

УДК 004.42

Соколов И.В.

студент магистратуры

2 курс, факультет информатики

Самарский национальный исследовательский университет имени

академика С.П. Королева

Россия, г. Самара

Sokolov IV

graduate student

2 year, Faculty of Information Technology

Samara National Research University

Russia, Samara

МЕТОДЫ ЧТЕНИЯ ВЕКТОРНОЙ ГРАФИКИ НА ПРИМЕРЕ DXF ФОРМАТА.

READING METHODS OF VECTOR GRAPHICS ON THE EXAMPLE OF DXF FORMAT.

Аннотация:

Статья посвящена актуальной на сегодняшний день задаче отображения инженерной графической информации. Рассматриваются особенности реализации графического формата DXF.

Ключевые слова: векторная графика, DXF, структура данных, формат данных, код группы.

Abstract:

The article is devoted to the current task of displaying engineering graphic information. The features of the implementation of the graphic format DXF are considered.

Keywords: *vector graphics, DXF, data structure, data format, group code.*

Как правило, любая инженерная графика (схемы, планы и т.п.)

представляется в векторном формате. Основным преимуществами использования векторной графики являются [1]:

- полная и эффективная возможность редактирования как объекта в целом, так и его частей;
- масштабируемость без потери качества и изменения размера файла;
- небольшой размер файла;
- качество не зависит от операций редактирования;
- размеры обычно указаны в аппаратно-независимых единицах;
- редактируемый текст с произвольным размещением.

Ряд работ [2], [3] посвящён поиску нестандартных решений для представления компьютерной информации.

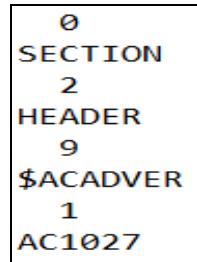
Однако найти недорогой редактор обработки векторной графики, предоставляющий широкие возможности кастомизации и пользовательской настройкой отдельных элементов графики на сегодняшний день не представляется возможным. В связи с этим, часто возникает задача создания своего собственного обработчика векторной графики, способного работать с один из существующих форматов векторной графики.

Оптимальным выбором является формат DXF, разработанный фирмой Autodesk в качестве универсального формата обмена чертежами между разными CAD-системами – этот формат открыт и бесплатен (в отличие, например, от формата DWG), широко распространен (практически во всех CAD-системах присутствует экспорт в формат DXF) и удобен для программного считывания [4].

Сложность заключается в большом объеме данных, хранимых в этом формате, что требует детального анализа структуры формата и выделения из него интересующих нас частей.

DXF-файл представляет собой текстовый файл в формате ASCII, заполненный так называемыми группами. Группа является минимальной структурной единицей файла. Внутреннее представление DXF файла

представлено на рисунке 1.



0
SECTION
2
HEADER
9
\$ACADVER
1
AC1027

Рисунок 1 – Представление DXF файла

Каждые две строчки являются группой. В первой строчке записывается код группы, во второй – значение. Код группы – это идентификатор того, что эта группа описывает. Например, коды 0-9 используются для обозначения строковых данных, а коды 10-59 – данных в формате с плавающей запятой двойной точности (double). Но коды используются не только для указания типа значения группы; они также обозначают, что именно содержится в значении группы. Например, код 0 используется в строго определенных случаях – начало секции, конец секции, начало таблицы и т.д., код 2 означает, что в значении группы описано имя (секции, таблицы, примитива...). Общая структура DXF-файла представлена на рисунке 2.

DXF-файл состоит из разделов, которые называются секциями (SECTION). Каждая секция начинается с двух групп – 0:SECTION и 2:Имя_секции, и заканчивается группой 0:ENDSEC. Количество и порядок секций в различных версиях DXF формата могут меняться, в новых версиях добавляются новые разделы.

К секциям, которые можно встретить в любой версии формата DXF относятся секции HEADER (заголовок), TABLES (таблицы), ENTITIES (примитивы) и BLOCKS (блоки примитивов).

В секции HEADER хранятся различные переменные чертежа, имеющие свое имя. Например, здесь хранится название и версия программы, создавшей чертеж, положение базовой точки чертежа, максимальные и минимальные координаты в чертеже и т.д.

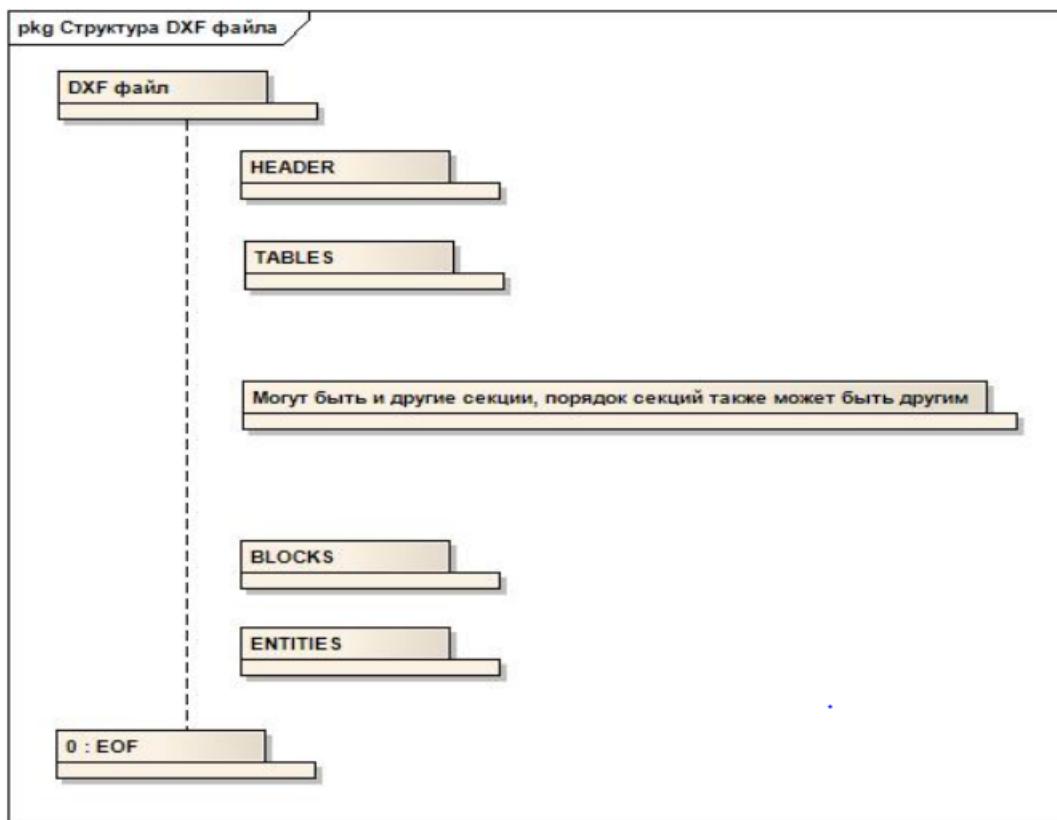


Рисунок 2 – Общая структура DXF файла

В секции TABLES хранятся массивы данных, такие как таблица слоев со всеми их свойствами, таблица стилей и т.д.

Особый интерес для считывания представляют секции BLOCKS и ENTITIES, в которых хранятся отображаемые объекты.

В секции ENTITIES хранятся данные о примитивах – базовых графических данных чертежа. Примитив – это какая-то геометрическая фигура, например, точка, линия, окружность, дуга и т.д. Существуют также сложные примитивы, состоящие из других примитивов, например, полилиния (POLYLINE), состоящая из вертексов, соединенных прямыми или дугами.

Начало примитива определяется группой 0:Имя_примитива, заканчивается описание примитива следующей группой 0:Имя_примитива или 0:ENDSEC. Сложные примитивы необходимо считывать по-особому, т.к. группа 0:Имя_примитива может встречаться внутри них. В этом случае конец сложного примитива будет находиться на первую группу 0:Имя_примитива после группы 0:SEQEND.

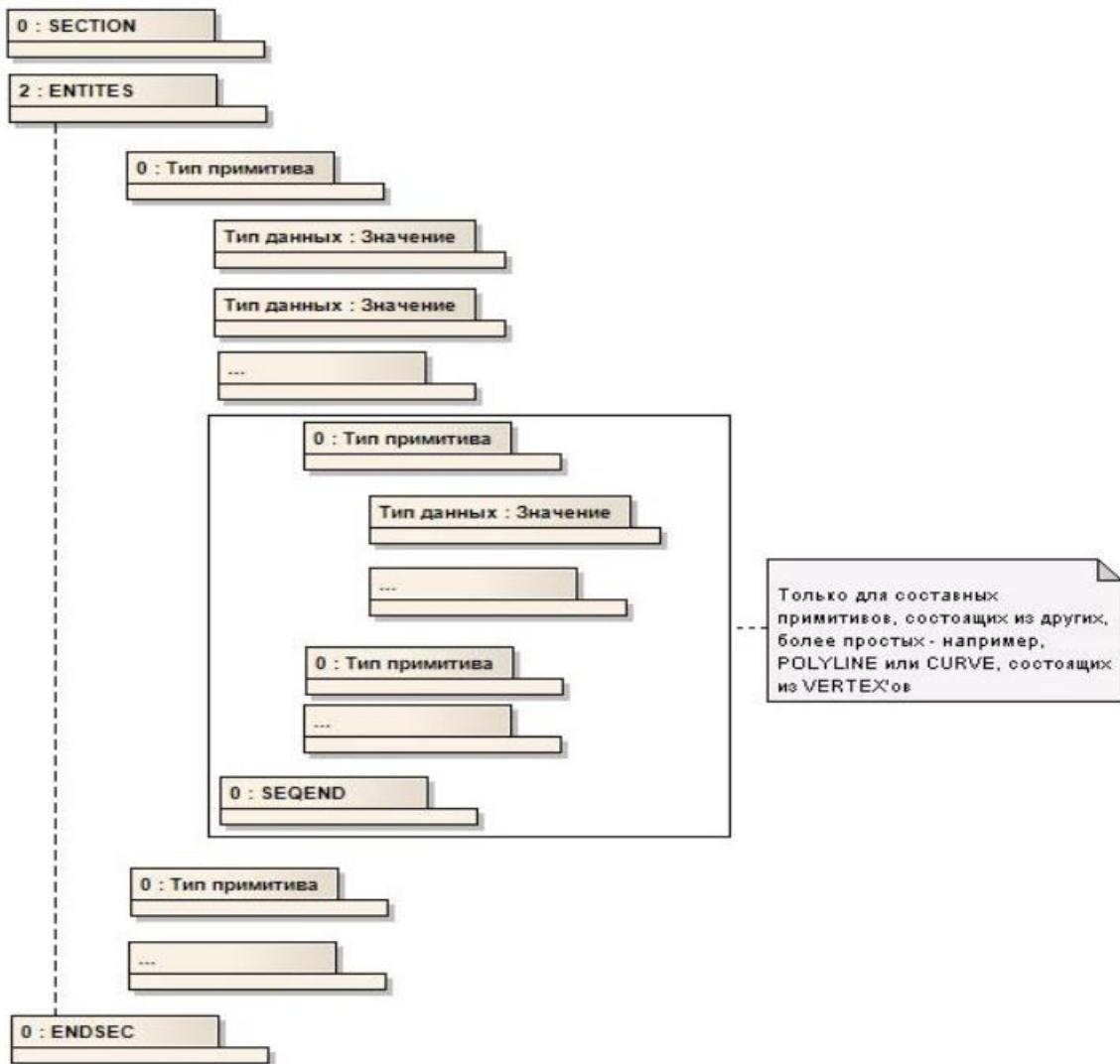


Рисунок 3 – Структура секции ENTITIES

Пример описания простейшего примитива – линии представлен на рисунке 4. Значение каждой строчки описано в комментариях.

В примере представлены поля, которые присутствуют в описании практически любой линии, в реальности в описании примитивов может присутствовать намного больше параметров. Полный список всех групповых кодов для примитивов и их значений представлен в DXF Reference [5].

В секции BLOCKS также содержится описание примитивов, однако здесь они объединены в блоки. Каждый блок имеет уникальный идентификатор и название и может использоваться в секции ENTITIES с помощью примитива INSERT.

```

0 ;Начало секции
SECTION

2 ;Секция Примитивы
ENTITIES

0 ;Начало примитива Линия
LINE

5 ;Уникальный идентификатор примитива
1EC

8 ;Номер слоя, на котором расположен примитив
0

10 ;Координата X начальной точки
0.0

20 ;Координата Y начальной точки
0.0

30 ;Координата Z начальной точки
0.0

11 ;Координата X конечной точки
452.55480855023052

21 ;Координата Y конечной точки
601.16555418995085

31 ;Координата Z конечной точки
0.0

```

Рисунок 4 – Описание линии в секции ENTITIES

Таким образом, можно, один раз нарисовать сложную фигуру, объединить входящие в нее примитивы в блок, присвоить ему уникальное имя и использовать его несколько раз в секции ENTITIES, вместо повторного рисования одной и той же фигуры. Преимуществом перед методом копирования и последующей вставки группы примитивов является изменение отображения всех вхождений блока в чертеж при изменении описания этого блока в секции. Общая структура секции BLOCKS показана на рисунке 5.

Именно наличие секции BLOCKS позволяет пользовательскому редактору добавлять дополнительные настройки отдельным элементам. Все сложные элементы описываются в этой секции и имеют уникальные идентификаторы, по которым можно выделить их из графических примитивов. Считав из секции ENTITIES такой элемент, мы имеем возможность добавить в него дополнительные функции, такие как изменение состояния и т.п.

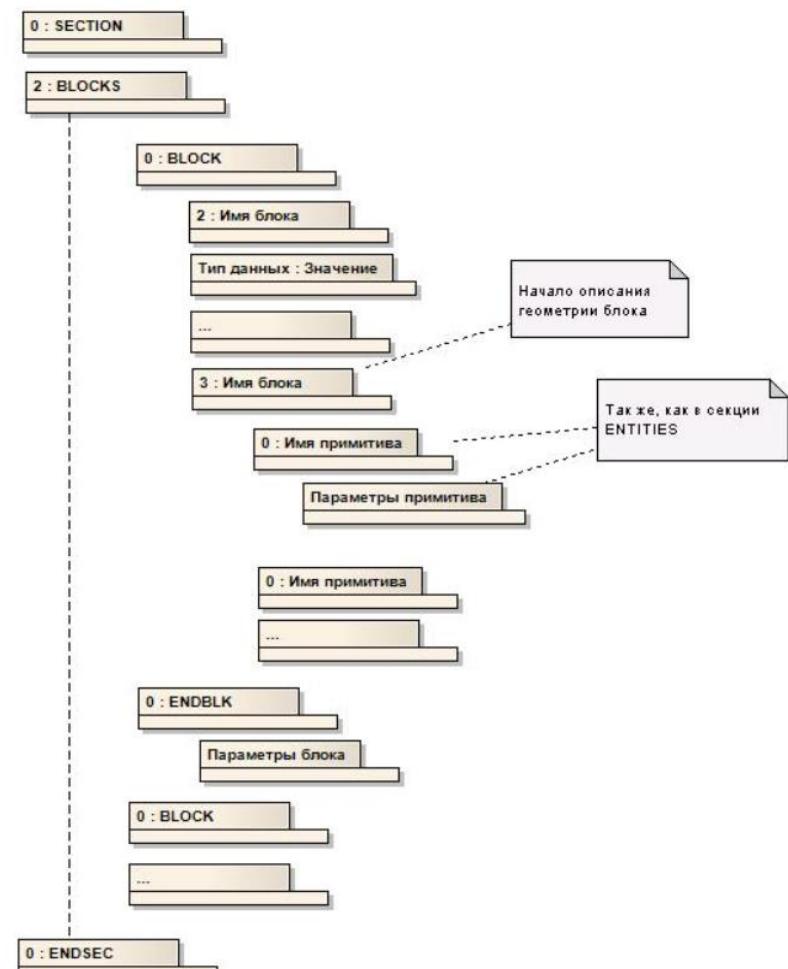


Рисунок 5 – Структура секции BLOCKS

За исключением заголовка, структуры отдельного блока из секции BLOCKS во многом совпадает с описанием примитивов в секции ENTITIES. Пример использования блока на рисунке 6.

0	INSERT
5	252
8	0
2	ВНЕШНИЙ_КОНТУР
10	260.65144090254489
20	503.97895235368458
30	0.0

Рисунок 6 – Использование блока в секции ENTITIES

Код 0 – имя команды. Код 5 – идентификатор. Код 8 – номер слоя. Код

2 – имя блока. Код 10 – смещение блока от начала координат по оси X. Код 20 – смещение блока от начала координат по оси Y. Код 30 – смещение блока от начала координат по оси Z.

Использованные источники:

1. Мацкевичюс, Д.А. Достоинства и недостатки растровой и векторной графики [Электронный ресурс] / Д. Мацкевичюс // it-dm.narod.ru: информатика в школе, вузе и жизни. – 2018. – URL: http://it-dm.narod.ru/it_DM/graph/it_DM_graph_compare.html .
2. Стуликова К.А., Полукаров Д.Ю. Проблемы отображения автономных систем с помощью графов //Известия Самарского научного центра Российской академии наук. – 2014. – Т. 16. – №. 4-2.
3. Капустин И.В., Полукаров Д.Ю. Реализация графовых структур данных с помощью библиотек JAVASCRIPT //IT & Transport/ИТ & Транспорт: сб. науч. статей – Самара, 2016. – С. 81-88.
4. Меньшиков П. В. Импорт графических данных из DXF файла средствами C# [Электронный ресурс] / П. Меньшиков// <http://www.3e-club.ru>: клуб «Трех инженеров». – 2012. – URL: http://www.3e-club.ru/view_full/ .
5. ENTITIES Section [Электронный ресурс]. – Электрон. Текстовые дан. – AutoDesk, 2018 – URL: https://www.autodesk.com/techpubs/autocad/acad2000/dxf/entities_section.htm.