отрезки (GL\_LINE\_STRIP) |
| 13 | Параллелограмм с точками на углах | Точка (GL\_POINTS),полигон (GL\_POLYGON) |
| 14 | «Домик» – смежные треугольник и квадрат | Треугольник (GL\_TRIANGLES), четырехугольник (GL\_QUADS) |
| 15 | Треугольник и квадрат с общей вершиной | Замкнутый контур (GL\_LINE\_LOOP), четырехугольник (GL\_QUADS) |
| 16 | Треугольник, вписанный в прямоугольник | Треугольник (GL\_TRIANGLES), связанные отрезки (GL\_LINE\_STRIP) |
| 17 | Смежные пятиугольник и треугольник | Полигон (GL\_POLYGON) |
| 18 | «Флажок» – вытянутый прямоугольник и треугольник | Четырехугольник (GL\_QUADS), треугольник (GL\_TRIANGLES) |
| 19 | «Флажок» – вектор и невыпуклый пятиугольник | Отрезок (GL\_LINES),полигон (GL\_POLYGON) |
| 20 | «Кораблик» – трапеция и вектор | Четырехугольник (GL\_QUADS), отрезок (GL\_LINES) |
| 21 | «Молоток» – трапеция и вытянутый прямоугольник | Полигон (GL\_POLYGON), четырехугольник (GL\_QUADS) |
| 22 | «Окно» – прямоугольник и многоугольник | Замкнутый контур (GL\_LINE\_LOOP), полигон (GL\_POLYGON) |
| 23 | «Лопата» – выпуклый пятиугольник и вытянутый прямоугольник | Полигон (GL\_POLYGON), отрезок (GL\_LINES) |
| 24 | «Дупель один-один» – два смежных квадрата с точками в центрах | Четырехугольник (GL\_QUADS), точка (GL\_POINTS), |
| 25 | Пятиугольник с двумя диагоналями | Треугольник (GL\_TRIANGLES), отрезок (GL\_LINES) |
| 26 | «Веселый Роджер» – вытянутая по верти-кали трапеция и ниже – косой крест из двух прямоугольников | Четырехугольник (GL\_QUADS),полигон (GL\_POLYGON) |
| 27 | «Рюмка» – равнобокая трапеция на 6-угольной ножке | Полигон (GL\_POLYGON), замкнутый контур (GL\_LINE\_LOOP), |

Программа выполнения задания

1.​ Спроектировать изображение, выбрав координаты вершин, которые ограничивают примитивы. Координаты *х* и *у* вершин не должны превышать ±1, координата *z* берется равной нулю. Выбрать размеры окна вывода. Определить цвет фона (не черный) и цвет изображения: разный для каждого многоугольника.

2.​ В программу-заготовку добавить команды задания окна и рисования изображения. Использовать формат команды задания вершин glVertex3f. Получить изображение заданных фигур на экране. Предъявить результаты преподавателю.

3.​ Изменяя координаты *х* и *у* вершин в диапазоне (-5,+5), определить границы объема видимости по соответствующим координатам, а также направления координатных осей и точку их начала.

4.​ Задать координату *z* одной из вершин равной +0.5, затем -0.5. Объяснить наблюдаемый эффект. Задать координату *z* одной из вершин равной +5, затем -5. Объяснить наблюдаемый эффект.