

Практическая работа №1

Определение параметров задания моделируемых тел

Основные определения

Трехмерное тело – область пространства, состоящая из однородного материала и ограниченная замкнутой поверхностью, состоящей из граней. Любое трехмерное тело состоит из базовых трехмерных элементов: граней, ребер и вершин.

Грань – гладкая (не обязательно плоская) часть поверхности детали, ограниченная замкнутым контуром из ребер.

Ребро – пространственная кривая произвольной конфигурации, полученная на пересечении двух граней.

Вершина – точка в трехмерном пространстве. Для твердого тела это может быть точкой на конце ребра или пересечении ребер.

Для задания трехмерных тел с плоскими гранями и прямолинейными ребрами достаточно задать координаты вершин и связи между вершинами.

Если тело имеет плоские грани и криволинейные ребра, для таких ребер должны быть заданы также либо набор функций, описывающих сегменты ребер, либо множество точек в плоскости грани, образующих данное ребро с заданным расстоянием между этими точками. Чем меньше данное расстояние, тем точнее данное множество точек описывает форму данного ребра.

Если тело содержит неплоские грани, для их задания могут быть использованы формообразующие операции на основе плоских эскизов. Данный способ лежит в основе твердотельного моделирования в системах автоматизированного проектирования.

Практическая работа №2

Задание вершин и связей трехмерных тел в программе

Основные определения

Трехмерное тело – область пространства, состоящая из однородного материала и ограниченная замкнутой поверхностью, состоящей из граней. Любое трехмерное тело состоит из базовых трехмерных элементов: граней, ребер и вершин.

Грань – гладкая (не обязательно плоская) часть поверхности детали, ограниченная замкнутым контуром из ребер.

Ребро – пространственная кривая произвольной конфигурации, полученная на пересечении двух граней.

Вершина – точка в трехмерном пространстве. Для твердого тела это может быть точкой на конце ребра или пересечении ребер.

Для задания трехмерных тел с плоскими гранями и прямолинейными ребрами достаточно задать координаты вершин и связи между вершинами.

Если тело имеет плоские грани и криволинейные ребра, для таких ребер должны быть заданы также либо набор функций, описывающих сегменты ребер, либо множество точек в плоскости грани, образующих данное ребро с заданным расстоянием между этими точками. Чем меньше данное расстояние, тем точнее данное множество точек описывает форму данного ребра.

Если тело содержит неплоские грани, для их задания могут быть использованы формообразующие операции на основе плоских эскизов. Данный способ лежит в основе твердотельного моделирования в системах автоматизированного проектирования.

Ввод параметров тела в программу

Для ввода параметров трехмерного тела в программу могут использоваться следующие способы:

1. Задание константных значений в тексте программы. Для этого в программе должны быть объявлены структуры данных типа массив или список, каждому элементу которых присваивается соответствующее значение координат и признаки наличия связей между вершинами в определенном формате.
2. Формирование файла с указанием координат вершин и связей между ними в определенном формате. Для считывания данных из этого файла программа должна реализовывать алгоритмы, позволяющие распознавать отдельные параметры в соответствии с форматом файла и записывать их во внутренние структуры данных типа массив или список (см. пункт 1).
3. Ввод данных посредством пользовательских форм или таблиц. При этом программа должна обеспечивать определенный пользовательский интерфейс, позволяющий вводить все необходимые параметры, и запись данных значений во внутренние структуры данных (см. пункт 1).

Сведения о массивах

Массив представляет собой структуру данных смежного типа. Массив - это непрерывная область памяти фиксированной длины. Все элементы массива имеют одинаковый размер и структуру и располагаются внутри области памяти массива без каких-либо пропусков. Определение местоположения элементов массива в памяти производится по индексу элемента, который может принимать значения от 0 до N-1, где N – количество элементов массива. При этом адрес памяти заданного элемента одномерного массива вычисляется с помощью следующей формулы:

$$\begin{array}{|c|} \hline \text{Адрес памяти} \\ \text{элемента} \\ \text{одномерного} \\ \text{массива} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{Адрес начала} \\ \text{области} \\ \text{памяти} \\ \text{массива} \\ \hline \end{array} + \begin{array}{|c|} \hline \text{Размер} \\ \text{элемента} \\ \text{массива} \\ \hline \end{array} * \begin{array}{|c|} \hline \text{Индекс} \\ \text{элемента} \\ \text{массива} \\ \hline \end{array}$$

В VBA объявление массива производится в соответствии со следующей синтаксической конструкцией:

Dim *varname*[[*(number of items)*]] [**As** *type*],

где:

Dim – ключевое слово, используемое для объявления переменных и структур данных;

Varname – имя массива;

Number of items – количество элементов массива;

As – ключевое слово, используемое для задания типа;

Type – тип элементов массива.

Например:

Dim DayArray(50) as Boolean - объявление одномерного массива *DayArray* из 50-ти элементов логического типа;

Dim Matrix(3, 4) As Integer - объявление двухмерного массива *Matrix*, представляющего собой таблицу элементов целого типа с 3-мя строками и 4-мя столбцами.

Для записи значений в элементы массива используется операция присваивания в соответствие со следующим синтаксисом:

Name (*Index*, [*Index*, ...]) = *value*,

где:

Name – имя массива;

Index – номер элемента массива;

Value – присваиваемое значение.

Например:

Matrix(0,3) = 5 - присваивание значения 5 элементу двухмерного массива, находящемуся на пересечении 1- строки и 4-го столбца (от нуля);

Links(3) = 10 - присваивание значения 10 4-му элементу (от нуля) одномерного массива *Links*.

Практическая работа №3

Отображение трехмерных тел

Отображение трехмерного тела на экране компьютера производится с помощью программы, в которой используются графические функции программного интерфейса языка программирования или операционной системы.

Для отображения простейших тел с плоскими гранями и прямолинейными ребрами необходимы исходные данные в виде координат вершин и связей между вершинами, которые определяют множество ребер тела.

Так как экран представляет собой плоскость, то прежде чем производить отображение трехмерного тела, необходимо задать проекцию данного тела на плоскость экрана. Первоначальная проекция представляет собой вид тела под углом зрения, образующим перпендикуляр к плоскости экрана. Он совпадает по направлению с осью *Z*. Две других координатных оси задают плоскость экрана. Ось *X* определяет положение точек (пикселей) изображения по горизонтали, а ось *Y* - по вертикали. Центр координат находится в верхнем левом углу экрана. Ось *X* направлена вправо, ось *Y* - вниз.

Ребра текущей проекции заданного трехмерного тела представляют собой отрезки, соединяющие вершины тела. Данные отрезки выводятся на экран с помощью графической функции **F_Line**, требующей задания следующих входных параметров:

1. значение координаты *X* в пикселях первого конца отрезка;
2. значение координаты *Y* в пикселях первого конца отрезка;
3. значение координаты *X* в пикселях второго конца отрезка;
4. значение координаты *Y* в пикселях второго конца отрезка;
5. цвет отрезка в виде числового кода, получаемого функцией RGB на основе кодов интенсивности красного, зеленого и синего цветов;
6. толщина линии отрезка в пикселях.

Вызов функции производится только для тех пар вершин, между которыми задана связь.

Например:

```
For i = 0 To LinkedNodes - 1
  If lm(fp(i), sp(i)) Then
    F_Line nx1(fp(i), ny1(fp(i)), nx1(sp(i)),
    ny1(sp(i)), ItemsCol, 2
  End If
Next,
```

где:

LinkedNodes	- переменная, содержащая количество связей между вершинами;
fp, sp	- массивы, содержащие номера вершин каждой связи <i>i</i> ;
lm	- двухмерный массив, представляющий собой таблицу, строки которой соответствуют номерам начальных вершин, а столбцы - номерам конечных вершин; на пересечении строк и столбцов заданы признаки связи между соответствующими начальными и конечными вершинами трехмерного тела;
nx1, ny1	- массивы, содержащие значения координат вершин по оси <i>X</i> , <i>Y</i> ;
ItemsCol	- переменная, содержащая значение цвета.

Таким образом, в приведенном примере происходит следующее:

В цикле **for** перебираются все связи между вершинами. Номер текущей вершины хранится в переменной **i**.

Внутри цикла проверяется условие **lm(fp(i), sp(i))=true**, для проверки связи между вершинами с номерами **fp(i)**, **sp(i)**.

Если условие выполняется, производится вызов функции отображения линии на экране:

```
F_Line nx1(fp(i)), ny1(fp(i)), nx1(sp(i)), ny1(sp(i)),  
ItemsCol, 2,
```

на вход которой подаются требуемые параметры (значения координат, цвет, толщина линии).

В результате на экране появляется контур текущей проекции заданного трехмерного тела (рис. 1).

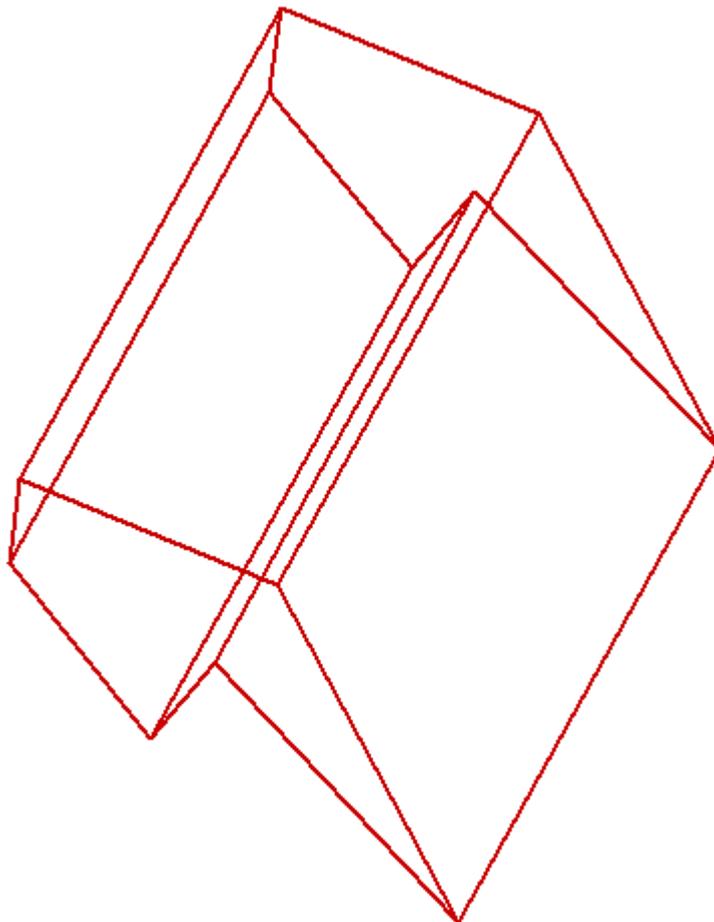


Рис.1 Пример отображения проекции трехмерного тела на экране

Практическая работа №4

Программирование геометрических уравнений на плоскости

Для отображения трехмерных тел на экране компьютера требуется построение проекций тел на плоскость экрана.

Различные проекции трехмерных тел могут быть результатом таких операций, как поворот и масштабирование.

Кроме этого, при отображении необходимо обеспечивать реалистичность получаемых проекций. Это достигается путем удаления с изображения проекции невидимых вершин и ребер.

Расчет новых координат вершин при повороте и масштабировании трехмерных тел и последующий анализ на наличие невидимых вершин и ребер требует применения определенного математического аппарата, включающий уравнения для геометрических преобразований на плоскости и в пространстве.

Уравнения на плоскости представляют собой следующие зависимости:

1. Уравнение прямой на плоскости;
2. Уравнение прямой, перпендикулярной заданной прямой;
3. Уравнение точки пересечения двух прямых на плоскости;
4. Формула определения расстояния между двумя точками.

Данные зависимости должны быть реализованы в программе в виде выражений, функций и подпрограмм, которые должны формировать результат на основе входных параметров.

Формулы зависимостей.

1. Уравнение прямой, проходящей через две точки

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} = y = kx + b, \quad k = \frac{y_2 - y_1}{x_2 - x_1}, \quad b = y_1 - \frac{y_2 - y_1}{x_2 - x_1} \cdot x_1$$

2. Уравнение прямой, перпендикулярной другой прямой в точке (x_1, y_1)

$$y = k_p \cdot x + b_p,$$

где:

$$k_p = -1/k,$$

$$b_p = y_1 - k_p \cdot x_1,$$

k – угловой коэффициент исходной прямой.

Примечание: в программе необходимо учитывать случаи деления на 0 и производить отдельную обработку таких случаев.

3. Координаты точки пересечения двух прямых

$$x = \frac{b_1 - b_2}{k_2 - k_1}, \quad y = \frac{k_2 \cdot b_1 - k_1 \cdot b_2}{k_2 - k_1}, \quad \text{где}$$

$b_1, k_1; b_2, k_2$ – коэффициенты первой и второй прямой соответственно

4. Формула определения расстояния между двумя точками на плоскости

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Средства языка программирования для записи математических выражений

В VBA имеются следующие операторы и функции для реализации приведенных выше зависимостей:

- = - операция присваивания (слева приемник, справа - источник);
- *
- / - деление;
- +
- - вычитание;
- () - круглые скобки;
- ^ - возведение в степень;
- Sqr() - функция вычисления квадратного корня

Например, выражение для расчета расстояния между двумя вершинами на языке VBA выглядит так:

$$d = \text{Sqr}((\text{nx1}(0) - \text{nx1}(1))^2 + (\text{ny1}(0) - \text{ny1}(1))^2),$$

где:

d – переменная вещественного типа, в которую записывается результат;

nx1(0), ny1(0), nx1(1), ny1(1) – элементы массивов координат для первой и второй вершин.

Практическая работа №5

Программирование пространственных геометрических уравнений

Для отображения трехмерных тел на экране компьютера требуется построение проекций тел на плоскость экрана.

Различные проекции трехмерных тел могут быть результатом таких операций, как поворот и масштабирование.

Кроме этого, при отображении необходимо обеспечивать реалистичность получаемых проекций. Это достигается путем удаления с изображения проекции невидимых вершин и ребер.

Расчет новых координат вершин при повороте и масштабировании трехмерных тел и последующий анализ на наличие невидимых вершин и ребер требует применения определенного математического аппарата, включающий уравнения для геометрических преобразований на плоскости и в пространстве.

Уравнения в пространстве представляют собой следующие зависимости:

1. Уравнение плоскости в пространстве, проходящей через три точки;
2. Параметрическое уравнение прямой в пространстве;
3. Уравнение точки пересечения прямой и плоскости в пространстве;
4. Формула определения расстояния между двумя точками в пространстве.

Данные зависимости должны быть реализованы в программе в виде выражений, функций и подпрограмм, которые должны формировать результат на основе входных параметров.

Формулы зависимостей.

5. Уравнение плоскости в пространстве

$$Ax + By + Cz + D = 0$$

6. Уравнение плоскости в пространстве, проходящей через три точки

$$\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix} = 0$$

$$A = (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1)$$

$$B = (z_2 - z_1)(x_3 - x_1) - (z_3 - z_1)(x_2 - x_1)$$

$$C = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)$$

$$D = -(Ax_1 + By_1 + Cz_1)$$

7. Параметрическое уравнение прямой в пространстве

$$\left. \begin{aligned} x &= \alpha t + x_0 \\ y &= \beta t + y_0 \\ z &= \gamma t + z_0 \end{aligned} \right\}$$

, где α, β, γ – координаты коллинеарного вектора, проведенного из начала координат.

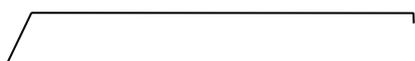
8. Координаты точки пересечения прямой и плоскости в пространстве

$$\left. \begin{aligned} x &= \alpha t + x_0 \\ y &= \beta t + y_0 \\ z &= \gamma t + z_0 \\ Ax + By + Cz + D &= 0 \end{aligned} \right\} \Rightarrow$$

$$\Rightarrow t = - \frac{Ax_0 + By_0 + Cz_0 + D}{A\alpha + B\beta + C\gamma}$$

Примечание: в программе необходимо учитывать случаи деления на 0 и производить отдельную обработку таких случаев.

9. Формула определения расстояния между двумя точками в пространстве



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Средства языка программирования для записи математических выражений

В VBA имеются следующие операторы и функции для реализации приведенных выше зависимостей:

- = - операция присваивания (слева приемник, справа - источник);
- * - умножение;
- / - деление;
- + - сложение;
- - вычитание;
- () - круглые скобки;
- ^ - возведение в степень;
- Sqr() - функция вычисления квадратного корня

Например, выражение для расчета расстояния между двумя вершинами на языке VBA выглядит так:

$$d = \text{Sqr}((\text{nx1}(0) - \text{nx1}(1)) ^ 2 + (\text{ny1}(0) - \text{ny1}(1)) ^ 2) + (\text{nz1}(0) - \text{nz1}(1)) ^ 2),$$

где:

d – переменная вещественного типа, в которую записывается результат;

nx1(0), ny1(0), nx1(1), ny1(1), nz1(0), nz1(1) – элементы массивов координат для первой и второй вершин в пространстве.

Практическая работа №6

Создание проекций трехмерных тел

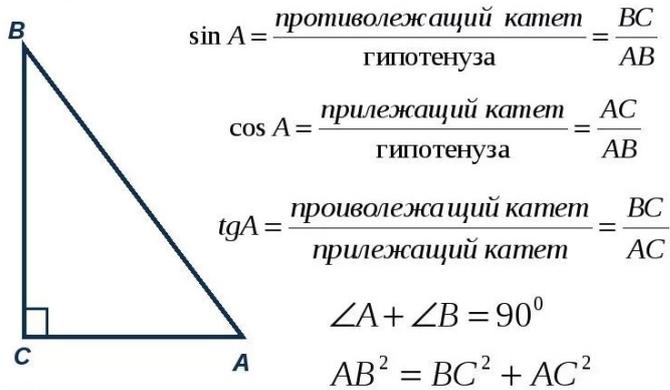
В результате операций поворота и масштабирования формируются различные проекции трехмерных тел, которые отображаются на экране.

Поворот тел

Поворот трехмерного тела осуществляется путем задания углов поворота вокруг каждой из координатных осей относительно заданного центра.

Для расчета новых координат вершин после поворота используются уравнения для решения прямоугольных треугольников.

Уравнения для решения прямоугольных треугольников

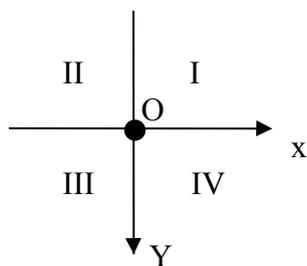


При этом угол поворота рассчитывается в плоскости, перпендикулярной заданной оси поворота. Т.е., если поворот тела осуществляется вокруг оси X, новые координаты рассчитываются в плоскости ZY; если вокруг оси Y – в плоскости ZX; если вокруг оси Z – в плоскости XY. При этом для последующих вычислений горизонтальная ось принимается как ось X, а вертикальная – как ось Y.

Центр поворота O задается путем определения трехмерных координат соответствующей точки. При задании оси поворота центр поворота O и вершины P тела проецируются на соответствующую плоскость.

Угол между радиусом, соединяющим центр поворота O и проекцию вершины P, и осью X вычисляется как $\arctg(P_x/P_y)$

Положительные значения углов отсчитываются при повороте против часовой стрелки вокруг заданного центра относительно горизонтальной оси. Поэтому при определении реального текущего угла α , который образует заданная вершина P относительно центра поворота O и оси X, необходимо учитывать квадрант, в котором находится точка вершины. При этом возможны следующие варианты:



1. $P_x=O_x$ и $P_y<O_y$, $\alpha=90^\circ$;
2. $P_x=O_x$ и $P_y>O_y$, $\alpha=270^\circ$;
3. $P_x<O_x$ и $P_y=O_y$, $\alpha=180^\circ$;
4. $P_x>O_x$ и $P_y=O_y$, $\alpha=0^\circ$;
5. $P_x>O_x$ и $P_y>O_y$ (IV квадрант), $\alpha=360^\circ - \alpha$;
6. $P_x>O_x$ и $P_y<O_y$ (I квадрант), $\alpha=\alpha$;
7. $P_x<O_x$ и $P_y<O_y$ (II квадрант), $\alpha=180^\circ - \alpha$;
8. $P_x<O_x$ и $P_y>O_y$ (III квадрант), $\alpha=270^\circ - \alpha$;

Масштабирование тел

Масштабирование трехмерного тела – это перерасчет координат всех вершин тела с учетом коэффициента масштабирования относительно центра масштабирования.

Порядок действий при масштабировании следующий:

1. Рассчитываются расстояния от центра до каждой вершины в пространстве.
2. Данные расстояния умножаются на коэффициент масштабирования.
3. Вычисляются новые координаты по теореме Пифагора на основе измененных расстояний.

В результате, если коэффициент масштабирования больше 1, происходит пропорциональное увеличение размеров тела, если от 0 до 1 – уменьшение.

После операций поворота и масштабирования полученная проекция трехмерного тела отображается на экране с помощью макросов, разработанных в предыдущих практических работах.

Практическая работа №7

Исследование алгоритмов определения граней и принадлежности точек граням

Для отображения трехмерных тел на экране компьютера требуется построение проекций тел на плоскость экрана.

Различные проекции трехмерных тел могут быть результатом таких операций, как поворот и масштабирование.

При отображении необходимо обеспечивать реалистичность получаемых проекций. Это достигается путем удаления с изображения проекции невидимых вершин и ребер. Отправной точкой для решения данной задачи является задача определения множества вершин, образующих грани заданного трехмерного тела. Определение граней производится на этапе задания трехмерного тела. Далее для каждой новой проекции тела определяются невидимые вершины и невидимые ребра либо сегменты ребер, которые исключаются при отображении проекции на экран.

Алгоритм определения граней

Алгоритм определения граней заключается в поиске минимальных замкнутых последовательностей вершин в соответствии с заданными связями между ними. Последовательность вершин называется замкнутой, если конечная вершина совпадает с начальной, причем номера вершин в пределах одной последовательности повторяться не могут. При этом проверяется принадлежность искомым вершин ранее определенным граням. Допускается наличие не более одного смежного ребра с ранее определенными гранями.

Выходные данные

В результате выполнения алгоритма формируется массив списков номеров вершин для каждой определенной грани заданного тела.

Алгоритм определения принадлежности точек граням

Алгоритм основывается на подсчете точек пересечения произвольного луча, проведенного из заданной точки, пересекающего заданный контур, с сегментами данного контура. При этом, если количество точек пересечения четное, данная точка не принадлежит данному контуру, если нечетное, то принадлежит. Алгоритм должен учитывать особые случаи, когда луч проходит через вершины контура. При этом вершина контура, через который проходит луч, считается точкой пересечения, если смежные сегменты контура в данной вершине располагаются по разные стороны луча.

Выходные данные

Результатом выполнения алгоритма является значение “Истина” или “Ложь”. Первое означает, что заданная точка лежит внутри заданного контура.

Практическая работа №8

Определение невидимых вершин проекции трехмерного тела

Для отображения трехмерных тел на экране компьютера требуется построение проекций тел на плоскость экрана.

Различные проекции трехмерных тел могут быть результатом таких операций, как поворот и масштабирование.

При отображении необходимо обеспечивать реалистичность получаемых проекций. Это достигается путем удаления с изображения проекции невидимых вершин и ребер.

Алгоритм определения невидимых вершин заданной проекции заданного тела

Требуется разработать алгоритм, определяющий множество вершин заданного трехмерного тела, которые не видны на заданной проекции данного тела в плоскости экрана.

Алгоритм основывается на анализе расположения вершин относительно граней трехмерного тела на заданной проекции данного тела. Для этого перебираются все грани и для каждой грани перебираются все вершины. Множество граней тела и принадлежность вершин контурам граней является результатом выполнения соответствующих алгоритмов, рассмотренных в предыдущей практической работе. Каждая грань представляет собой множество вершин, которые ее образуют.

Для определения положения текущей вершины относительно текущей грани на заданной проекции необходимо выполнить следующие действия:

1. Для любых трех вершин текущей грани рассчитать коэффициенты уравнения плоскости грани в пространстве, проходящей через данные вершины.
2. Вычислить точку пересечения перпендикуляра к плоскости экрана, восстановленного в текущей вершине, с плоскостью грани.
3. Определить принадлежность текущей вершины контуру текущей грани на заданной проекции тела.
4. Если текущая вершина находится в пределах контура проекции текущей грани, сравнить координаты по оси Z точки пересечения и текущей вершины. Если значение координаты текущей вершины меньше соответствующей координаты точки пересечения, данная вершина является невидимой, т.к. находится за текущей гранью на данной проекции.
5. Выполнить данные действия для всех граней и всех вершин.

Выходные данные

Результатом выполнения алгоритма является массив невидимых вершин для текущей проекции заданного трехмерного тела.

Практическая работа №9

Определение невидимых граней проекции трехмерного тела

Для отображения трехмерных тел на экране компьютера требуется построение проекций тел на плоскость экрана.

Различные проекции трехмерных тел могут быть результатом таких операций, как поворот и масштабирование.

При отображении необходимо обеспечивать реалистичность получаемых проекций. Это достигается путем удаления с изображения проекции невидимых вершин и ребер.

Алгоритм определения невидимых граней заданной проекции заданного тела

Требуется разработать алгоритм, определяющий множество ребер заданного трехмерного тела, которые не видны на заданной проекции данного тела в плоскости экрана, либо множество невидимых сегментов ребер, которые видны на данной проекции частично.

Алгоритм основывается на анализе видимости вершин и срединных точек сегментов ребер. Если обе вершины какого-либо ребра видимы, ребро также видимо. Если обе вершины невидимы, данное ребро также невидимо. Если одна вершина видима, а другая нет, определяются точки пересечения данного ребра с другими ребрами. Для каждого сегмента текущего ребра между точками пересечения определяются срединные точки. Для каждой срединной точки проверяется ее видимость. Если данная срединная точка видима, то и сегмент, которому она принадлежит, так же видим. Если срединная точка невидима, то и соответствующий сегмент тоже невидим.

Выходные данные

Результатом выполнения алгоритма является массив координат начальных и конечных точек видимых сегментов.

Определение видимости конкретных точек на заданной проекции производится в соответствии с алгоритмами, полученными в предыдущих практических работах.

Для определения невидимых ребер или их сегментов необходимо выполнить следующие операции:

1. Для всех связей между вершинами определить видимость связанных вершин на заданной проекции и отобразить связи, в которых одна вершина видима, а другая нет.
2. Для каждого ребра, соответствующего отображенным в шаге 1 связям найти множество точек пересечений с другими ребрами.
3. Для каждого сегмента между точками пересечения определить срединную точку.
4. Для каждой срединной точки установить ее видимость.
5. Сохранить координаты начала и конца тех сегментов, срединные точки которых являются видимыми.
6. Отобразить на экране видимые ребра и видимые сегменты частично невидимых ребер.

Практическая работа №10

Создание классов трехмерных тел в VBA

Трехмерные тела представляют собой геометрические объекты, для которых могут создаваться компьютерные модели с использованием средств объектно-ориентированного программирования (ООП).

ООП основывается на трех принципах:

1. Инкапсуляция - это использование объединения данных и инструкций по их обработке в единую сущность - класс. Во время написания программ на одном из языков ООП происходит разграничение между информацией внутри сущности и снаружи. Таким образом достигается обеспечение безопасности данных и методов их реализации от внешних воздействий, например, со стороны других классов, не относящихся к этому объекту. Внутри сущности данные успешно взаимодействуют друг с другом, но надежно защищены от несанкционированного доступа извне.
2. Наследование - это возможность дочернего класса использовать свойства, методы и события родительского класса без повторного их описания и реализации.
3. Полиморфизм – это возможность однообразного вызова одних и тех же методов разных объектов, в каждом из которых вызываемые методы реализуют различные функции.

ООП базируется на использовании классов.

Класс – это сложный тип данных, в котором описываются свойства, методы и события объектов, создаваемых в качестве экземпляров данного класса. Класс может рассматриваться как составной пользовательский тип данных.

Объект – это экземпляр какого-либо класса. Объекты объявляются в программе также как переменные, но в качестве типа данных указывается имя класса. При этом в оперативной памяти резервируется область для всех свойств, методов и событий, описанных в соответствующем классе. От одного класса может быть произведено сколько угодно объектов, и все они будут иметь отдельные области памяти для хранения их содержимого.

Класс имеет следующую структуру:

Поле – элемент класса для хранения данных, представляющий собой обычную переменную элементарного или составного типа, объявленную внутри класса.

Свойство – элемент класса, позволяющий производить считывание или запись полей с заданием какой-либо предварительной обработки. При этом сами поля могут быть недоступными для прямого доступа извне.

Метод – процедура или функция, созданные внутри класса и являющиеся доступными извне.

Событие – сигнал, генерируемый объектом в зависимости от каких-либо условий, который перехватывается в каком-либо модуле программы, в котором был создан данный объект. Таким образом, программа способна реагировать на события, генерируемые объектами, и производить немедленную обработку таких событий.

В VBA имеются следующие инструменты создания классов:

1. Создание модулей типа Класс (*Главное меню\Insert\Class Module*). После создания модуля класса необходимо задать имя класса в окне свойств модуля. Это имя будет использоваться в программе при объявлении объектов данного класса.
2. Создание полей производится так же как создание переменных в программе. При этом, если нужно задать полный доступ к данному полю извне, необходимо использовать ключевое слово **Public**.
3. Создание свойства. Свойства могут быть типа “Чтение” и “Запись”. Для этого в VBA используются следующие конструкции:

Свойство типа “Чтение”:

```
Public Property Get <Имя свойства> ([<Параметры>]) As <Тип>
    .....
    операции
    .....
    <Имя свойства>=<Поле или выражение>
End Property
```

Свойство типа “Запись”:

```
Public Property Let <Имя свойства> ([<Параметры>])
    .....
    операции
    .....
    <Имя поля>=<переменная или выражение>
End Property
```

4. Создание события. Для этого необходимо сделать следующее:

1) в начале класса поместить объявление события с помощью конструкции:

```
Public Event <Имя события> ([<Параметры>])
```

2) внутри метода, где должен генерироваться сигнал события, вставить команду возбуждения события:

```
RaiseEvent <Имя события> ([<Параметры>])
```

В списке параметров указываются те поля или переменные, значения которых нужно передать наружу вместе с сигналом события.

Практическая работа №11

Создание объектов трехмерных тел в VBA

Трехмерные тела представляют собой геометрические объекты, для которых могут создаваться компьютерные модели с использованием средств объектно-ориентированного программирования (ООП).

ООП базируется на использовании классов.

Класс – это сложный тип данных, в котором описываются свойства, методы и события объектов, создаваемых в качестве экземпляров данного класса. Класс может рассматриваться как составной пользовательский тип данных.

Объект – это экземпляр какого-либо класса. Объекты объявляются в программе также как переменные, но в качестве типа данных указывается имя класса. При этом в оперативной памяти резервируется область для всех свойств, методов и событий, описанных в соответствующем классе. От одного класса может быть произведено сколько угодно объектов, и все они будут иметь отдельные области памяти для хранения их содержимого.

Класс имеет следующую структуру:

Поле – элемент класса для хранения данных, представляющий собой обычную переменную элементарного или составного типа, объявленную внутри класса.

Свойство – элемент класса, позволяющий производить считывание или запись полей с заданием какой-либо предварительной обработки. При этом сами поля могут быть недоступными для прямого доступа извне.

Метод – процедура или функция, созданные внутри класса и являющиеся доступными извне.

Событие – сигнал, генерируемый объектом в зависимости от каких-либо условий, который перехватывается в каком-либо модуле программы, в котором был создан данный объект. Таким образом, программа способна реагировать на события, генерируемые объектами, и производить немедленную обработку таких событий.

В VBA имеются следующие инструменты создания и использования объектов:

5. Объявление экземпляра класса (объекта). Для этого используется тот же синтаксис, что и для объявления переменных:

```
Dim <Имя экземпляра> As <Имя класса>
```

Если предполагается обработка событий данного объекта, объявление выглядит так:

```
Dim WithEvents <Имя экземпляра> As <Имя класса>
```

В памяти резервируется область указателя на объект, в котором будет храниться адрес объекта.

6. Создание экземпляра класса производится с помощью следующей конструкции:

```
Set <Имя экземпляра> = New <Имя класса>
```

В памяти выделяется область для хранения содержимого объекта, а адрес области записывается в указатель (см. 1). При создании объекта автоматически выполняется метод объекта `Class_Initialize`, называемый конструктором.

7. Создание массива экземпляров класса. Для этого необходимо объявить массив типа `Object`:

```
Dim <Имя массива> (<Количество элементов>) As Object,
```

А затем инициализировать элементы массива нужными объектами:

```
Set <Имя массива> (0) = New <Имя класса>  
Set <Имя массива> (1) = New <Имя класса>  
.....
```

8. Удаление объекта выполняется с помощью следующей команды:

```
Set <Имя объекта> = Nothing
```

9. Обращение к полям объекта производится с помощью следующей синтаксической конструкции:

```
<Имя объекта>.<Имя поля>
```

Поле должно быть объявлено внутри класса как Public.

10. Обращение к свойствам происходит так:

```
<Имя объекта>.<Имя свойства> (<Значения параметров>)
```

11. Вызов методов:

```
<Имя объекта>.<Имя метода> (<Значения параметров>)
```

Методы должны быть объявлены внутри класса как Public.

12. Обработка событий.

Создается подпрограмма обработки события посредством следующего синтаксиса:

```
Sub <Имя объекта>_<Имя события> (<Значения параметров>)  
.....  
    Операции  
.....  
End sub
```

Обработка событий возможна, если объект был объявлен с ключевым словом WithEvents.

Практическая работа №12

Задание пространственных плоскостей для построения эскизов

Теоретическая часть.

Эскиз – это двухмерное изображение, размещенное на плоскости в трехмерном пространстве.

Эскизы являются основой формообразующих операций, которые применяются для построения твердых тел.

Для создания эскиза в КОМПАС-3D необходимо задать плоскость в пространстве. По умолчанию в редакторе модели имеется три стандартных ортогональных проекционных плоскости: XY, ZX, ZY. Однако данные плоскости являются только вертикальными или

горизонтальными. В процессе построения твердых тел и вырезов часто необходимо задавать произвольные плоскости в пространстве.

В КОМПАС-3D имеются команды для задания рабочих плоскостей. Они находятся в разделе главного меню Операции-Плоскость. Данное подменю предназначено для создания вспомогательных объектов при построении 3D-модели. Входящие в него команды предоставляют более десятка различных способов построения вспомогательных плоскостей:



Смещенная плоскость – предназначена для создания вспомогательной плоскости, смещенной от указанной плоскости или плоской грани на определенное расстояние. Для построения такой плоскости необходимо сначала указать базовую плоскость или грань, после чего задать величину и направление смещения. Величину и направление смещения можно указать на панели свойств или с помощью перетаскивания характерной точки.



Плоскость через три вершины – строит плоскость по трем указанным в модели вершинам. Вершинами могут быть как концы ребер (вершины тела модели), так и трехмерные точки в пространстве.



Плоскость под углом к другой плоскости – также часто употребляемая команда. Она позволяет строить плоскость, проходящую через прямолинейное ребро под заданным углом к базовой (указанной пользователем) плоскости.



Плоскость через ребро и вершину – плоскость строится подобно выполненной по трем вершинам, только вместо двух вершин указывается прямолинейное ребро.



Плоскость через вершину параллельно другой плоскости – плоскость строится через любую указанную в пространстве модели точку (трехмерную точку, вершину) и параллельно любой другой плоскости либо плоской грани.



Плоскость через вершину перпендикулярно ребру – плоскость создается перпендикулярно прямолинейному ребру (или оси). Для ее фиксации вдоль ребра необходимо указать произвольную точку, не лежащую на ребре. Эта точка будет принадлежать создаваемой плоскости и тем самым определит ее точное размещение в пространстве.



Нормальная плоскость – создает одну или несколько плоскостей, нормальных к цилиндрической или конической поверхности детали.



Касательная плоскость – плоскость строится касательно к указанной цилиндрической или конической поверхности. Для точного позиционирования вспомогательной плоскости необходимо также задать плоскую грань или плоскость, нормальную к цилиндрической или конической поверхности (то есть проходящую через ее ось).



Плоскость через ребро параллельно/перпендикулярно другому ребру – формирует вспомогательную плоскость, проходящую через первое указанное в модели ребро параллельно или перпендикулярно другому ребру. На панели свойств с помощью

переключателя Положение плоскости можно задать, параллельно или перпендикулярно будет проходить плоскость. Данная вспомогательная плоскость используется редко.



Плоскость через ребро параллельно/перпендикулярно грани – действие команды аналогично предыдущей, только плоскость размещается параллельно или перпендикулярно не ребру, а выделенной грани.



Средняя плоскость – позволяет построить вспомогательную плоскость-биссектрису двугранного угла и иногда бывает очень полезной (рис. 3.33). Для построения такой плоскости достаточно указать две плоские грани или плоскости. Если заданные грани непараллельны, то построенная плоскость пройдет через линию их пересечения и будет размещена под одинаковым углом к каждой из них (бисекторная плоскость). В противном случае построенная плоскость будет точно посередине между двумя параллельными гранями или плоскостями.

Практическая работа №13

Создание эскизов в КОМПАС-3D

Теоретическая часть.

Эскиз – это двухмерное изображение, размещенное на плоскости в трехмерном пространстве.

Эскизы являются основой формообразующих операций, которые применяются для построения твердых тел.

Требования к эскизам

Как правило, эскиз представляет собой сечение объемного элемента. Реже эскиз является траекторией перемещения другого эскиза - сечения. Для создания объемного элемента подходит не любое изображение в эскизе, оно должно подчиняться некоторым правилам.

- Контуры в эскизе не пересекаются и не имеют общих точек.
- Контуры в эскизе не имеют самопересечений.
- Контур в эскизе изображается стилем линии "Основная".

Замечание. Иногда для построения контура в эскизе (особенно параметрическом) требуются вспомогательные объекты, не входящие в контур. Их можно изображать другими стилями линий; такие объекты не будут учитываться при выполнении операций.

- Эскиз, как и КОМПАС-фрагмент, может содержать несколько слоев. При выполнении операции учитываются объекты во всех слоях, кроме погашенных.

Существуют дополнительные (частные) требования, предъявляемые к эскизам, предназначенным для выполнения конкретных операций:

1. для операций выдавливания:

- в эскизе может быть один или несколько контуров;
- если контур один, то он может быть разомкнутым или замкнутым;

- если контуров несколько, все они должны быть замкнуты;
- если контуров несколько, один из них должен быть наружным, а другие – вложенными в него;
- допускается один уровень вложенности контуров

2. для операций вращения:

- ось вращения должна быть изображена в эскизе отрезком со стилем линии "Осевая".
- ось вращения должна быть одна.
- в эскизе может быть один или несколько контуров.
- если контур один, то он может быть разомкнутым или замкнутым.
- если контуров несколько, все они должны быть замкнуты.
- если контуров несколько, один из них должен быть наружным, а другие – вложенными в него.
- допускается один уровень вложенности контуров.
- ни один из контуров не должен пересекать ось вращения (отрезок со стилем линии "Осевая" или его продолжение).

3. для кинематических операций:

- в эскизе-сечении может быть только один контур, который может быть разомкнутым или замкнутым.
- в эскизе-траектории может быть только один контур, он может быть разомкнутым или замкнутым, если контур разомкнут, его начало должно лежать в плоскости эскиза-сечения, если контур замкнут, он должен пересекать плоскость эскиза-сечения.

4. для операций по сечениям:

для эскиза-сечения:

- эскизы могут быть расположены в произвольно ориентированных плоскостях.
- эскиз начального (конечного) сечения может содержать контур или точку.
- эскиз промежуточного сечения может содержать только контур.
- контур в эскизе может быть только один.
- контуры в эскизах должны быть или все замкнуты, или все разомкнуты.

для эскиза - осевой линии:

- в эскизе может быть только один контур.
- контур может быть разомкнутым или замкнутым.
- контур должен пересекать плоскости всех эскизов.
- эскиз должен лежать в плоскости, не параллельной плоскостям эскизов сечений.

Для корректного формирования элемента по сечениям рекомендуется начинать построение осевой линии в плоскости первого сечения, а заканчивать - в плоскости последнего.

Практическая работа №14

Применение формообразующих операций

Теоретическая часть.

КОМПАС-3D – это система твердотельного моделирования. Это значит, что все ее операции по созданию и редактированию трехмерных моделей предназначены только для работы с твердыми телами.

Общие понятия

Твердое тело – область трехмерного пространства, состоящая из однородного материала и ограниченная замкнутой поверхностью, которая сформирована из одной или нескольких стыкующихся граней. Любое твердое тело состоит из базовых трехмерных элементов: граней, ребер и вершин.

Грань – гладкая (не обязательно плоская) часть поверхности детали, ограниченная замкнутым контуром из ребер.

Ребро – пространственная кривая произвольной конфигурации, полученная на пересечении двух граней.

Вершина – точка в трехмерном пространстве. Для твердого тела это может быть точкой на конце ребра или пересечении ребер.

Любое изменение формы детали (твердого тела) называется *трехмерной формообразующей операцией*, или просто *операцией*.

Формировать твердотельные модели в КОМПАС-3D можно в двух типах документов: КОМПАС-Деталь и КОМПАС-Сборка.

Документ Деталь предназначен для создания с помощью формообразующих операций и хранения модели целостного объекта (чаще всего какого-либо простого изделия, отдельной детали, компонента).

В документе Сборка собираются в единый агрегат смоделированные и сохраненные ранее детали.

Формообразующие операции

Эскиз – это обычное двухмерное изображение, размещенное на плоскости в трехмерном пространстве.

Существует четыре основных подхода к формированию трехмерных формообразующих элементов в твердотельном моделировании:

- *Выдавливание*. Форма трехмерного элемента образуется путем смещения эскиза операции строго по нормали к его плоскости.
- *Вращение*. Формообразующий элемент является результатом вращения эскиза в пространстве вокруг произвольной оси.
- *Кинематическая операция*. Поверхность элемента формируется в результате перемещения эскиза операции вдоль произвольной трехмерной кривой.
- *Операция по сечениям*. Трехмерный элемент создается по нескольким сечениям-эскизам. Эскизов может быть сколько угодно, и они могут быть размещены в произвольно ориентированных плоскостях.

В случае многотельного моделирования применяются также *булевы операции*, которые предназначены для объединения, вычитания или пересечения указанных тел. Они доступны, только если в детали присутствует более одного тела.

При этом возможны следующие варианты (режимы) построения:

- **Добавление материала:**

- новое тело – добавляемый трехмерный элемент формирует в детали новое твердое тело, независимо от того, пересекается он с уже существующими телами или нет.

- объединение – добавляемый элемент соединяется с твердым телом, с которым он пересекается;

- **Удаление материала:**

- вычитание элемента – удаление материала детали происходит внутри замкнутой поверхности, сформированной по заданному эскизу и типу операции (выдавливание, вращение и т. д.);

- пересечение элементов – удаление материала детали, находящегося снаружи поверхности, которая сформирована в результате операции;

Практическая работа №15

Инструменты редактирования твердых тел

Теоретическая часть.

Редактирование элементов модели в КОМПАС-3D может осуществляться следующими способами:

- с помощью операций контекстного меню дерева модели;
- с помощью изменения свойств объектов в панели свойств;
- с помощью задания дополнительных элементов;
- с помощью применения специальных команд.

Команды дерева модели

В контекстном меню дерева модели имеются следующие команды, позволяющие управлять составом операций текущей модели:

1. Скрыть (Показать) – управляет отображением элемента детали, выбранного в дереве построения. После ее выполнения элемент будет скрыт (спрятан) в модели.

2. Отношения в дополнительном окне – команда позволяет создать дополнительное окно дерева модели и отобразить в нем объекты, являющиеся исходными и производными для объекта, выделенного в дереве.

3. Исключить из расчета (включить в расчет) – исключает из расчета выбранную операцию, вследствие чего модель перестраивается так, как будто исключенной операции вообще нет в модели.

4. Исключить из расчета последующие (Включить в расчет последующие) – исключает из расчета детали все трехмерные формообразующие элементы, которые следуют за выделенным элементом (для которого было вызвано контекстное меню).

5. Изменить плоскость – позволяет переназначить плоскость эскиза

Специальные команды

1. Сечение поверхностью  и Сечение по эскизу . При выполнении сечения поверхностью указывается любая поверхность в модели (грань, вспомогательную плоскость) и направление операции (прямое или обратное). Поверхность не обязательно должна быть плоской. Направление в этой операции указывает, какую часть модели вырезать, другими словами, по какую сторону от указанной поверхности рассекать модель. Сечение по эскизу применяется, когда необходимо сформировать разрез более сложного профиля.

2. Массив по сетке  – размещает копируемые элементы в узлах двухмерной сетки, количество копий по каждому из направлений задается отдельно.

3. Массив по концентрической сетке  – копии выбранных трехмерных элементов располагаются равномерно по концентрическим окружностям;

4. Массив вдоль кривой  – создает одномерный массив трехмерных элементов, которые размещаются вдоль произвольной кривой. Копии трехмерного элемента должны быть приклеены (или вырезаны) к тому телу, которому принадлежит исходный элемент.

5. Зеркальный массив  служит для создания зеркального отражения выбранных элементов модели относительно плоскости или плоской грани.

6. Зеркально отразить тело  позволяет получить как одно целое тело, симметрично отразив созданную его часть относительно грани или плоскости, так и два отдельных, симметричных друг другу относительно выбранной плоскости.

ЛИТЕРАТУРА

1. Овечкин Г., Овечкин П., Компьютерное моделирование. Учебник, "Академия", 2017
2. Кобелев, Н.Б. Имитационное моделирование: учебное пособие, М.: КУРС: Инфра-М, 2013
3. Баранова И. Компас 3D для школьников: черчение и компьютерная графика, М: ДМК Пресс, 2009
4. Кудрявцев Е. М. КОМПАС-3D V10, Максимально полное руководство, М:ДМК Пресс, 2008
5. Кидрук М. И., КОМПАС-3D V10 на 100 %, электронное учебное пособие.
6. С.А. Орлов, Теория и практика языков программирования: учебник для вузов. Стандарт 3-го поколения - СПб:Питер, 2013 - 688 стр.
7. Джон Уокенбах, Профессиональное программирование на VBA Excel 2010 — М.: «Диалектика», 2011. — С. 944
8. Колдаев В.Д., Основы алгоритмизации и программирования: учебное пособие – М: ИД Форум: НИЦ Инфра-М, 2012 - 416 стр.
9. Справочная система КОМПАС 3D V12 LT
10. Справочная система MS Excel 2007