Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

Уфимский государственный авиационный технический университет

Кафедра вычислительной математики и кибернетики

**Точка в 3-D пространстве**

Пояснительная записка по лабораторной работе№1

Дисциплина:

«Инженерная и компьютерная графика»

Выполнила:

студентка группы МО-213

Сиротенко Е.И.

Проверил:

канд. техн. наук Верхотуров М.А.

**Уфа - 2018**

**План:**

* Теоретическая часть
* Содержательная и формальная постановка задачи
* Структура решения (этапы решения и их взаимосвязь)
* Обзор и анализ методов решения(каждого этапа)
* Описание реализации применяемых методов
* Руководство программиста
* Руководство пользователя
* Заключение

**Теоретическая часть**

## *Ортогональная система трех плоскостей проекций*

В основу построения любого изображения положена операция проецирования.

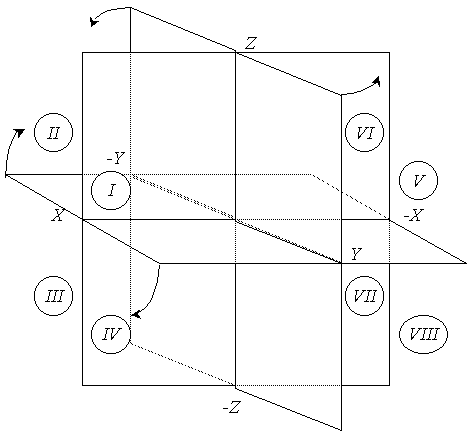
Сущность метода ортогонального проецирования заключается в том, что предмет проецируется на две взаимно перпендикулярные плоскости лучами, ортогональными (перпендикулярными) к этим осям.

Одну из плоскостей проекций П1 располагают горизонтально, а вторую П2 - вертикально. Плоскость П1 называют горизонтальной плоскостью проекций, П2 - фронтальной. Плоскости П1 и П2 бесконечны и непрозрачны.

При построении проекций необходимо помнить, что проекцией точки на плоскость называется основание перпендикуляра, опущенного из данной точки на эту плоскость. Проекцию точки Т на горизонтальную плоскость называют горизонтальной проекцией и обозначают Т1, проекцию точки Т на фронтальную плоскость - фронтальной проекцией и обозначают Т2. Каждая из них является основанием перпендикуляра, опущенного из данной точки А соответственно на плоскости П1 и П2. Две проекции точки определяют ее положение в пространстве. Так как каждая фигура или тело представляет собой совокупность точек, то можно утверждать, что две ортогональные проекции предмета (при наличии буквенных обозначений) вполне определяют его форму.

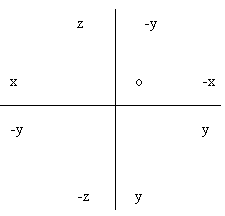
Однако в практике изображения строительных конструкций, машин и различных инженерных сооружений возникает необходимость в создании дополнительных проекций. Чтобы сделать проекционный чертеж более ясным и удобочитаемым, используют третью плоскость, перпендикулярную П1 и П2. Эта плоскость обозначается буквой П3 и называется профильной. Проекции точек на плоскость П3 называются профильными и обозначают Т3.

Плоскости проекций, попарно пересекаясь, определяют три оси: Ox, Oy и Oz, которые можно рассматривать как систему прямоугольных декартовых координат в пространстве с началом в точке O. Система знаков, соответствующая “правой системе” координат, показана на Рисунок1.

Три плоскости проекций делят пространство на восемь трехгранных углов - это так называемые октанты. Нумерация октантов дана на Рисунок 1.

*Рисунок 1 Ортогональная система плоскостей*

Рассматривая ортогональные проекции, предполагают, что наблюдатель находится в первом октанте. Проекционный чертеж, на котором плоскости проекций со всем тем, что на них изображено, совмещены определенным образом одна с другой, называется эпюром. Для получения эпюра плоскости П1 и П3 вращают как показано на Рисунок 2, до совмещения с плоскостью П2. В результате вращения передняя полуплоскость П1 оказывается совмещенной с нижней полуплоскостью П2, а задняя полуплоскость П1 - с верхней полуплоскостью П2.

Окончательный вид всех совмещенных плоскостей проекций

*Рисунок 2 Вид совмещенных плоскостей*

## *Три координаты и три проекции точки*

Координатами называют числа, которые ставят в соответствие точке для определения ее положения в пространстве или на поверхности.

В трехмерном пространстве положение точки устанавливают с помощью прямоугольных декартовых координат x, y, z.. Абсцисса x определяет расстояние от данной точки до плоскости П3, ордината y - до плоскости П2 и аппликата z- до плоскости П1. Приняв для отсчета координат точки систему, показанную на Рисунок 1 , составим таблицу знаков координат во всех восьми октантах (см. Рисунок 1).

|  |  |  |  |
| --- | --- | --- | --- |
| *Октант* | *Знаки координат* | | |
| x | y | z |
| I | + | + | + |
| II | + | - | + |
| III | + | - | - |
| IV | + | + | - |
| V | - | + | + |
| VI | - | - | + |
| VII | - | - | - |
| VIII | - | + | - |

*Таблица 1 Октанты и знаки координат*

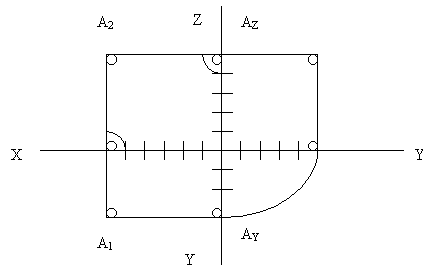
Какая-либо точка пространства A, заданная координатами, будет обозначаться так: A (x, y, z). Построение изображения самой точки и ее проекций на пространственной модели рекомендуется осуществлять с помощью координатного прямоугольного параллелепипеда. Прежде всего на осях координат от точки O откладывают отрезки, соответственно равные единицам длины. На этих отрезках (OAX, OAY, OAZ), как на ребрах, строят прямоугольный параллелепипед. Вершина его, противоположная началу координат, и будет определять заданную точку Т.

Построение параллелепипеда позволяет определить не только точку A, но и все три ее ортогональные проекции.

Лучами, проецирующими точку на плоскости П1, П2 и П3, являются те три ребра параллелепипеда, которые пересекаются в точке Т.

Каждая из ортогональных проекций точки A, будучи расположенной на плоскости, определяется только двумя координатами. Так, горизонтальная проекция Т1 определяется координатами x и y, фронтальная проекция Т2 - координатами x и z, профильная проекция Т3 - координатами y и z. Но две любые проекции определяются тремя координатами. Вот почему задание точки двумя проекциями равносильно заданию точки тремя координатами.

На эпюре (Рисунок 2), где все плоскости проекций совмещены, проекции Т1 и Т2 окажутся на одном перпендикуляре к оси OX, а проекции Т2 и Т3 - на одном перпендикуляре к оси OZ.



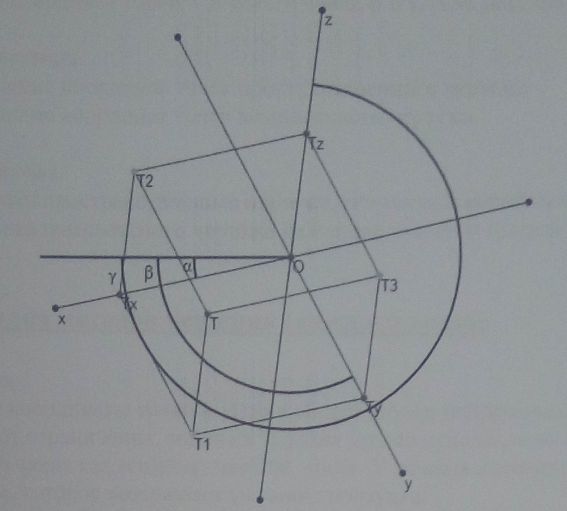
*Рисунок 3 Проекции точки (I октант) на комплексный чертеж*

*Точки Тх, Ту, Тz и углы α, β, γ между горизонталью экрана, направленной влево и осями координат Ох, Оу, Oz*

Точки Тх, Ту, Тz – это координаты точки Т, отложенные на осях координат Ох, Оу, Oz.

В рамках лабораторной работы №1 «Точка в 3D пространстве» мы делаем допущение, позволяя пользователю менять положение осей координат Ох, Оу, Oz. Для того, чтобы иметь возможность менять расположение осей, используем углы между горизонталью экрана, направленной влево, и осями координат.

Таким образом, имеем следующий пространственный чертеж с используемыми обозначениями, которые предстоит отобразить на экране:

*Рисунок 4 Пространственный чертеж*

**Содержательная и формальная постановка задачи**

Содержательная постановка задачи:

Разработать программу, реализующую отображение точки, заданной тремя координатами, на пространственном и комплексном чертежах на экране компьютера с возможностью изменять координаты точки, а также с возможностью изменять углы координатных осей.

Формальная постановка задачи:

Разработать программу, реализующую вывод на экран:

* аксонометрического чертежа с изображением точки *Т*, ее проекций *T1,Т2,Т3*и линий проекционной связи (в виде параллелепипеда),
* комплексного чертежа с изображением проекций точки,
* ползунковых переключателей (6 шт.) для интерактивного изменения координат *(x, y, z)* и углов координатных осей (α, β, γ).

**Структура решения задачи (этапы решения)**

1. Ввод координат точки Т и трех углов α, β, γ.
2. Вычисления:
   1. Вычисление координат точек пространственного чертежа
   2. Вычисление координат точек комплексного чертежа
3. Отрисовка
   1. Отрисовка пространственного чертежа
   2. Отрисовка комплексного чертежа

**Обзор и анализ методов решения (каждого этапа)**

1. Ввод данных

Нужно ввести в программу информацию о координатах точки и об углах наклона осей.

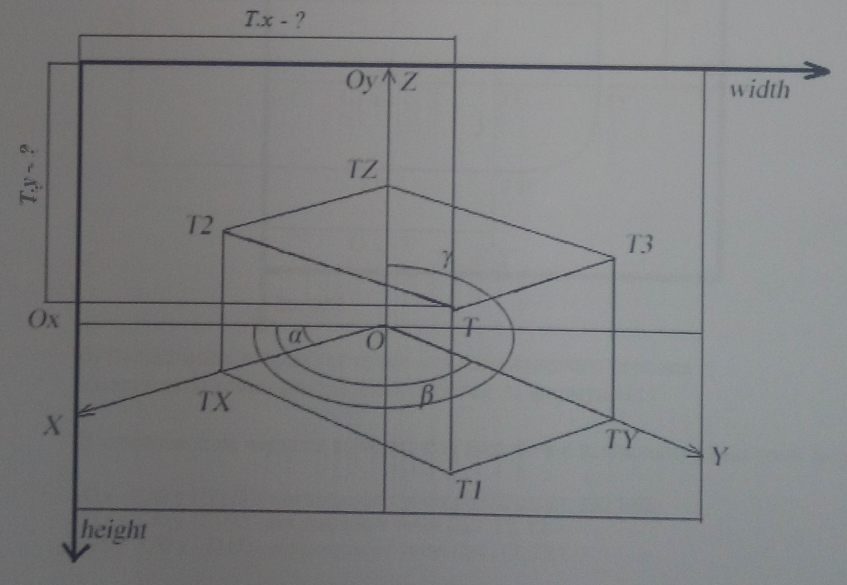
Для ввода углов и координат точки пользователю будет удобнее использовать непрерывный компонент, который представляет собой элемент управления в виде ползунка, который можно перемещать курсором мыши во время выполнения.

Обобщенный алгоритм:

1. Считать значения координат и записать в соответствующие переменные.
2. Считать значения углов наклона осей и записать в соответствующие переменные.
3. Вычисления

Дано:

* Координаты точки в трехмерном пространстве
* Значения углов наклона осей (α, β, γ)
* Значения width и height области рисования



* 1. Вычисление координат точек для пространственного чертежа

Для построения пространственного чертежа точки T нужно найти:

- экранные координаты начала отсчета O пространственного чертежа;

- экранные координаты точки T;

- экранные координаты ее проекций T1, T2, T3;

- экранные координаты ее проекций на координатные оси – TХ, TY, TZ;

- экранные координаты точек начала и конца отрисовки осей.

Координаты точки начала отсчета О

О.х = width/2 (1)

O.y = height/2

Координаты точки Т вычисляются по формуле:

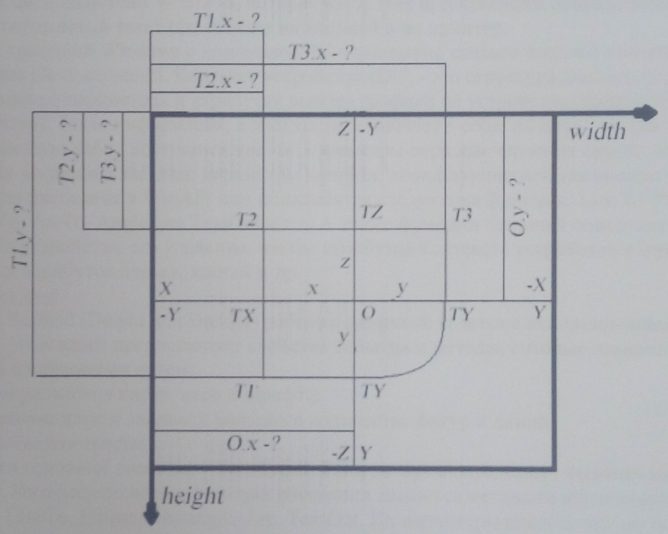
Т.х = O.x – x \* cos(α) – y \* cos(β) – z \* cos(γ) (2)

Т.у = O.y + x \* sin(α) + y \* sin(β) + z \* sin(γ)

Координаты точек TX, TY, TZ, T1, T2, T3, а также координаты точек отрисовки осей являются производными этой формулы.

Обобщенный алгоритм

1. Высчитываем координаты центра с помощью формулы (1)
2. Высчитываем координаты точки Т, ее проекций и точек отрисовки осей с помощью формул (2)



* 1. Вычисление координат точек для комплексного чертежа

Координаты точки начала отсчета вычисляются по формуле (1)

В комплексном чертеже координаты всех точек вычисляются по трем формулам:

(О.х – Х, О.у + Y) (3) – для координат точек Ох1, Оу2, Т1, ТХ, ТY3;

(O.x – X, O.y – Z) (4) – для координат точек Ох2, Т2, TZ;

(O.x + Y, O.y – Z) (5) – для координат точек Oy1, T3, TY1

Обобщенный алгоритм

1. Высчитываем координаты точки начала отсчета с помощью формулы (1)
2. Высчитываем координаты проекций точки Т и точек отрисовки осей с помощью формул (3,4,5)
3. Отрисовка

Для того чтобы вывести необходимые чертежи на экран используется Интерфейс Графических Устройств GDI(Graphics Device Interface), представляющий собой интерфейс, который Windows использует для рисования 2D графики.

Работа выполнена в MS Visual C#. Для изображения точек, отрезков, дуги, вывода текста используются различные функции: FillEllipse, DrawLine, DrawArc, DrawString (уже определенные в GDI).

При каждом изменении положения одного из ползунков появляется необходимость в перерисовке тех или иных компонентов результирующего изображения. Во время перерисовки экрана, панель отображения рисунка стирается и лишь затем, рисуется новый чертеж. Это служит причиной возникновения мерцания на экране. Для решения этой проблемы используется параметр DoubleBuffered(двойная буферизация). Суть двойной буферизации заключена в использовании двух буферов – первичного и вторичного. Все операции рисования выполняются на вторичном буфере, а вывод картинки на экран – с первичного. После того, как на вторичном буфере будут отрисованы все примитивы, этот вторичный буфер становится первичным, а первичный – вторичным (меняются местами, программно - указателями). С «нового» первичного буфера (который был вторичным) изображение выводится на экран, а на «новый» вторичный (который был первичным) – рисуются примитивы. В дальнейшем такое чередование продолжается.

Обобщенный алгоритм

1. Пространственный чертеж

*Отрисовка осей X, Y, Z:*

Линия (Ох1, Ох2)

Линия (Oy1, Oy2)

Линия (Oz1, Oz2)

*Соединение точек проекции и точки Т*

Линия (Т1, ТХ)

Линия (Т1, ТY)

Линия (Т2, ТХ)

Линия (Т2, ТZ)

Линия (Т3, ТY)

Линия (Т3, ТZ)

Линия (Т1, Т)

Линия (Т2, Т)

Линия (Т2, Т)

*Отрисовка точек О, Т, Т1, Т2, Т3, ТХ, ТY, TZ с их именами*

Эллипс (О) Эллипс (Т)

Текст (О) Текст (Т)

Эллипс (Т1) Эллипс (ТХ)

Текст (Т1) Текст (ТХ)

Эллипс (Т2) Эллипс (ТY)

Текст (Т1) Текст (ТY)

Эллипс (Т3) Эллипс (TZ)

Текст (Т3) Текст (TZ)

1. Комплексный чертеж

*Отрисовка осей:*

Линия (Ox1, Ox2)

Линия (Oy1, Oy2)

*Соединение точек проекции*

Линия (T1, TX) Линия (T2, TZ)

Линия (T1, TY1) Линия (T3, TY3)

Линия (T2, TX) Линия (T3, TZ)

*Отрисовка точек O, T1, T2, T3, TX, TY1, TY3, TZ с их именами*

Эллипс (О) Эллипс (ТХ)

Текст (О) Текст (ТХ)

Эллипс (T1) Эллипс (TY1)

Текст (T1) Текст (TY1)

Эллипс (T2) Эллипс (TY3)

Текст (T2) Текст (TY3)

Эллипс (T3) Эллипс (TZ)

Текст (T3) Текст (TZ)

*Соединяющая дуга с учетом знака координат Y*

Дуга (TY1, TY3)

**Описание реализации применяемых методов**

Типы и структуры данных:

Переменная length содержит значение длины осей.

Тип Point3D хранит значения координат точки в трехмерном пространстве, а также имена точек.

Point3D{

X; - координата по Х

Y; - координата по Y

Z; - координата по Z

Name; - название точки

}

Тип Point2D хранит значения координат точки в плоскости экрана, а также имена точек.

Point2D{

X; - координата по Х

Y; - координата по Y

Name; - название точки

}

Массив Point3D\_mas, состоящий из 14 элементов структуры Point3D, хранит координаты точки Т, ее проекций и координаты отрисовки осей в трехмерном пространстве.

* 0 элемент – координаты точки О
* 1-3 элементы – координаты начала отрисовки осей
* 4-6 элементы – координаты конца отрисовки осей
* 7 элемент – координаты точки T
* 8-10 элементы – координаты точек TX, TY, TZ
* 11-13 элементы – координаты точек проекции Т1, Т2, Т3

Массив Point2d\_mas, состоящий из 14 элементов структуры Point2D, хранит координаты точки Т, ее проекций и координаты отрисовки осей в плоскости экрана на пространственном чертеже.

* 0 элемент – координаты точки О
* 1-3 элементы – координаты начала отрисовки осей
* 4-6 элементы – координаты конца отрисовки осей
* 7 элемент – координаты точки T
* 8-10 элементы – координаты точек TX, TY, TZ
* 11-13 элементы – координаты точек проекции Т1, Т2, Т3

Массив PointComplex\_mas, состоящий из 12 элементов структуры Point2D, хранит координаты проекций точки Т и координаты отрисовки осей в плоскости экрана на комплексном чертеже.

* 0 элемент – координаты точки О
* 1-2 элементы – координаты начала отрисовки осей
* 3-4 элементы – координаты конца отрисовки осей
* 5-8 элементы – координаты точек TX, TY1, TY3, TZ
* 9-11 элементы – координаты точек проекции Т1, Т2, Т3

**Укрупненный алгоритм**

1. Ввод значений координат точки Т – X, Y, Z и трех углов α, β, γ.
   1. Получаем координаты точки T и заполняем массив Point3D\_mas.

Точка О: Point3D\_mas[0] = (0, 0, 0)

Начало осей Конец осей

Point3D\_mas[1] = (length, 0, 0) Point3D\_mas[4]=(-length, 0, 0)

Point3D\_mas[2] = (0, length, 0) Point3D\_mas[5]=(0, -length, 0)

Point3D\_mas[3] = (0, 0, length) Point3D\_mas[6]=(0, 0, -length)

Точка Т: Point3D\_mas[7] = (X, Y, Z)

Точки TX, TY, TZ

Point3D\_mas[8] = (X, 0, 0)

Point3D\_mas[9] = (0, Y, 0)

Point3D\_mas[10] = (0, 0, Z)

Точки Т1, Т2, Т3

Point3D\_mas[11] = (X, Y, 0)

Point3D\_mas[12] = (X, 0, Z)

Point3D\_mas[13] = (0, Y, Z)

* 1. Получаем значения трех углов α, β, γ и записываем их в соответствующие переменные.

a = α b = β c = γ

1. Вычисления:
   1. Пространственный чертеж

Для создания пространственного чертежа координаты всех точек преобразуются из трехмерных в массиве Point3D\_mas в двумерные, записываемые в массив Point2D\_mas, в цикле с 0 по 13 элемент массива с помощью формулы (2).

* 1. Комплексный чертеж

Для расчета координат точки О (PointComplex\_mas[0]) используем формулу (1)

С помощью формулы (3) высчитываем координаты точек:

Ox1 (PointComplex\_mas[1])

Ox2 (PointComplex\_mas[4])

T1 (PointComplex\_mas[9])

TX (PointComplex\_mas[5])

TY2 (PointComplex\_mas[7])

С помощью формулы (4) высчитываем координаты точек:

Ox2 (PointComplex\_mas[3])

T2 (PointComplex\_mas[10])

TZ (PointComplex\_mas[8])

С помощью формулы (5) высчитываем координаты точек:

Oy1 (PointComplex\_mas[2])

T3 (PointComplex\_mas[11])

TY1 (PointComplex\_mas[6])

1. Отрисовка
   1. Пространственный чертеж

*Отрисовка осей:*

С помощью метода DrawLine() соединяем попарно точки:

Ox1 – Ox2 (Point2D\_mas[1] – Point2D\_mas[4]) – ось Х

Oy1 – Oy2 (Point2D\_mas[2] – Point2D\_mas[5]) – ось Y

Oz1 – Oz2 (Point2D\_mas[3] – Point2D\_mas[6]) – ось Z

*Соединение точек проекции:*

С помощью метода DrawLine() соединяем попарно точки:

Т1 – TX (Point2D\_mas[11] – Point2D\_mas[8])

T1 – TY (Point2D\_mas[11] – Point2D\_mas[9])

T2 – TX (Point2D\_mas[12] – Point2D\_mas[8])

T2 – TZ (Point2D\_mas[12] – Point2D\_mas[10])

T3 – TY (Point2D\_mas[13] – Point2D\_mas[9])

T3 – TZ (Point2D\_mas[13] – Point2D\_mas[10])

T1 – T (Point2D\_mas[11] – Point2D\_mas[7])

T2 – T (Point2D\_mas[12] – Point2D\_mas[7])

T3 – T (Point2D\_mas[13] – Point2D\_mas[7])

*Отрисовка точек О, T, T1, T2, T3, TX, TY, TZ:*

С помощью методов FillEllipse() в цикле отрисовываем точки по координатам из массива Point2D\_mas с 0 по 13 элементы. По этим же координатам отрисовываем текст с названием точки с помощью DrawString().

* 1. Комплексный чертеж

*Отрисовка осей:*

С помощью метода DrawLine() соединяем попарно точки:

Ox1 – Ox2 (PointComplex\_mas[1] – PointComplex\_mas[3])

Oy1 – Oy2 (PointComplex\_mas[2] – PointComplex\_mas[4])

*Соединение точек проекции:*

С помощью метода DrawLine() соединяем попарно точки:

Т1 – TX (PointComplex\_mas[9] – PointComplex\_mas[5])

T1 – TY2 (PointComplex\_mas[9] – PointComplex\_mas[7])

T2 – TX (PointComplex\_mas[10] – PointComplex\_mas[5])

T2 – TZ (PointComplex\_mas[10] – PointComplex\_mas[8])

T3 – TY1 (PointComplex\_mas[11] – PointComplex\_mas[6])

T3 – TZ (PointComplex\_mas[11] – PointComplex\_mas[8])

*Отрисовка точек О, T1, T2, T3, TX, TY, TZ:*

С помощью методов FillEllipse() в цикле отрисовываем точки по координатам из массива PointComplex\_mas с 0 по 11 элементы. По этим же координатам отрисовываем текст с названием точки с помощью DrawString().

*Отрисовка соединяющей дуги:*

С помощью метода DrawArc() соединяем две точки для отрисовки дуги. Начальный и конечный угол поворота определяются по знаку координаты Y.

TY1 – TY2 (PointComplex\_mas[6] – PointComplex\_mas[7])

**Детализированный алгоритм**

Так как все координаты в пространственном чертеже вычисляются одной и той же формулой имеет смысл выделить их вычисление в отдельную функцию.

Projection2D (point3d, point2d, a, b, c, O)

Point2d.x=O.x – cos(a)\*point3d.x – cos(b)\*point3d.y – cos(c)\*point3d.z

Point2d.y = O.y + sin(a)\*point3d.x + sin(b)\*point3d.y + sin(c)\*point3d.z

* point3d – координаты точки в трехмерном пространстве
* point2d – координаты точки в экранной системе координат
* a, b, c – углы между горизонтальным вектором и осями
* O – координаты центра

1. Ввод значений координат точки T – X, Y, Z и трех углов α, β, γ.
   1. Считываем с ползунковых переключателей координаты точки T и заполняем массив Point3D\_mas.

X = trackBar\_X.Value;

Y = trackBar\_Y.Value;

Z = trackBar\_Z.Value;

Point3D\_mas[0] = (0, 0, 0) - точка О

Point3D\_mas[1] = (length, 0, 0) – точка Ox1

Point3D\_mas[2] = (0, length, 0) – точка Oy1

Point3D\_mas[3] = (0, 0, length) – точка Oz1

Point3D\_mas[4]=(-length, 0, 0) – точка Ox2

Point3D\_mas[5]=(0, -length, 0) – точка Oy2

Point3D\_mas[6]=(0, 0, -length) – точка Oz2

Point3D\_mas[7] = (X, Y, Z) – T

Point3D\_mas[8] = (X, 0, 0) – TX

Point3D\_mas[9] = (0, Y, 0) – TY

Point3D\_mas[10] = (0, 0, Z) – TZ

Point3D\_mas[11] = (X, Y, 0) – T1

Point3D\_mas[12] = (X, 0, Z) – T2

Point3D\_mas[13] = (0, Y, Z) – T3

* 1. Считываем с ползунковых переключателей три угла - α, β, γ, заносим в соответствующие переменные.

a = trackBar\_a.Value;

b = trackBar\_b.Value;

c = trackBar\_c.Value;

1. Вычисления:
   1. Пространственный чертеж

Для расчёта координат точки О воспользуемся формулой (1)

Point2D\_mas[0].x = width/2;

Point2D\_mas[0].y = height/2;

Для создания пространственного чертежа координаты всех точек преобразуются с помощью формулы (2) из трехмерных точек массива Point3D\_mas в двумерные, записываемые в массив Point2D\_mas.

Для i от 1 до 13 с шагом 1

Projection2d(Point3D\_mas[i], Point2D\_mas[i], a, b, c, Point2D\_mas[0])

* 1. Комплексный чертеж

Для расчёта координат точки О воспользуемся формулой (1)

PointComplex\_mas[0].x = width/2;

PointComplex\_mas[0].y = height/2;

С помощью формул (3, 4, 5) высчитываем координаты всех точек комплексного чертежа

PointComplex\_mas[1] = (PointComplex\_mas[0].x – lenght, PointComplex\_mas[0].y)

PointComplex\_mas[2] = (PointComplex\_mas[0].x, PointComplex\_mas[0].y + length)

PointComplex\_mas[3] = (PointComplex\_mas[0].x + length, PointComplex\_mas[0].y)

PointComplex\_mas[4] = (PointComplex\_mas[0].x, PointComplex\_mas[0].y – lenght)

PointComplex\_mas[5] = (PointComplex\_mas[0].x – X, PointComplex\_mas[0].y)

PointComplex\_mas[6] = (PointComplex\_mas[0].x + Y, PointComplex\_mas[0].y)

PointComplex\_mas[7] = (PointComplex\_mas[0].x, PointComplex\_mas[0].y + Y)

PointComplex\_mas[8] = (PointComplex\_mas[0].x, PointComplex\_mas[0].y – Z)

PointComplex\_mas[9] = (PointComplex\_mas[0].x – X, PointComplex\_mas[0].y + Y)

PointComplex\_mas[10] = (PointComplex\_mas[0].x – X, PointComplex\_mas[0].y – Z)

PointComplex\_mas[11] = (PointComplex\_mas[0].x + Y, PointComplex\_mas[0].y – Z)

1. Отрисовка
   1. Пространственный чертеж

*Отрисовка осей:*

DrawLine (Point2D\_mas[1] – Point2D\_mas[4]) – ось Х

DrawLine (Point2D\_mas[2] – Point2D\_mas[5]) – ось Y

DrawLine (Point2D\_mas[3] – Point2D\_mas[6]) – ось Z

*Соединение точек проекции:*

DrawLine (Point2D\_mas[11] – Point2D\_mas[8]) Т1 – TX

DrawLine (Point2D\_mas[11] – Point2D\_mas[9]) T1 – TY

DrawLine (Point2D\_mas[12] – Point2D\_mas[8]) T2 – TX

DrawLine (Point2D\_mas[12] – Point2D\_mas[10]) T2 – TZ

DrawLine (Point2D\_mas[13] – Point2D\_mas[9]) T3 – TY

DrawLine (Point2D\_mas[13] – Point2D\_mas[10]) T3 – TZ

DrawLine (Point2D\_mas[11] – Point2D\_mas[7]) T1 – T

DrawLine (Point2D\_mas[12] – Point2D\_mas[7]) T2 – T

DrawLine (Point2D\_mas[13] – Point2D\_mas[7]) T3 – T

*Отрисовка точек О, Т, Т1, Т2, Т3, TX, TY, TZ*

для i от 0 до 13 с шагом 1

FillEllipse (Point2D\_mas[i]);

DrawString (Point2D\_mas[i].Name, Point2D\_mas[i])

* 1. Комплексный чертеж

*Отрисовка осей:*

DrawLine (PointComplex\_mas[1] – PointComplex\_mas[3]) – ось Х

DrawLine (PointComplex\_mas[2] – PointComplex\_mas[4]) – ось Y

*Соединение точек проекции:*

DrawLine (PointComplex\_mas[9] – PointComplex\_mas[5]) T1 – TX

DrawLine (PointComplex\_mas[9] – PointComplex\_mas[7]) T1 – TY

DrawLine (PointComplex\_mas[10] – PointComplex\_mas[5]) T2 – TX

DrawLine (PointComplex\_mas[10] – PointComplex\_mas[8]) T2 – TZ

DrawLine (PointComplex\_mas[11] – PointComplex\_mas[6]) T3 – TY

DrawLine (PointComplex\_mas[11] – PointComplex\_mas[8]) T3 - TZ

*Отрисовка точек О, T1, T2, T3, TX, TY, TZ:*

для i от 0 до 11 с шагом 1

FillEllipse (PointComplex\_mas[i]);

DrawString (PointComplex\_mas[i].Name, PointComplex\_mas[i])

*Отрисовка соединяющей дуги:*

Начальный и конечный угол поворота определяются по знаку координаты Y.

Если Y>0

DrawArc(PointComplex[6], PointComplex[7], 0, 90)

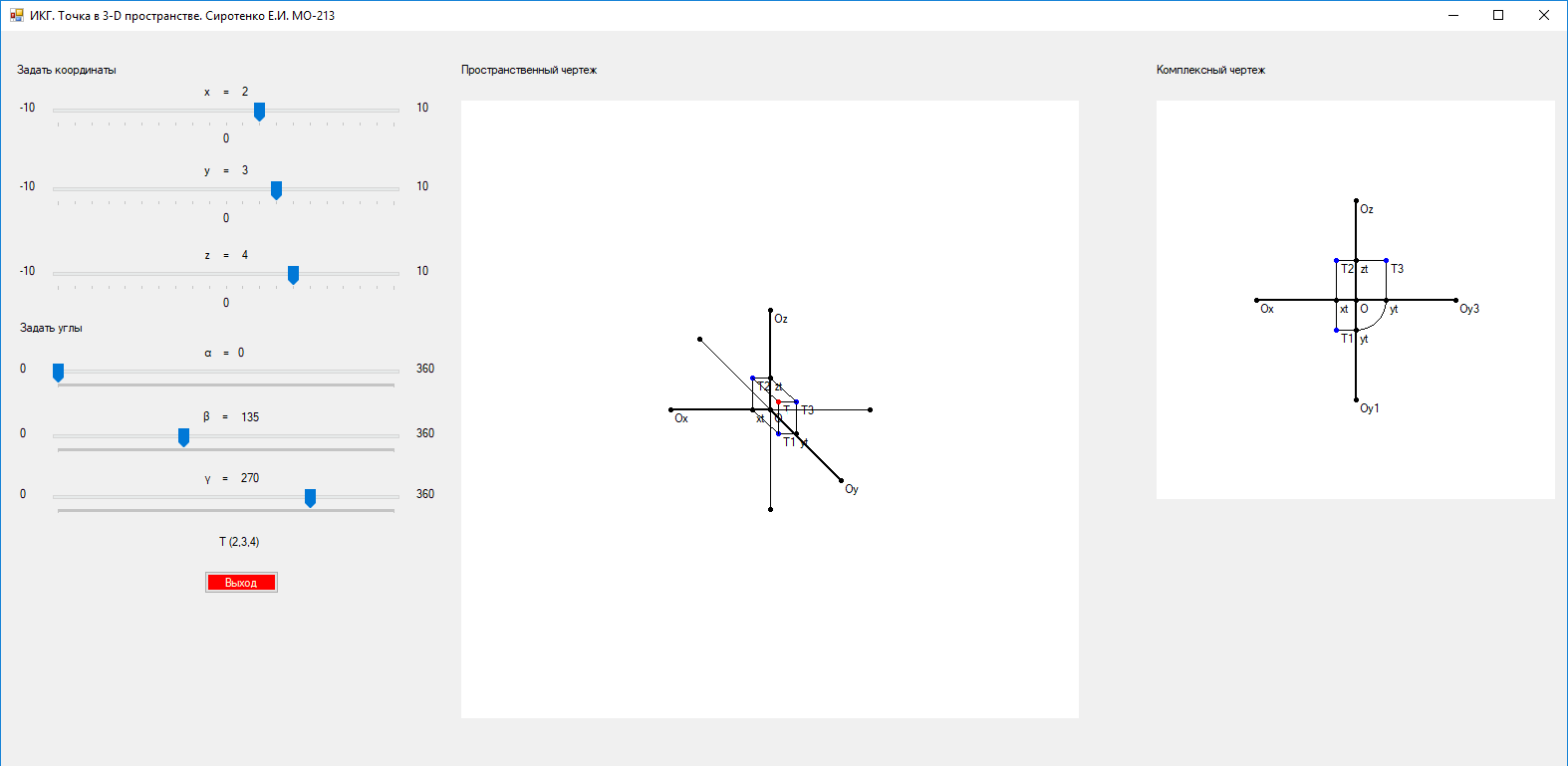
Если Y<0

DrawArc(PointComplex[6], PointComplex[7], 180 90)

**Руководство программиста**

Данное приложение разработано на платформе .NET с использованием среды разработки VisualDtudio на языке программирования C#.

*Интерфейс*

 Для вывода изображения используется компонента Panel. Для ввода данных используются ползунковые переключатели TrackBar. Кнопка «Выход» реализована с помощью компоненты Button. Весь необходимый текст выводится посредством элемента Label.

*Описание элементов, расположенных на экране:*

*TrackBar*

trackbar\_X – ползунок для изменения координаты X

trackbar\_Y – ползунок для изменения координаты У

trackbar3\_Z – ползунок для изменения координаты Z

trackbar\_a – ползунок для изменения угла наклона оси X

trackbar\_b – ползунок для изменения угла наклона оси Y

trackbar\_c – ползунок для изменения угла наклона оси Z

*Panel*

panel2D – объект для отрисовки пространственного чертежа

panelComplex – объект для отрисовки комплексного чертежа

*Button*

button\_exit – кнопка для выхода из программы

*Label*

label\_2D – отрисовка подписи для пространственного чертежа

label\_Complex – отрисовка подписи для комплексного чертежа

label\_angle – отрисовка подписи для переключателей, изменяющих угол

label\_coor – отрисовка подписи для переключателей, изменяющих координаты

label\_a – отрисовка текущего значения угла α

label\_b – отрисовка текущего значения угла β

label\_c – отрисовка текущего значения угла γ

label\_maxa – отрисовка максимального значения α

label\_mina – отрисовка минимального значения α

label\_maxb – отрисовка максимального значения β

label\_minb – отрисовка минимального значения β

label\_maxc – отрисовка максимального значения γ

label\_minc – отрисовка минимального значения γ

label\_x – отрисовка текущего значения координаты X

label\_y – отрисовка текущего значения координаты Y

label\_z – отрисовка текущего значения координаты Z

label\_maxx – отрисовка максимального значения координаты Х

label\_minx – отрисовка минимального значения координаты Х

label\_maxy – отрисовка максимального значения координаты Y

label\_miny – отрисовка минимального значения координаты Y

label\_maxz – отрисовка максимального значения координаты Z

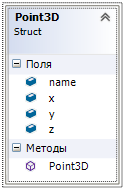
label\_minz – отрисовка минимального значения координаты Z

label\_zerox – отрисовка значения 0 координаты X

label\_zeroy – отрисовка значения 0 координаты Y

label\_zeroz – отрисовка значения 0 координаты Z

*Используемые структуры и классы*

**Структура Point3D для хранения координат точки в трехмерном пространстве.

*Поля*

string name – имя точки

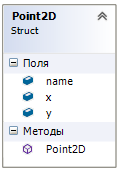
float x – координата точки по X

float y – координата точки по Y

float Z – координата точки по Z

*Методы*

Point3D – конструктор структуры Point3D

 Структура Point2D для хранения координат точки в плоскости экрана.

*Поля*

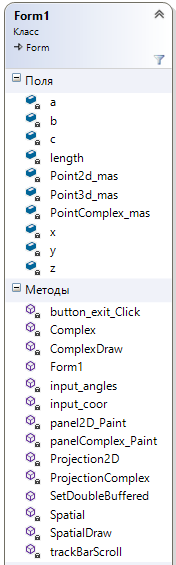
string name – имя точки

float x – координата точки по X

float y – координата точки по Y

*Методы*

Point2D – конструктор структуры Point2D



Класс Form1 – основной класс форма, инициализирует все компоненты формы, обработчик событий.

*Поля*

int a – переменная, в которую записывается значение угла α

int b – переменная, в которую записывается значение угла β

int с – переменная, в которую записывается значение угла γ

int length – переменная, в которой хранится длина осей

Point2D[] Point2d\_mas – массив элементов структуры Point2D, содержащий координаты точек в экранной системе координат для пространственного чертежа

* Point2d\_mas[0] – координаты точки О
* Point2d\_mas[1]-Point2d\_mas[3] – координаты начала отрисовки осей
* Point2d\_mas[4]-Point2d\_mas[6] – координаты конца отрисовки осей
* Point2d\_mas[7] – координаты точки T
* Point2d\_mas[8]-Point2d\_mas[10] – координаты точек TX, TY, TZ
* Point2d\_mas[11]-Point2d\_mas[13] – координаты точек проекции Т1, Т2, Т3

Point3D[] Point3d\_mas – массив элементов структуры Point3D, содержащий координаты точек в трехмерном пространстве

* Point3d\_mas[0] – координаты точки О
* Point3d\_mas[1]- Point3d\_mas[3]– координаты начала отрисовки осей
* Point3d\_mas[4]- Point3d\_mas[6] – координаты конца отрисовки осей
* Point3d\_mas[7] – координаты точки T
* Point3d\_mas[8]- Point3d\_mas[10] – координаты точек TX, TY, TZ
* Point3d\_mas[11]- Point3d\_mas[13] – координаты точек проекции Т1, Т2, Т3

Point2D[] PointComplex\_mas – массив элементов структуры Point2D, содержащий координаты точек в экранной системе координат для комплексного чертежа.

* PointComplex\_mas[0] – координаты точки О
* PointComplex\_mas[1] - PointComplex\_mas[2] – координаты начала отрисовки осей
* PointComplex\_mas[3] - PointComplex\_mas[4] – координаты конца отрисовки осей
* PointComplex\_mas[5] - PointComplex\_mas[8] – координаты точек TX, TY1, TY3, TZ
* PointComplex\_mas[9] - PointComplex\_mas[11] – координаты точек проекции Т1, Т2, Т3

int x –переменная, в которую записывается значение координаты Х

int y –переменная, в которую записывается значение координаты Y

int z –переменная, в которую записывается значение координаты Z

*Методы*

button\_exit\_Click – метод, который срабатывает при нажатие на button\_exit. Закрывает форму и заканчивает работу программы.

Complex – метод, в котором координаты точек для комплексного чертежа переводятся из трехмерных координат в координаты экрана.

ComplexDraw – метод отрисовки комплексного чертежа

Form1 – конструктор формы. Срабатывает при загрузке формы. В нем инициализируется поверхность для рисования, а также вызываются методы пересчета комплексного и пространственного чертежей.

input\_angles – метод, который записывает значения углов из trackBar в соответствующие пременные.

input\_coor – метод, который записывает значения координат из trackBar в соответствующие переменные, а также в нем каждой точке чертежа в трехмерном пространстве присваиваются координаты в зависимости от координат точки Т.

panel2D\_Paint – метод, вызывается при перерисовки пространственного чертежа. В нем вызывается метод SpatialDraw.

panelComplex\_Paint - метод, вызывается при перерисовки комплексного чертежа. В нем вызывается метод ComplexDraw.

Projection2D – метод, в котором координаты точек для пространственного чертежа из трехмерного пространства переводятся в экранные координаты.

ProjectionComplex – метод, в котором координаты точек для комплексного чертежа из трехмерного пространства переводятся в экранные координаты.

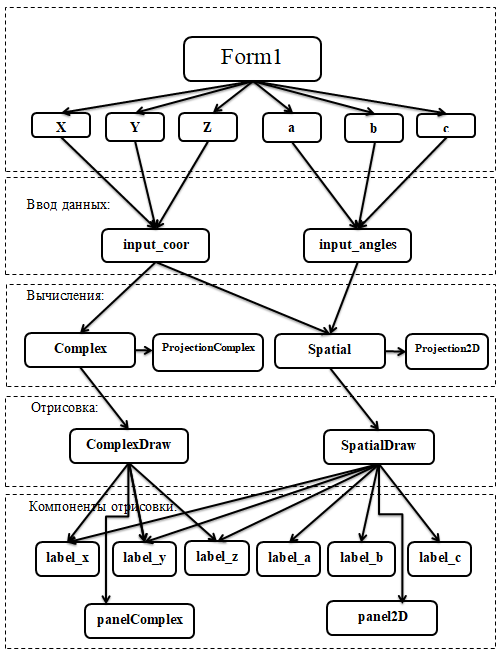
SetDoubleBuffered – метод, который позволяет избавиться от мерцания при перерисовки чертежей.

Spatial – метод, в котором координты точек для пространственного чертеже переводятся из трехмерных координат в координаты экрана

SpatialDraw – метод отрисовки пространственного чертежа

trackBarScroll – метод, который вызывается при передвижение trackBar. В нем распознается, какой trackBar был задействован, в зависимости от этого меняются значения переменных x, y, z, a, b, c.

**Функциональная схема программы**



*Работа программы*

При запуске программы происходит инициализация формы **Form1**. Создаются переменные **x, y, z, a, b, c**.

*Ввод данных*

Переменные x,y,z посылаются в метод **input\_coor**, где считываются значения с trackBar и заносятся в соответствующие переменные, а также заполняется массив Point3D\_mas. Переменные a, b, c посылаются в метод **input\_angles**, где считываются значения с trackBar и заносятся в соответствующие переменные.

*Вычисления*

При пересчете пространственного чертежа вызывается метод **Spatial**. В нем считается точка начала отсчета O по формуле (1), а также в цикле с 1 по 13 элемент массива Point2D\_mas вызывается метод **Projection2D**, в котором точка пространственного чертежа переводится их трехмерных координат в координаты экрана с помощью формулы (2).

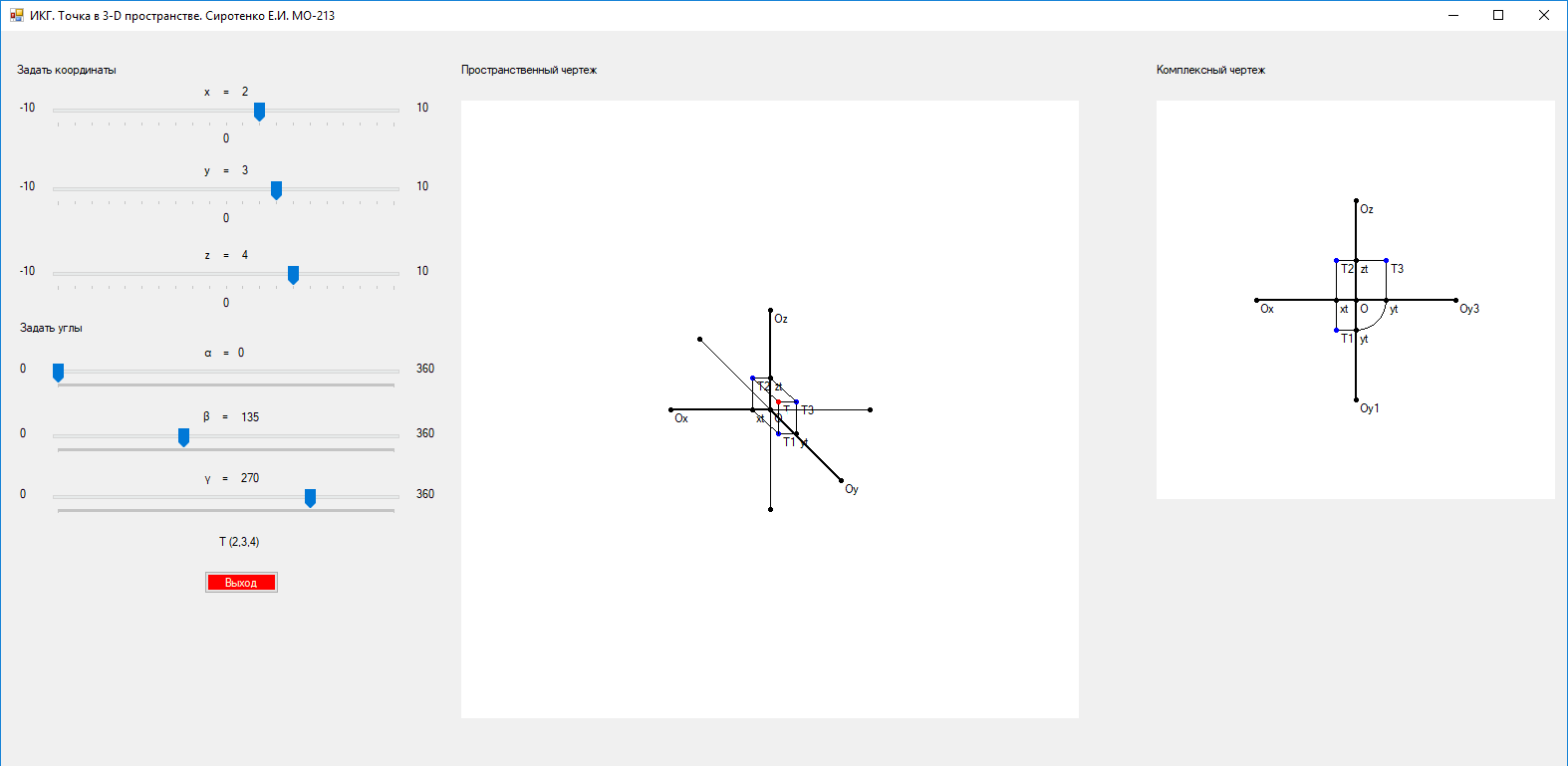
При пересчете комплексного чертежа вызывается метод **Complex**. В нем считается точка начала отсчета O по формуле (1), а также с 1 по 11 элементы массива PointComplex\_mas вызывается метод **ProjectionComplex**, где происходит пересчет координат комплексного чертежа, используя формулы (3), (4), (5).

*Отрисовка*

Отрисовка пространственного чертежа происходит в методе **SpatialDraw**, в котором используя стандартные методы работы с графикой отрисовываются координатные оси, точки проекции точки T и сама точка Т, а также линии, соединяющие точки проекции. Чертеж выводится в **panel2D**, а текущие значения переменных x, y, z, a, b, c в **label\_x, label\_y, label\_z, label\_a, label\_b, label\_c** соответственно.

Отрисовка комплексного чертежа происходит в методе **ComplexDraw**, в нем используются те же методы работы с графикой, что и при отрисовки пространственного чертежа. Чертеж выводится в **panelComplex**, а текущие значения переменных x, y, z, в **label\_x, label\_y, label\_z** соответственно.

**Руководство пользователя**

Запустив программу, пользователь на экране увидит следующее:

На экране 2 чертежа и 6 ползунков. Левый чертеж – пространственный, правый – комплексный. При движении верхних трех ползунков чертежи перерисовываются, а при движении нижних трех ползунков перерисовывается только пространственный чертеж. Над каждым из ползунков можно увидеть их соответствующие значения. Также возле каждого ползункового переключателя расположены наибольшее и наименьшее значения данного переключателя. При нажатии кнопки «Выход» программа закроется.

**Заключение**

Выведены формулы точек в экранной системе координат для построения пространственного и комплексного чертежей. Разработана программа, выводящая на экран пространственный и комплексный чертеж точки, и позволяющая динамически изменять координаты точки и углы координатных осей.