

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем
Кафедра «Информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по дисциплине «Системы компьютерной графики»

на тему «**Параметризация элементов крепления круглых
оптических деталей**»

Выполнил: студент группы ИТ-22

Храбров Д. Е.

Проверил: старший преподаватель

Мурашко В. С.

Дата проверки: _____

Дата допуска: _____

Дата защиты: _____

Оценка работы: _____

Подписи членов комиссии

по защите курсовой работы: _____

Лист для рецензии

Содержание

| | |
|---|-----------|
| Введение | 4 |
| 1 Параметрическое проектирование | 7 |
| 2 Математическая модель | 15 |
| 2.1 Сущность математического моделирования | 15 |
| 2.2 Классификация математических моделей | 16 |
| 2.3 Описание технического объекта | 19 |
| 3 Алгоритмический анализ задачи | 21 |
| 3.1 Постановка задачи | 21 |
| 3.2 Описание входных данных | 21 |
| 3.3 Описание рабочих и выходных данных | 22 |
| 3.4 Графическая схема алгоритма | 23 |
| 4 Реализация математической модели в AutoLISP | 25 |
| 4.1 Руководство пользователя и программиста | 26 |
| 4.2 Описание результатов | 28 |
| Заключение | 29 |
| Список литературы | 30 |
| Приложение А Слайды детали | 31 |
| Приложение Б Исходный код AutoLISP программы | 32 |
| Приложение В Листинг DCL-файла | 48 |
| Приложение Г Результат (Параметрический чертеж А4) | 52 |
| Приложение Д Результат (Параметрический чертеж А3) | 53 |
| Приложение Е Результат (Параметрический чертеж А2) | 54 |

Введение

Ещё в древние времена существовала профессия шамана. Древние шаманы всегда были одними из самых уважаемых людей племени. Они могли предсказывать будущее, излечивать людей, смирять гнев богов. Шаманы совершали малопонятные простым людям действия и ритуалы, чем показывали свои магические силы, высшую неземную природу. Однако, действия шаманов обычно приводили к хорошим последствиям, и люди радовались этому и всё больше уважали шаманов. И лишь редкий мальчишка мог стать учеником шамана, чтобы в будущем продолжить его дело. Обучение длилось годами, и знания передавались от шамана к ученику в основном в устной форме.

Времена меняются, меняются и профессии людей. Довольно часто инженерам после разработки основной части задачи нужно также сделать документацию к своему проекту, чтобы нужные детали можно было создать на заводе, либо чтобы пользователям было легче использовать получившийся продукт. Однако, современные инженеры не имеют возможности обучать своих “учеников” годами устно, как это делали древние шаманы. Поэтому возникает проблема профессионального графического отображения какой-либо области задачи с максимально точными выходными результатами и максимально малыми затратами на построение. При этом часто бывает необходимо произвести не одно графическое построение, а целый ряд, что при простом ручном построении вызывает не только массу затруднений, связанных с их реализацией, но и проблему вычислительных мощностей. Именно поэтому были разработаны и используются по настоящее время так называемые системы автоматического проектирования (САПР) – универсальные программные приложения для разработки чертежей, конструкторской и технологической документации. Конечно же, древние шаманы не знали о САПР, поэтому они и использовали устную форму передачи данных. Возможно, будущим поколениям проектирование с использованием САПР покажется таким же устаревшим способ проектирования, как нам шаманство. Однако на текущий

момент проектирование с использованием САПР - самый современный способ проектирования.

Примеры различных САПР:

- Российские САПР: «Компас», «T-FLEX CAD» (АПР для машиностроения).
- Импортные проприетарные САПР: AutoCAD, ArhiCAD (САПР для архитектуры), Vocad-3D, CATIA (САПР для аэрокосмической промышленности), Cadmech (универсальная САПР для машиностроения), MathCAD (математическое моделирование), Pro/Engineer, Solid Edge, SolidWorks, Unigraphics, ADEM.
- Бесплатные САПР с открытым исходным кодом: FreeCAD, IndorCAD/Road (система проектирования автомобильных дорог компании ИндорСофт), QCad (Открытая 2х-мерная система проектирования), BRL-CAD (Открытая 3D система проектирования).

Кстати, цена САПР также играет немаловажную роль. Если инженер собирается работать не в странах СНГ, а за рубежом, то ему придётся купить САПР. Например, по данным сайта <http://autodesk.com/> полная версия AutoCAD 2008 на текущий момент (9 декабря 2007) стоит \$3995. Обновление с AutoCAD 2007 на AutoCAD 2008 - \$595. Древние шаманы от таких цен упали бы в обморок и предпочли бы купить 3 копченых мамонта, а для проектирования использовали бы OpenSource программное обеспечение. Которое по мнению многих людей не намного хуже, да и к тому же почти полностью бесплатно.

При выполнении данной курсовой работы мы будем придерживаться Единой Системы Конструкторской Документации (ЕСКД). ЕСКД - это комплекс стандартов, устанавливающих взаимосвязанные нормы и правила по разработке, оформлению и обращению конструкторской документации, разрабатываемой и применяемой на всех стадиях жизненного цикла изделия. Стандарт ЕСКД активно используется только в странах СНГ.

Данный курсовой проект является актуальным на текущее время для стран СНГ, поскольку в ходе его выполнения будет наработан опыт работы с востребованными и признанными в нынешнее время системами проектирования и правилами оформления технической документации в соответствии с ЕСКД. Однако, приобретаются практические навыки работы с AutoCAD, которые могут пригодиться и за рубежом.

В данном курсовом проекте осуществляется разработка и реализация программного обеспечения обладающего возможностью построения в графическом пакете AutoCAD заданной детали по необходимым данным, заранее вводимым пользователем. Благодаря использованию переменных данных при построении чертежа детали наша программа обладает гибкостью – меняя различные параметры в диалоговом окне перед построением мы изменяем и чертеж детали, то есть для построения детали аналогичного типа, но с другими параметрами, пользователю не придется писать новую программу.

1 Параметрическое проектирование

Система автоматизации проектных работ (САПР) или САД (англ. Computer-Aided Design) — программный пакет, предназначенный для создания чертежей, конструкторской и/или технологической документации и/или 3D моделей. В современных системах проектирования САД получает данные из систем твёрдотельного моделирования САЕ (Computer-aided engineering), и передаёт в САМ (Computer-aided manufacturing) для подготовки производства.

Обычно охватывает создание геометрических моделей изделия (твёрдых, трёхмерных, составных), а также генерацию чертежей изделия и их сопровождение. Следует отметить, что русский термин «САПР» по отношению к промышленным системам имеет более широкое толкование, чем «САД» — он включает в себя как САД, так и элементы САМ, а иногда и элементы САЕ. Древние шаманы, конечно же, обо всём это и не подозревали. Однако каждый современный инженер для увеличения производительности труда должен знать различные САПР и их различия.

Рассмотрим следующие САПР:

- ArhiCAD - программное обеспечение для 3D дизайна и проектирования зданий, сооружений;
- КОМПАС-3D - система трёхмерного твёрдотельного моделирования;
- MathCAD - математически ориентированные универсальные системы;
- AutoCAD - 2-х и 3-х мерная система автоматизированного проектирования и черчения.

Программа ArhiCAD компании Graphisoft - один из самых мощных на сегодня инструментов архитектурного проектирования - не просто позволяет, используя интеллектуальные объекты, создать трёхмерный мир проекта. Она меняет само представление об архитектурных системах автоматизированного

проектирования. ArhiCAD — это много больше, чем обычный электронный кульман: это мощная, сложная среда, позволяющая достигать результата в кратчайшие сроки. Уникальная технология “Виртуальное Здание” открывает возможность по-настоящему творческой и в то же время высокопроизводительной работы: архитектор занимается дизайном проекта, ArhiCAD автоматически подготавливает документацию.

ArhiCAD - программа, обеспечивающая разработку архитектурно-дизайнерских решений. На любом этапе работы можно увидеть проектируемое здание в трехмерном виде, в разрезе, в перспективе, сделать анимационный ролик. В ArhiCAD можно одновременно работать над созданием проекта и составлять сопутствующую строительную документацию, так как программа хранит информацию о проектируемом здании: планы, разрезы, перспективы, перечень необходимых стройматериалов, а также замечания архитектора, сделанные в процессе работы.

Программа Архикад предназначена не только для профессиональных архитекторов и дизайнеров, но и для неискушенных пользователей - людей, занятых перепланировкой квартиры, постройкой дома и благоустройством участка, но нужно будет ознакомиться с технической документацией.

Система КОМПАС-3D от компании “АСКОН” предназначена для создания трехмерных ассоциативных моделей отдельных деталей и сборочных единиц, содержащих как оригинальные, так и стандартизованные конструктивные элементы. Параметрическая технология позволяет быстро получать модели типовых изделий на основе однажды спроектированного прототипа. Многочисленные сервисные функции облегчают решение вспомогательных задач проектирования и обслуживания производства.

Ключевой особенностью КОМПАС-3D является использование собственного математического ядра и параметрических технологий, разработанных специалистами АСКОН.

Основная задача, решаемая системой — моделирование изделий с целью существенного сокращения периода проектирования и скорейшего их запус-

ка в производство. Эти цели достигаются благодаря возможностям быстрого получения конструкторской и технологической документации, необходимой для выпуска изделий (сборочных чертежей, спецификаций, детализовок), передачи геометрии изделий в расчетные пакеты, передачи геометрии в пакеты разработки управляющих программ для оборудования с ЧПУ, создания дополнительных изображений изделий (например, для составления каталогов, создания иллюстраций к технической документации).

Средства импорта/экспорта моделей (КОМПАС-3D поддерживает форматы IGES, SAT, XT, STEP, VRML) обеспечивают функционирование комплексов, содержащих различные CAD/CAM/CAE системы.

Широкую известность и заслуженную популярность еще в середине 80-х годов приобрели интегрированные системы для автоматизации математических расчетов класса MathCAD, разработанные фирмой MathSoft (США). По сей день они остаются единственными математическими системами, в которых описание решения математических задач дается с помощью привычных математических формул и знаков. Такой же вид имеют и результаты вычислений. Так что системы MathCAD вполне оправдывают аббревиатуру CAD (Computer Aided Design), говорящую о принадлежности к наиболее сложным и продвинутым системам автоматического проектирования — САПР.

С момента своего появления системы класса MathCAD имели удобный пользовательский интерфейс — совокупность средств общения с пользователем в виде масштабируемых и перемещаемых окон, клавиш и иных элементов. У этой системы есть и эффективные средства типовой научной графики, они просты в применении и интуитивно понятны. Словом, системы MathCAD ориентированы на массового пользователя — от ученика начальных классов до академика.

MathCAD — математически ориентированные универсальные системы. Помимо собственно вычислений они позволяют с успехом решать задачи, которые с трудом поддаются популярным текстовым редакторам или электронным таблицам. С их помощью можно не только качественно подготовить

тексты статей, книг, диссертаций, научных отчетов, дипломных и курсовых проектов, они, кроме того, облегчают набор самых сложных математических формул и дают возможность представления результатов, в изысканном графическом виде.

Если бы древние шаманы знали высшую математику - они бы использовали MathCAD. Но ему также есть и бесплатные альтернативы: Octave и Maxima. Однако, исследование существующих математических пакетов не входит в цели данной работы.

AutoCAD — 2-х и 3-х мерная система автоматизированного проектирования и черчения компании Autodesk. Семейство продуктов AutoCAD является одним из наиболее распространённых САПР в мире.

Компания Autodesk занимается разработкой системы автоматизированного проектирования AutoCAD более 20-ти лет. За это время были созданы тысячи дополнений и специализированные решения от сторонних фирм и самой компании Autodesk. На данный момент в мире насчитывается около шести миллионов пользователей AutoCAD, а число лицензированных языковых переводов - 18.

DXF (англ. Drawing eXchange Format) — открытый формат файлов для обмена двумерной графической информацией между приложениями САПР. Был создан фирмой Autodesk для системы AutoCAD. Поддерживается практически всеми САД-системами на платформе PC.

По мере того, как AutoCAD становился всё сложнее и поддерживал всё более сложные типы объектов, DXF становился всё менее полезен. Новые объекты в спецификации формата описывались не полностью либо не описывались вовсе. Большинство разработчиков коммерческих приложений, включая конкурентов Autodesk, в качестве основного формата обмена с AutoCAD используют формат DWG, через библиотеки, предоставляемые некоммерческой организацией Open Design Alliance, которой была выполнена обратная разработка формата DWG.

Конструктор, освоивший графический редактор AutoCAD и убедив-

шийся на практике, что эффект от автоматизации чертежных работ незначителен, может обнаружить, что система AutoCAD предоставляет гораздо более эффективные возможности, чем возможности графического редактора. Оказывается, чертежи можно получать без всяких затрат времени и усилий на "рисование". Чертеж, который в ручном варианте или с помощью графического редактора получается за несколько часов и даже дней, может быть сформирован за минуты (не считая времени автоматического вывода чертежа на графопостроитель). Более того, чертеж может получить конструктор, практически ничего не знающий о системе AutoCAD. Но чтобы получить такой эффект, кто-то предварительно должен был написать программу для генерации семейства сходных чертежей. Именно этой программой теперь может пользоваться любой конструктор для получения любого чертежа, принадлежащего данному семейству. Шаманы, к сожалению, не могли использовать автоматизацию процесса обучения. Мы должны учитывать ошибки предков и использовать автоматизацию разработки документации там, где нужно.

В соответствии с общей идеологией системы Автокад главное ее предназначение - вовсе не рисование чертежей на компьютере (это приводит к падению производительности труда конструктора в 2..3 раза), а создание на ее основе специализированной САПР определенного класса изделий. Такие САПР резко, в 15..20 раз повышают производительность.

Многие изделия представляются в виде вариантных чертежей, когда изделие состоит из постоянной части с варьируемыми размерами и вариантной части с уникальной геометрией. Например, для станочных тисок проектируются уникальные губки под каждую конкретную деталь, фиксируемую этими тисками, но корпус, ходовой винт, струбцина и прочие ДСЕ тисок остаются конструктивно неизменными. В данном случае определение конструктивно подобного изделия не выполняется.

Сущность параметрического проектирования состоит в создании математической модели класса конструктивно однородных изделий, а затем в генерации изображений этих изделий по набору задаваемых размерных па-

раметров.

При параметрическом проектировании конструктор запускает программу, рассчитанную на определенный класс изделий, и вводит требуемые размеры. Программа отрисовывает на экране чертеж детали. Конструктор оценивает его и при необходимости вводит размеры снова до достижения требуемого результата. Одновременно может рассчитываться масса детали, что позволяет контролировать ее "на ходу прямо в процессе проектирования.

Таким образом, программа-параметризатор работает по следующему алгоритму:

1. Ввод исходных данных;
2. Отрисовка текущего варианта;
3. Запрос пользователю: повторить?
4. Если да, то переход на пункт 1 данного алгоритма
5. Конец

Для проектирования формирования чертежей используются два основных метода: генерирующий и вариантный.

Генерирующий метод позволяет создавать чертежи из отдельных элементов. Принцип работы системы, использующей этот метод, основан на разделении чертежа детали на элементы и создании новых чертежей из существующих элементов.

Суть вариантного метода состоит в том, что при создании чертежей определённого класса деталей из уже созданных рисунков выбираются самые сложные для этого класса детали. Далее на их основе разрабатывается чертёж типовой детали. [5, стр. 68]

Для создания программ, выпускающих чертежи, в системе AutoCAD имеется мощное средство - язык AutoLISP . Но чтобы использовать богатые возможности AutoLISP, нужно научиться программировать на этом языке.

Конструктору, ранее не знакомому с программированием, освоить язык графического программирования AutoLISP сложнее, чем язык графического редактирования (или, как говорят, "электронный кульман"). Однако, практика показывает, что это вполне по силам. Графический язык программирования AutoLISP является расширением языка программирования Лисп. Лисп - это язык высокого уровня, ориентированный на обработку списков, который выбран в качестве базового потому, что графические примитивы (начиная, с точки), блоки, наборы примитивов и блоков удобно представляются в виде списков.

В составе системы AutoCAD поставляется интерпретатор с языка AutoLISP (входящий в модуль ADE-3). Если при генерации AutoCAD интерпретатор AutoLISPа был подключен, то он загружается в оперативную память после запуска графического редактора AutoCAD и доступен в течение всего сеанса работы с AutoCAD. Таким образом, графический редактор AutoCAD и интерпретатор AutoLISP представляют собой единую систему: любая функция AutoLISP может быть вызвана из графического редактора и любая команда редактора может быть использована в программе на AutoLISP.

Возможности применения AutoLISPа весьма широки и разнообразны. Наиболее характерны следующие классы применений: Программирование чертежей с параметризацией. Создается программа, позволяющая при каждом обращении к ней формировать новый чертеж, отличающийся от предыдущих чертежей, построенных этой же программой, размерами, а также, возможно, и топологией: могут появиться новые элементы обогащения, сечения, измениться текстовая часть чертежа и т.д. Время получения чертежа с помощью такой программы может быть в десятки раз меньше времени, необходимого для его создания с помощью редактора AutoCAD, и, что не менее важно, получить чертеж сможет любой конструктор, не знакомый ни с командами AutoCAD, ни с AutoLISP.

Создание и использование графических баз данных. Если накоплено

большое количество чертежных файлов, программ на AutoLISP, соответствующих чертежным фрагментам, деталям, узлам, то их можно в некотором смысле считать графической базой данных. Программы на AutoLISP в сочетании с пользовательскими меню могут организовывать просмотр, поиск, подключение к объектам их частей и т.п. Тогда работа конструктора в системе AutoCAD будет сводиться к поиску нужных объектов (сборочных единиц, деталей) или частей чертежа, обращению к соответствующим LISP-программам и ответам на вопросы этих программ. А ведь все тем же древним шаманам приходилось всё помнить в голове. Каково же будет инженеру помнить тысячи файлов, расположенных на жестком диске?

Анализ и (или) автоматическое преобразование изображений. Программа на AutoLISP может воспринимать чертеж на экране, построенный с помощью графического редактора и обсчитывать его; например, могут быть решены такие задачи: обнаружение пересечений трубопроводов, разведенных в пространстве; подсчет суммарной длины трасс; расчет центра масс и моментов инерции; выдача на печать спецификации по сборочному чертежу и т.п. Программа также может быстро осуществить преобразование изображения, на которое при работе в графическом редакторе пришлось бы затратить значительное время, например: заменить все вставки одного типа на вставки другого типа из какого-либо чертежного файла; перенести все объекты с одного слоя на другой слой; повернуть все блоки на заданный угол - каждый относительно своей базовой точки и т.п.

Расширение системы команд графического редактора AutoCAD и построение на основе универсального редактора специализированных САПР, имеющих гораздо более простой и естественный для пользователей язык, ориентированный на конкретную предметную область. В этом случае хорошим дополнением к AutoLISP является возможность создания пользовательских меню. [3, стр. 115]

2 Математическая модель

2.1 Сущность математического моделирования

Сущность математического моделирования заключается в том, что деформация модели процесса изучается не на физической модели, а непосредственно на математической модели с применением ЭВМ.

Смысл моделирования заключается в том, чтобы по результатам опытов на модели можно было судить о явлениях, происходящих в натуральных условиях. Основные виды моделирования, с которыми мы встретимся в дальнейшем: 1) физическое, когда модель воспроизводит изучаемое явление с сохранением его физической природы и геометрического подобия и отличается от оригинала только размерами и значением физических параметров (скорости, вязкости, модуля упругости и т. д.); 2) аналоговое (разновидность математического), когда модель относится к другой области физических явлений или не сохраняет геометрического подобия, например моделирование механических колебаний электрическими или моделирование течения жидкости течением электрического тока.

Достоинство метода математического моделирования заключается в том, что различные по характеру процессы могут иметь сходные математические модели. Это свойство аналогий позволяет, во-первых, при решении задач моделирования и оптимизации использовать аналоговую вычислительную технику, а во-вторых, в результате решения одной конкретной задачи получать информацию о свойствах целого класса объектов, характеризующихся аналогичными математическими описаниями. Последнее обстоятельство является одним из важнейших следствий применения метода математического моделирования. Становится возможным использовать результаты, полученные при изучении одних объектов, для исследования других, вероятно, даже относящихся к другой области науки или техники.

И даже древние шаманы понимали пользу от моделирования процесса.

Ведь довольно часто перед тем, как излечить человека от болезни новым методом, этот метод применялся на животных. Также шаманы наблюдали за различными частными природными процессами и из их анализа делать аналогичные выводы об общих природных процессах. Например, могли предсказывать дождь или похолодание по поведению животных. Аналогично и современный инженер. Чтобы построить мост по новой конструкции вовсе не обязательно строить его вживую и наблюдать, как после годов постройки он развалится. Гораздо лучше смоделировать данную ситуацию на компьютере и ещё на этапе проектирования внести нужные изменения.

2.2 Классификация математических моделей

Классификация в любой области знаний чрезвычайно важна. Она позволяет обобщить накопленный опыт, упорядочить понятия предметной области. Не является исключением в этом смысле и математическое моделирование. В таблице 1 показаны виды математических моделей по различным признакам классификации.

Математические модели на микроуровне производственного процесса отражают физические процессы, протекающие, например, при резании металлов. Они описывают процессы на уровне перехода (прохода).

Математические модели на макроуровне производственного процесса описывают технологические процессы.

Математические модели на метауровне производственного процесса описывают технологические системы (участки, цехи, предприятие в целом).

Структурные математические модели предназначены для отображения структурных свойств объектов. Например, в САПР ТП для представления структуры технологического процесса, расцеховки изделий используется структурно – логические модели.

Функциональные математические модели предназначены для отображения информационных, физических, временных процессов, протекающих в

Таблица 1 – Виды математических моделей

| Признаки классификации | Виды математических моделей |
|--|---|
| Принадлежность к иерархическому уровню | Модели микроуровня Модели макроуровня Модели метауровня |
| Характер отображаемых свойств объекта | Структурные Функциональные |
| Способ представления свойств объекта | Аналитические Алгоритмические Имитационные |
| Способ получения модели | Теоретические Эмпирические |
| Особенности поведения объекта | Детерминированные Вероятностные |

работающем оборудовании, в ходе выполнения технологических процессов и так далее.

Аналитические математические модели представляют собой явные математические выражения выходных параметров как функций от параметров входных и внутренних.

Алгоритмические математические модели выражают связи между выходными параметрами и параметрами входными и внутренними в виде алгоритма.

Имитационные математические модели – это алгоритмические модели, отражающие развитие процесса (поведение исследуемого объекта) во времени при задании внешних воздействий на процесс (объект). Например, это модели систем массового обслуживания, заданные в алгоритмической форме.

Теоретические математические модели создаются в результате исследования объектов (процессов) на теоретическом уровне. Например, существуют

выражения для сил резания, полученные на основе обобщения физических законов. Но они не приемлемы для практического использования, так как очень громоздки и не совсем адаптированы к реальным процессам обработки материалов.

Эмпирические математические модели создаются в результате проведения экспериментов (изучения внешних проявлений свойств объекта с помощью измерения его параметров на входе и выходе) и обработки их результатов методами математической статистики.

Детерминированные математические модели описывают поведение объекта с позиций полной определенности в настоящем и будущем. Примеры таких моделей : формулы физических законов, технологические процессы обработки деталей и так далее.

Вероятностные математические модели учитывают влияние случайных факторов на поведение объекта, то есть оценивают его будущее с позиций вероятности тех или иных событий. Примеры таких моделей: описание ожидаемых длин очередей в системах массового обслуживания, ожидаемых объемов выпуска сверхплановой продукции производственным участком, точности размеров в партии деталей с учетом явления рассеяния и так далее. [6, стр. 90]

К математическим моделям предъявляются следующие основные требования:

- Универсальность. Характеризует полноту отражения в ней свойств реального объекта. Математическая модель отражают не все, а лишь некоторые свойства реального объекта.
- Точность. Оценивается степень совпадения значений выходных параметров реального объекта и значений тех же параметров, рассчитанных с помощью модели.
- Адекватность. Это способность математической модели отражать заданные свойства объекта с погрешностью, не выше заданной.

- Вычислимость, то есть возможность ручного или с помощью ЭВМ исследования качественных и количественных закономерностей функционирования объекта (системы).
- Модульность, то есть соответствие конструкций модели структурным составляющим объекта (системы).
- Алгоритмизируемость, то есть возможность разработки соответствующих алгоритма и программы, реализующей математическую модель на ЭВМ.
- Наглядность, то есть удобное визуальное восприятие модели.

2.3 Описание технического объекта

Эксцентриковая оправа представлена на рисунке 1. Её целесообразно использовать в случаях, когда необходима точная юстировка линз относительно оптической оси прибора. То есть примеряются оправы, имеющие эксцентриситет. Величина эксцентриситета (e) для всех типов оправ - порядка 0,5 мм, что совместно с кольцом позволяет смещать ось до 1мм в любую сторону.

Оправы крепятся в приборе резьбовым кольцом, а линзы - закатыванием. В таблице 2 можно увидеть приблизительные соотношения размеров данных оправ.

Таблица 2 – Сравнительные соотношения размеров оправ

| d_1 | d_2 | d_3 | D_2 | D_3 | D_4 |
|--------------|----------------|----------------|-------------|-------------|-------------|
| , , 18, , 30 | $(d_1 - 1, 5)$ | $(d_1 + 0, 8)$ | $(d_1 + 3)$ | $(d_1 + 5)$ | $(d_1 + 3)$ |
| , , 30, , 50 | $(d_1 - 2)$ | $(d_1 + 1)$ | $(d_1 + 4)$ | $(d_1 + 7)$ | $(d_1 + 3)$ |

Однако, данная таблица не совсем соответствует истине. Например, при

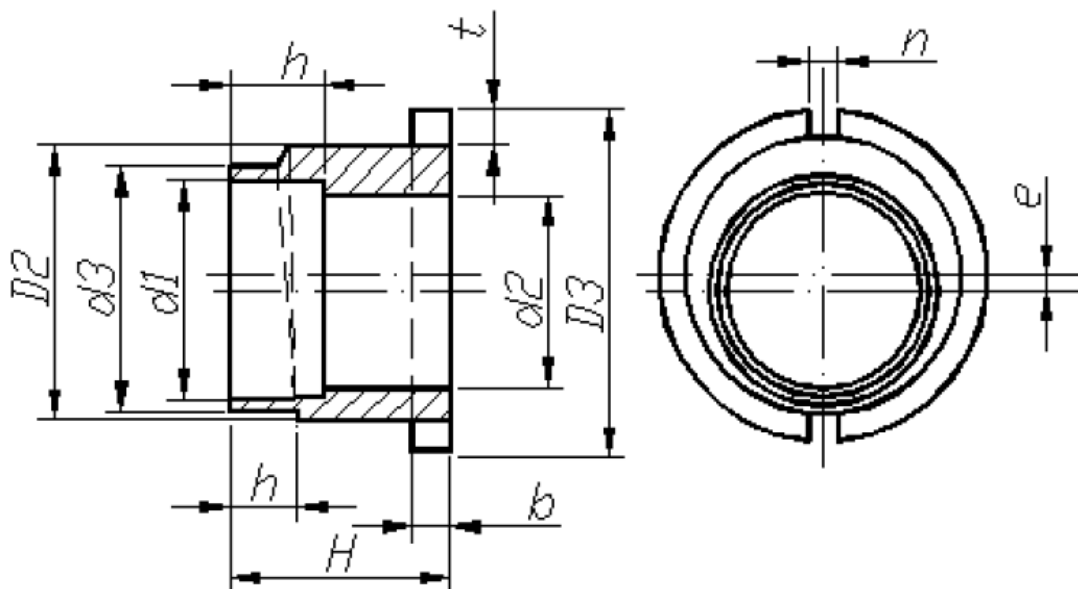


Рисунок 1 – Эскиз детали

определённых значения эксцентриситета значение $d3$ может превышать $D2$, чего не должно случаться. Поэтому были выведены относительные соотношения сторон, учитывающие подобные случаи. [2, стр. 84] Именно они и реализованы в программе.

Как видно из рисунка, левая проекция состоит только из линий. Все размеры строго определены. Точку привязки можно положить на любую вершину, либо на пересечении оси со стороной. В программной реализации было выбрано использовать точку пересечения самой левой стороны и оси больших несмещённых диаметров. От неё последовательно по координатам строится вся остальная деталь. Так как деталь полностью не симметрична, то построить её методом черчения одной части а потом её отражением не получится. Однако, есть точки, имеющие похожие координаты. Если избрать точку привязки за $(0;0)$, то некоторые координаты будут полностью симметричны и отличаться лишь знаком ординаты. Этот факт должен использоваться при написании программы.

3 Алгоритмический анализ задачи

3.1 Постановка задачи

1. Начертить рабочий параметрический эскиз заданной детали.
2. Проанализировать и выявить все размеры, которые необходимы для программирования данной детали. Обозначить эти размеры на эскизе.
3. Продумать последовательность рисования детали. Указать все точки, расчет которых необходим для отображения детали полностью. Обозначить точки на эскизе.
4. Написать программу, отображающую заданную деталь на поле чертежа. Размеры детали должны вводиться в программу пользователем с помощью языка DCL. Названия переменных и точек должны соответствовать указанным в эскизе.
5. Предусмотреть возможность создания диалогового окна выбора формата листа A1, A2, A3, A4 с атрибутами, на котором будет отображаться деталь.
6. Организовать отображение контура детали в заданном масштабе.
7. Также запрограммировать отображение детали в заданном масштабе под углом.
8. Отражение должно быть организовано в подпрограмме параметрического рисования, после отображения контура детали.

3.2 Описание входных данных

Все параметры детали можно выразить через один параметр - $d1$. Однако, пользователь также может захотеть изменить значения других параметров. Например, значение эксцентриситета (e), общую длину детали (H),

либо какой-нибудь другой параметр. Для реализации лучшим вариантом будет ввод параметра $d1$, расчёт остальных параметров автоматически и предоставление пользователю возможность редактировать их.

Как организована форма для ввода данных можно посмотреть на рисунке 2. Её исходный код на языке DCL приведен в приложении В.

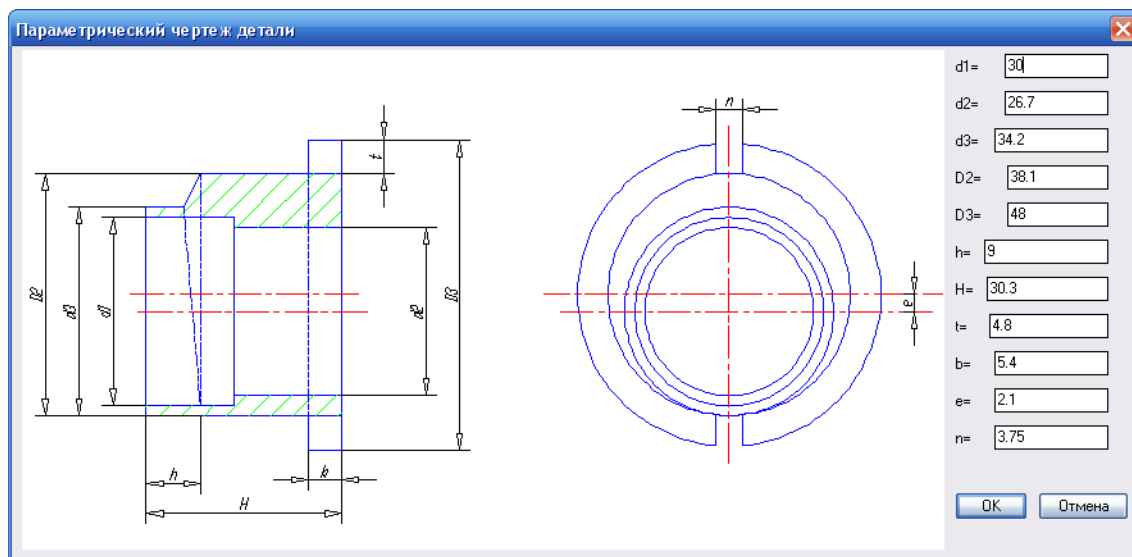


Рисунок 2 – Форма ввода данных пользователя (основная форма)

По умолчанию поле $d1$ активное. При его изменении (вводе в него числа) остальные поля устанавливаются автоматически. Далее их можно редактировать нужным образом. При вводе некорректных данных отображается ошибка и просьба ввести корректные данные.

По начальному эскизу детали для более удобного анализа задачи были построены слайды детали, на которых были просчитаны точки, по которым строится деталь, а также отображены все нужные размеры. Эти слайды приведены в приложении А. Так как основная часть программы писалась при помощи этих слайдов, то их тоже можно считать входными данными.

3.3 Описание рабочих и выходных данных

В программе использованы следующие переменные:

d1 d2 d3 bd2 bd3 h bh tt b e - данные, отображённые на эскизе. Причем функция `set_fields` заполняет поля ввода в зависимости от d1 сама.

p[n], dp[n], dl[n], sh[n], os[n], dr[n], tp[n] - точки, которые будут использованы для рисования. Используются только в функции `DrawDetail`. (вместо [n] подставить номер)

`РусуНоК` - данная переменная содержит в себе все примитивы, относящиеся к детали. Используется при масштабировании, повороте, перемещении и т.п.

`on_d1...` - содержат значения полей ввода, соответствующих данным переменным.

`flag, what_next` - используются для организации циклов.

Для удобства реализации DCL-файл располагается в папке с основной программой. Там же расположен файл `empty.dwg`, который нужно предварительно загрузить. Таким образом программа узнаёт какая директория текущая и привязывается к относительному пути на локальном компьютере, а не к полному. Это одна из особенностей программы. Ведь даже древние шаманы, когда объясняли простым людям дорогу куда-нибудь, использовали относительные пути. Например, после этого дерева нужно повернуть в красноглазой скале. Ведь данный шаман не говорит, куда нужно повернуть относительно текущей точки, он говорит, куда нужно повернуть после дерева. То есть использует относительный путь.

Выходными данными служат построенные на основе вышеперечисленных входных данных – чертёж детали.

3.4 Графическая схема алгоритма

Графическая схема алгоритма позволяет увидеть саму программу в виде схемы, которая состоит из определённой последовательности блоков. Графическая схема, также называемая “блок-схема”, наглядно показывает всю цепочку последовательности операций, которые происходят в программе.

Графическая схема алгоритма работы программы представлена на рисунке 3.

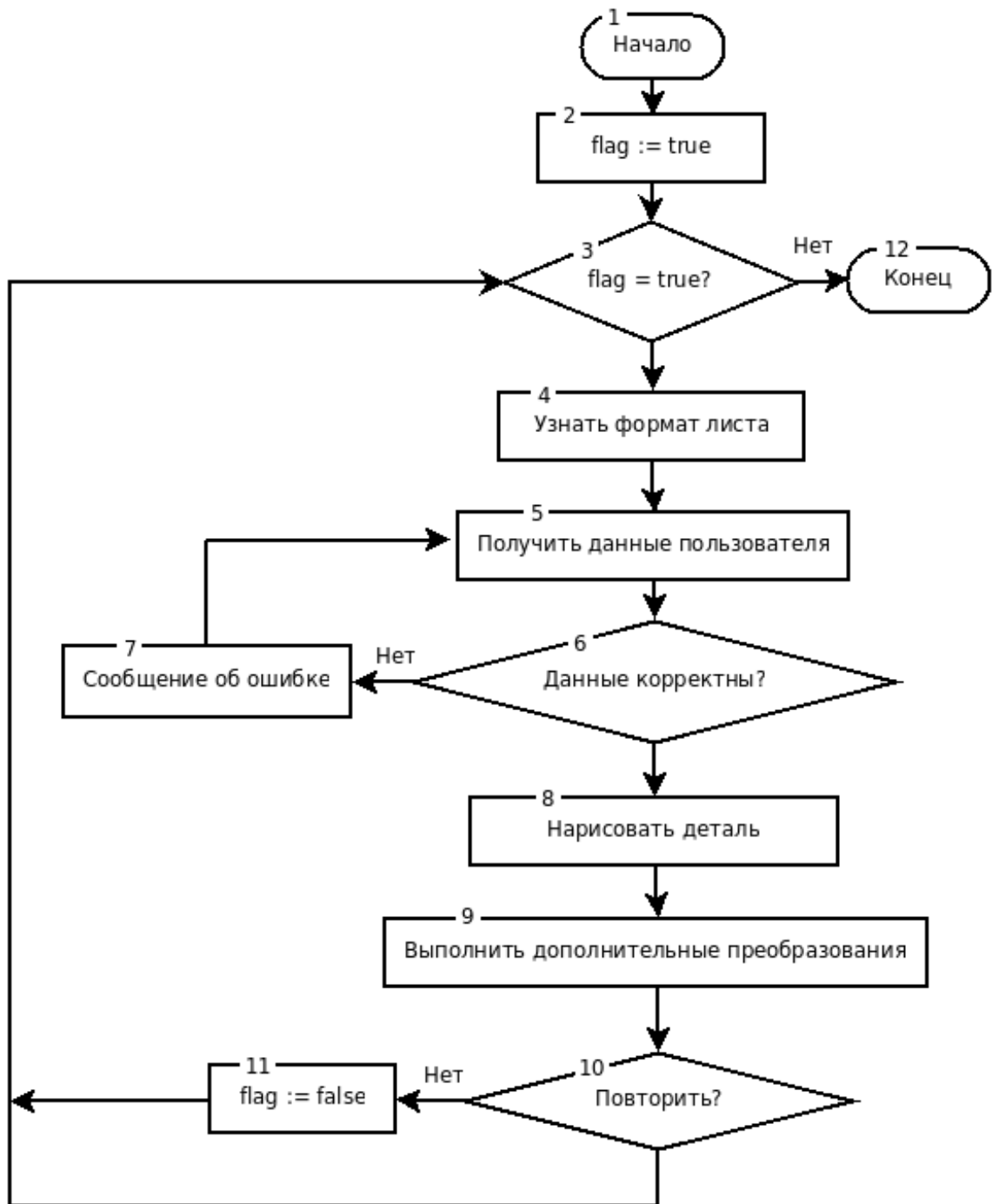


Рисунок 3 – Графическая схема алгоритма

4 Реализация математической модели в AutoLISP

Рабочая модель детали представлена на рисунке 4. На ней можно увидеть предварительный чертёж основной проекции и точки, по которым строится деталь. Также можно увидеть основную ось детали и вспомогательную, смещённую вниз относительно основной на величину e .

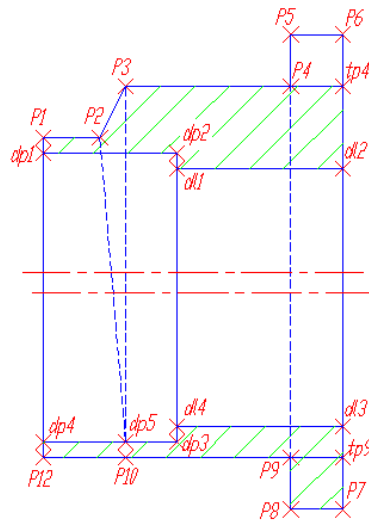


Рисунок 4 – Рабочая модель детали

Координата точки в программе обычно узнаётся через координату предыдущей точки и входные параметры (такие, как $d1$, H и подобные). Ниже представлен формулы для вычисления координат точек:

$$\begin{aligned}
 P_1 &= \left(0, \frac{d_3}{2} - e\right) & P_2 &= (h - e, P_{1y}) \\
 P_3 &= \left(P_{2x} + e, \frac{D_2}{2}\right) & P_4 &= (H - b, P_{3y}) \\
 P_5 &= (P_{4x}, P_{4y} + tt) & P_6 &= (H, P_{5y}) \\
 P_7 &= (P_{6x}, -P_{6y}) & P_8 &= (P_{5x}, -P_{5y}) \\
 P_9 &= (P_{4x}, -P_{4y}) & P_{10} &= (P_{3x}, -P_{3y}) \\
 P_{11} &= \left(P_{10x}, -\left(\frac{d_3}{2} + e\right)\right) & P_{12} &= (P_{3x}, -P_{3y}) \\
 tp_4 &= (H, P_{4y}) & tp_9 &= (H, P_{9y})
 \end{aligned}$$

$$\begin{aligned}
dp_1 &= (0, \frac{d_1}{2} - e) & dp_2 &= (h + b, dp_{1y}) \\
dp_3 &= (dp_{2x}, -(\frac{d_1}{2} + e)) & dp_4 &= (dp_{1x}, dp_{3y}) \\
dp_4 &= (h, dp_{3y}) & dl_1 &= (dp_{2x}, \frac{d_2}{2} - e) \\
dl_2 &= (P_{6x}, dl_{1y}) & dl_3 &= (dl_{2x}, -(\frac{d_2}{2} + e))
\end{aligned}$$

При написании программы были использованы различные функции AutoLISP'a:

`command` - вызов команды AutoCAD. Использовалось для рисования примитивов, выноски размеров, установки системных переменных и т.п.

`list` - создание списка из переданных параметров

`nth` - взять элемент с переданным номером из списка

`setq` - установка значения переменной

`entlast` - получение последнего нарисованного примитива

условия (IF), циклы (WHILE) и д.р.

Исходный код AutoLISP программы приведен в приложении Б.

4.1 Руководство пользователя и программиста

Для запуска программы нужно загрузить файл `empty.dwg`, находящийся в директории программы. Это пустой файл, однако таким образом программа узнает текущую директорию. Далее нужно загрузить саму программу в память AutoCAD. Для этого нужно в меню "Сервис" AutoCAD выбрать пункт "AutoLISP" и в выпавшем подменю выбрать пункт "Загрузить".

После этого нужно запустить программу на выполнение. Для этого в командной строке AutoCAD нужно ввести "`(run_dialog)`" (без кавычек). Это вызовет основную рабочую функцию программы - `run_dialog`. Она уже выведет нужные диалоги для работы с пользователем и нарисует деталь.

Первым функция `run_dialog` предложит выбрать формат листа. Данная форма показана на рисунке 5.

После выбора формата листа будет выведено основное диалоговое окно. Оно было рассмотрено на рисунке 2. Если ввести неправильные параметры -

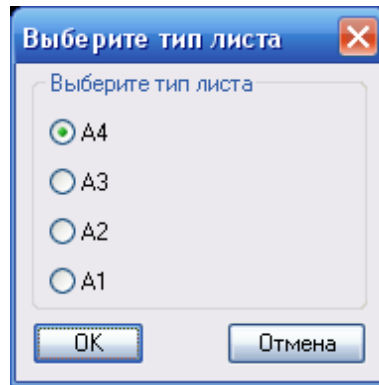


Рисунок 5 – Форма выбора формата листа

будет показано сообщение об ошибке. Если всё правильно - программа нарисует чертёж в соответствии со введёнными параметрами и нарисует обрамление в соответствии с данным форматом листа.

После этого деталь можно повернуть, отмасштабировать, переместить. Об этом программа выдаст запрос в командную строку. Это отображено на рисунке 6.

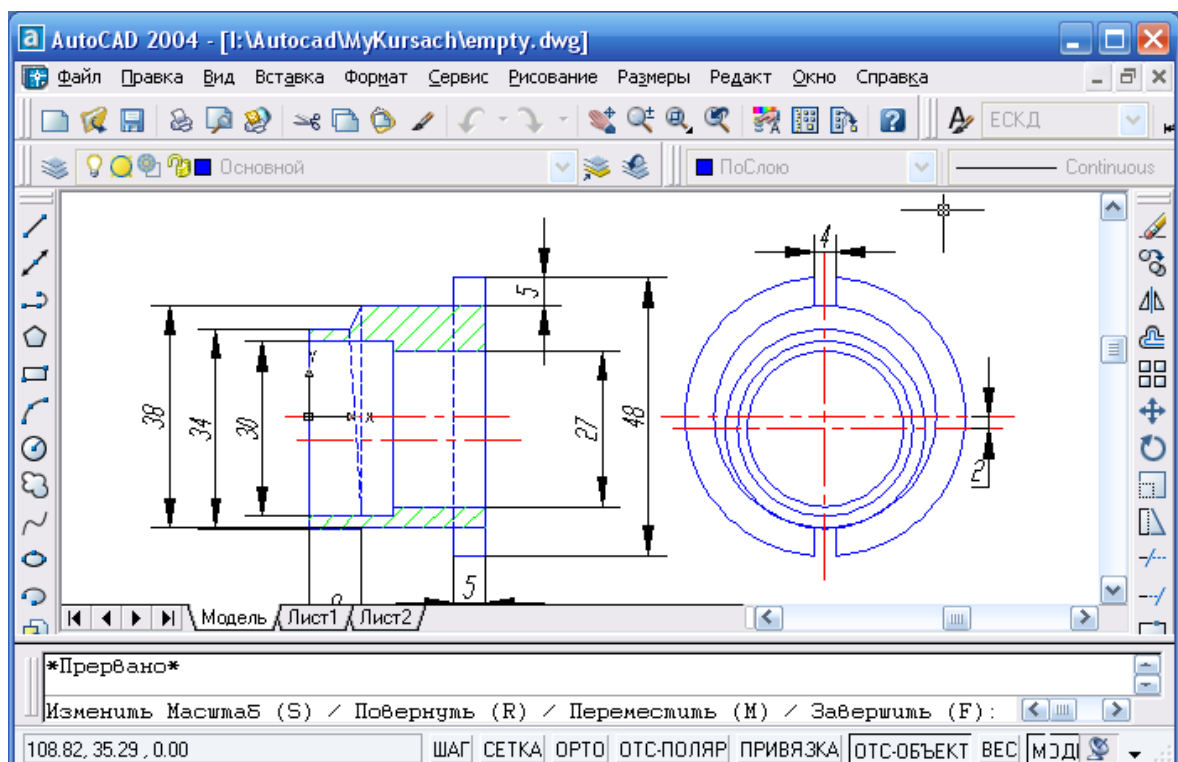


Рисунок 6 – Отображение чертежа в среде AutoCAD

Если что-то не устраивает, весь процесс также можно повторить снова-

ла.

Программисту при отладке или добавлении функционала может пригодиться функция RunDraw без параметров. Она рисует деталь с параметрами по умолчанию.

Если параметры рисования по умолчанию функции RunDraw не устраивают, можно Вызвать функцию DrawDetail, передав ей 12 параметров: d1 d2 d3 bd2 bd3 h bh tt b e n nformat. nformat - формат листа. Может принимать значения: radio1 (A4), radio2 (A3), radio3 (A2), radio4 (A1).

Функция set_fields используется для вычисления всех значений по d1. Функция dd_radio - получить формат листа. При использовании вне программы смысла не имеют.

4.2 Описание результатов

В итоге был реализован программный комплекс на языке AutoLISP, который может автоматизировать рисование эксцентрической оправы при различных параметрах детали. Есть возможность выбора формата листа для чертежа. Также, рисуется обрамление листа в соответствии с ЕСКД. При подготовке информации о детали запрашивается нужная информация у пользователя. Обеспечен удобный пользовательский интерфейс, позволяющий отменить неправильный ввод параметров и отображающий вычисленные программно параметры. Также предусмотрена возможность масштабирования детали, поворота её на угол, перемещение по полю чертежа.

Результаты выполнения программы в виде чертежей представлены в приложениях Г, Д и Е.

Заключение

В рамках данной курсовой работы были получены нужные знания, полезные только на территории СНГ, и лишь частично за рубежом, и навыки работы с проприетарной программой, которую рационально использовать только в странах СНГ, так как цена её порядка цены нового автомобиля.

Также была написана очень полезная программа на языке AutoLISP, хотя аналогичную можно было быстрее и проще написать в Delphi используя те же самые функции AutoCAD через программный API интерфейс AutoCAD.

Был получен бесценный опыт работы в системе AutoCAD как для автоматизации труда, так и для ведения компьютерной документации. Аналогичные программы были рассмотрены лишь теоретически. Хотя многие из них имеют схожий функционал и для аналогичных задач могут быть вполне успешно использованы.

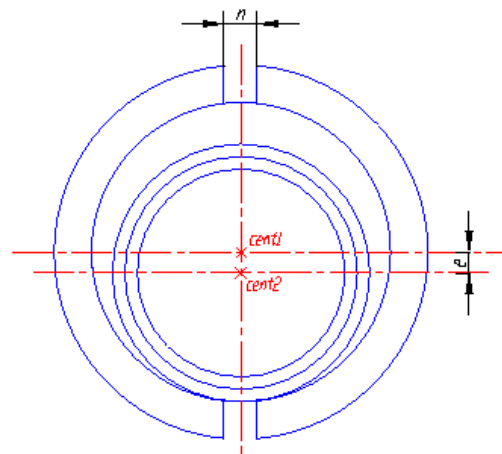
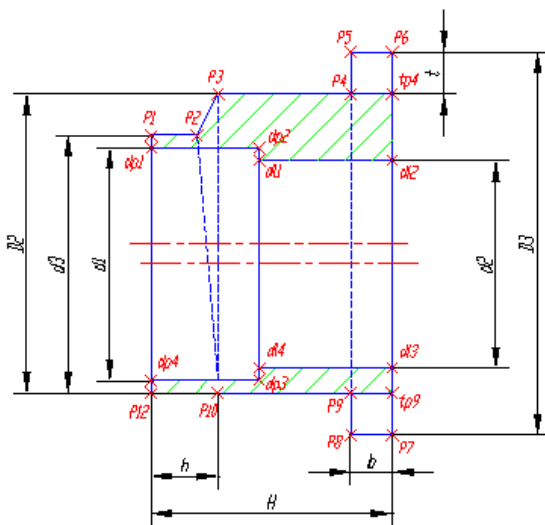
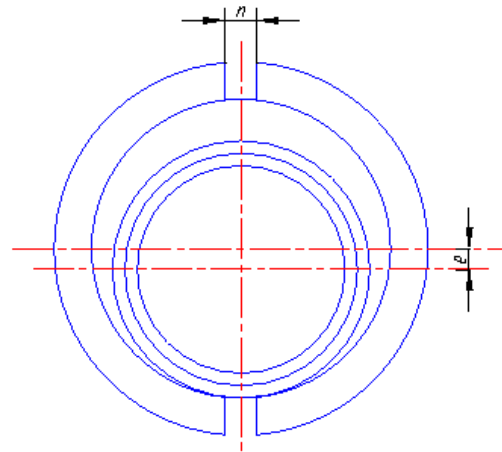
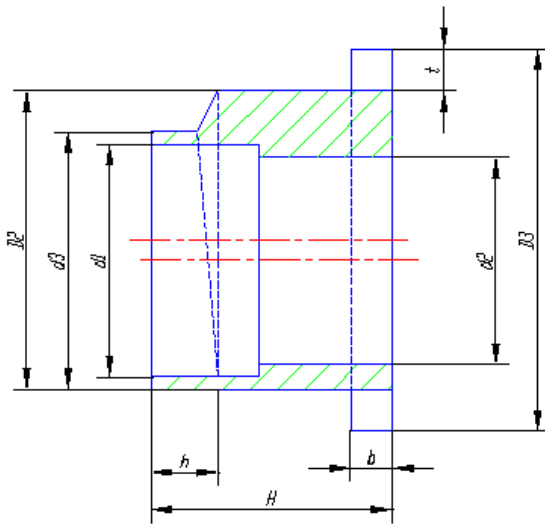
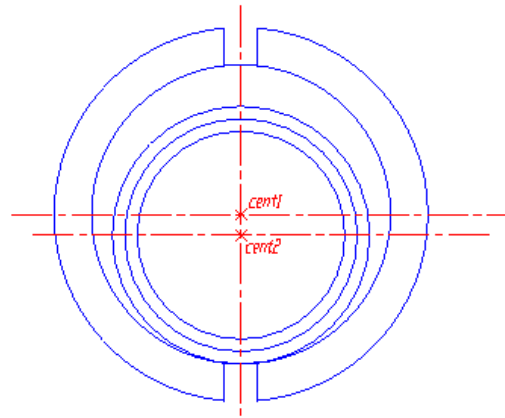
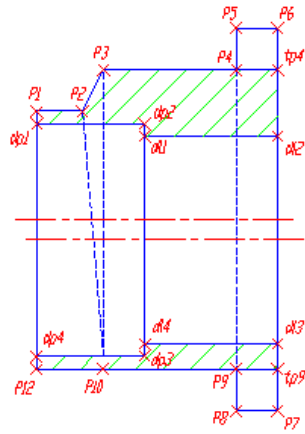
Во времена современности, когда OpenSource программы уже начали покорять мир, программные продукты стоимостью порядка нескольких современных компьютеров, вероятно, утратили свой приоритет. Однако, каждый инженер должен уметь работать с САПР. Пускай даже если полученные знания пригодятся только на теории и оставят неизгладимые отрицательные впечатления в душе человека.

Документация AutoLISP также оставляет желать лучшего. Русскоязычной документации вообще нет. Печатных книг по AutoLISP - единицы. Однако, это не показывает отрицательности AutoLISP и AutoCAD. На практике было прочувствовано, как обстоят дела с САПР на современном горизонте и увидели необъятные просторы для работы, и для создания чего-то лучшего. Борцы за свободный софт не спят и возможно уже в ближайшем времени у AutoCAD появится достойный OpenSource конкурент, снабжаемый энтузиастами большим количеством литературы, поддерживаемый в современном состоянии и удобный в использовании.

Список литературы

1. Малашенко В.С. Практическое пособие по курсу «САПР» для студентов специальности Т.03.01.00 – «Технология, оборудование и автоматизация». - Гомель: ГГТУ им. П.О. Сухого, 2000 – 36 с.
2. Толстоба Н.Д. Проектирование узлов оптических приборов учебное пособие. – СПб, 2002 – 128 с.
3. Норенков И. П. САПР. Часть 1. Принципы построения и структура. – Мн. : Выш. шк., 1987. – 123 с.
4. Бурькин А. В. Шаманы. Те, кому служат духи. – Санкт-Петербург : Петербургское Востоковедение, 2007 г. – 288 с.
5. Бугрименко Г. А. Автолисп – язык графического программирования в системе AutoCAD. – Москва : Машиностроение, 1992. – 144 с.
6. Кудрявцев Е. М. AutoLISP. Основы программирования в AutoCAD 2000 – Москва : ДМК Пресс, 2000. – 416 с.

Приложение А Слайды детали



Приложение Б Исходный код AutoLISP программы

```
; << Курсовой проект по дисциплине "Системы компьютерной  
графики" >>  
; Выполнил студент группы ИТ-22  
; Храбров Дмитрий а.к.а. DeXPeriX  
; ICQ: 606986  
; E-mail: dexperix[Ha]mail.ru  
; Jabber: dexperix@jabber.ru  
; Беларусь, Гомель, ГГТУ им. П. О. Сухого, 2007г.  
  
; Функция отвечает непосредственно за рисование детали на  
чертеже  
; соответствует 8 блоку графической схемы  
(defun DrawDetail (d1 d2 d3 bd2 bd3 h bh tt b e n nformat  
/ ee)  
; "настройка" AutoCAD перед началом выполнения  
(command "_ .erase" "_all" "")  
(command "_ .osnap" "_none")  
(setvar "DIMPOST" "")  
; параметр для выноски размеров и смещения 2 детали  
(setq ee 4)  
; основной контур. по часовой стрелке  
(setq  
p1 (list 0 (- (/ d3 2) e) )  
p2 (list (- h e) (NIH 1 p1) )  
p3 (list (+ (NIH 0 p2) e) (/ bd2 2) )  
p4 (list (- bh b) (NIH 1 p3) )
```

```

p5 (list (NIH 0 p4) (+ (NIH 1 p4) tt) )
p6 (list bh (NIH 1 p5) )
p7 (list (NIH 0 p6) (- (NIH 1 p6) ) )
p8 (list (NIH 0 p5) (- (NIH 1 p5) ) )
p9 (list (NIH 0 p4) (- (NIH 1 p4) ) )
p10 (list (NIH 0 p3) (- (NIH 1 p3) ) )
p11 (list (NIH 0 p10) (- (+ (/ d3 2) e) ) )
p12 (list (NIH 0 p1) (NIH 1 p11) )
)

```

;дополнительные контуры

```
(setq
```

```
    tp4 (list bh (NIH 1 p4) )
```

```
    tp9 (list bh (NIH 1 p9) )
```

```
)
```

```
(setq
```

```
    dp1 (list 0 (- (/ d1 2) e) )
```

```
    dp2 (list (+ h b) (NIH 1 dp1) )
```

```
    dp3 (list (NIH 0 dp2) (- (+ (/ d1 2) e) ) )
```

```
    dp4 (list (NIH 0 dp1) (NIH 1 dp3) )
```

```
    dp5 (list h (NIH 1 dp3) )
```

```
)
```

```
(setq
```

```
    dl1 (list (NIH 0 dp2) (- (/ d2 2) e) )
```

```
    dl2 (list (NIH 0 p6) (NIH 1 dl1) )
```

```
    dl3 (list (NIH 0 dl2) (- (+ (/ d2 2) e) ) )
```

```
    dl4 (list (NIH 0 dl1) (NIH 1 dl3) )
```

```
)
```

;данные для штриховки

```
(setq
```

```
    sh1 (list (NIH 0 dp2) (+ (NIH 1 dp2)
```

```

        (/ (- (NTH 1 p4) (NTH 1 dp2) ) 2)) )
sh2 (list (NTH 0 dp3) (+ (NTH 1 dp3)
        (/ (- (NTH 1 p9) (NTH 1 dp3) ) 2)) )
mashtab (/ (+ bd3 bh) 100.0)
ugol "0"
)
; данные для 2 проекции
(setq
  cent1 (list (+ bh (* 7 ee) (* 3 e) (/ bd3 2) ) 0 )
  cent2 (list (NTH 0 cent1) (- e) )
)
; данные для осей 2 проекции
(setq
  os11 (list (- (NTH 0 cent1) (/ bd3 2) (* 2 e) ) 0 )
  os12 (list (+ (NTH 0 cent1) (/ bd3 2) (* 4 e) ) 0
  )
  os21 (list (+ (NTH 0 os11) e) (- e) )
  os22 (list (- (NTH 0 os12) e) (- e) )
  os31 (list (NTH 0 cent1) (+ (/ bd3 2) ee) )
  os32 (list (NTH 0 os31) (- (NTH 1 os31) ) )
)
; данные для вынесения размеров 2 проекции
(setq
  dr11 (list (- (NTH 0 os12) (* 2 e) ) 0 )
  dr12 (list (NTH 0 dr11) (- e) )
  dr1r (list (NTH 0 dr11) 0)
  dr21 (list (- (NTH 0 os31) (/ n 2) ) (/ bd3 2) )
  dr22 (list (+ (NTH 0 os31) (/ n 2) ) (/ bd3 2) )
  dr2r (list (NTH 0 dr21) (+ (/ bd3 2) (* 2 e) ) )
)

```

; непосредственно рисование

```
(command "_layer" "_s" "ОСНОВНОЙ" "")  
(command "_linetype" "_s" "CONTINUOUS" "")  
(command "_pline" p1 p2 p3 p4 p5 p6  
                p7 p8 p9 p10 p11 p12 "_c")  
(command "_pline" dp1 dp2 dp3 dp4 "")  
(command "_line" dl1 dl2 "")  
(command "_line" dl3 dl4 "")  
(command "_line" p4 tp4 "")  
(command "_line" p9 tp9 "")
```

; непосредственное рисование 2 проекции

```
(command "_circle" cent1 (/ bd2 2))  
(setq bd2name (entlast) )  
(command "_circle" cent1 (/ bd3 2))  
(setq bd3name (entlast) )  
(command "_circle" cent2 (/ d1 2))  
(command "_circle" cent2 (/ d2 2))  
(command "_circle" cent2 (/ d3 2))  
  
(command "_line" (list (- (NTH 0 os31) (/ n 2) )  
                      (+ (NTH 1 os31) ee) )  
                (list (- (NTH 0 os32) (/ n 2) )  
                      (- (NTH 1 os32) ee) ) "" )  
(setq l1name (entlast) )  
(command "_line" (list (+ (NTH 0 os31) (/ n 2) )  
                      (+ (NTH 1 os31) ee) )  
                (list (+ (NTH 0 os32) (/ n 2) )  
                      (- (NTH 1 os32) ee) ) "" )  
(setq l2name (entlast) )
```

```
(command "_ .trim" l1name l2name ""  
  (list (NIH 0 os31) (- (/ bd3 2) ) ) "")
```

```
(command "_ .trim" bd2name bd3name ""  
  (list (- (NIH 0 os31) (/ n 2) )  
    (+ (/ bd3 2) ee) ) "")
```

```
(command "_ .trim" bd2name bd3name ""  
  (list (- (NIH 0 os31) (/ n 2) )  
    (- (+ (/ bd3 2) ee) ) ) "")
```

```
(command "_ .trim" bd2name bd3name ""  
  (list (- (NIH 0 os31) (/ n 2) )  
    (/ bd2 4) ) "")
```

```
(command "_ .trim" bd2name bd3name ""  
  (list (+ (NIH 0 os31) (/ n 2) )  
    (+ (/ bd3 2) ee) ) "")
```

```
(command "_ .trim" bd2name bd3name ""  
  (list (+ (NIH 0 os31) (/ n 2) )  
    (- (+ (/ bd3 2) ee) ) ) "")
```

```
(command "_ .trim" bd2name bd3name ""  
  (list (+ (NIH 0 os31) (/ n 2) )  
    (/ bd2 4) ) "")
```

```
(command "_ .trim" l1name l2name ""  
  (list (NIH 0 os31) (/ bd3 2) ) "")
```

;штриховка

```
(command "_ .layer" "_s" "Штриховка" "")
```

```

(command "_ .bhatch" "c" "ansi31" mashtab ugol sh1 "")
(command "_ .bhatch" "c" "ansi31" mashtab ugol sh2 "")
;дополнительные линии
(command "_ .linetype" "_s" "НЕВИДИМАЯ2" "")
(command "_ .layer" "_s" "ОСНОВНОЙ" "")
(command "_ .line" p4 p9 "")
(command "_ .line" dp5 p3 "")
(command "_ .line" dp5 p2 "")
;оси
(command "_ .layer" "_s" "Оси" "")
(command "_ .linetype" "_s" "ОСЕВАЯ2" "")
(command "_ .line" (list (- ee) 0) (list (+ bh ee) 0) "")
(command "_ .line" (list (- (* 0.5 ee)) (- ee) )
                  (list (+ bh (* 1.5 ee) ) (- ee)) "")
;оси для 2 проекции
(command "_ .line" os11 os12 "")
(command "_ .line" os21 os22 "")
(command "_ .line" os31 os32 "")
(command "_ .linetype" "_s" "CONTINUOUS" "")
;проставка размеров
(command "_ .layer" "_s" "Размеры" "")
;горизонтальные основной проекции
(command "_ .dimlinear" p6 tp4 (list (+ (* 2.5 ee) bh) 0)
)
;вертикальные основной проекции
(command "_ .dimlinear" p12 p10 (list 0 (- (- (/ bd2 2) )
                                           (* 4 ee) ) ) )
(command "_ .dimlinear" p12 p7 (list 0 (- (- (/ bd3 2) )
                                           (* 5 ee) ) ) )

```

```

(command "_ .dimlinear" p8 p7 (list 0 (- (- (/ bd3 2) )
                                     (* 2 ee) ) ) )

;дополнительной проекции
(command "_ .dimlinear" dr11 dr12 dr1r)
(command "_ .dimlinear" dr21 dr22 dr2r)

;Простановка диаметров
;задание приставки к размерам в виде знака диаметра
(setvar "DIMPOST" "%C<>")
(command "_ .dimlinear" dp1 dp4 (list (- (* 2 ee) ) 0) )
(command "_ .dimlinear" p1 p12 (list (- (* 4 ee) ) 0) )
(command "_ .dimlinear" p3 p10 (list (- (* 6 ee) ) 0) )
(command "_ .dimlinear" dl2 dl3 (list (+ (* 5 ee) bh) 0)
)
(command "_ .dimlinear" p6 p7 (list (+ (* 7 ee) bh) 0) )

;Все, кроме рамки нарисовано. Запоминаем примитивы,
чтобы потом
;можно было выполнить над ним "масштаб" и "повернуть"
(setq Рисунок (SSGET "X" ) )

;рамка
(command "_ .ATTDIA" "1")
;рассчитываю точки вставки
(setq A4x (+ (- -105 (* 5 ee) ) (/ (NTH 0 os12) 2) ) )
(setq A3x (+ (- -210 (* 5 ee) ) (/ (NTH 0 os12) 2) ) )
(setq A2x (+ (- -320 (* 5 ee) ) (/ (NTH 0 os12) 2) ) )
(setq A1x (+ (- -420 (* 5 ee) ) (/ (NTH 0 os12) 2) ) )
;непосредственное рисование рамок
(if(= nformat "radio1") (command "_ .insert" "A4"

```

```

                                ( list A4x -180) 1 1 0) )
( if(= nformat "radio2") (command "_insert" "A3"
                                ( list A3x -180) 1 1 0) )
( if(= nformat "radio3") (command "_insert" "A2"
                                ( list A2x -210) 1 1 0) )
( if(= nformat "radio4") (command "_insert" "A1"
                                ( list A1x -360) 1 1 0) )

```

;Выполнение дополнительных действий над объектом.

;Соответствует 9 блоку графической схемы

;Action

```
(setq doac "")
```

```
( WHILE (AND (/= doac "f") (/= doac "F") )
```

```

  ( SETQ doac ( GETSTRING "\nИзменить
Масштаб (S)
Повернуть (R)
Переместить (M)
Завершить (F): " ) )

```

```

( if ( OR ( = doac "м" ) ( = doac "М" )
        ( = doac "s" ) ( = doac "S" ) )

```

```
(progn
```

```
(setq s_scale (GETREAL "\n Масштаб: " ) )
```

```
; (command "_DIMSCALE" (s_scale) )
```

```
(command "_DIMSCALE" 1)
```

```
(command "_scale" Пучок "" "0,0" s_scale "")
```

```
(command "_DIMLFAC" (/ 1 s_scale) )
```



```

)
)

(if ( OR ( = doac "п" ) ( = doac "П" )
          ( = doac "r" ) ( = doac "R" ) )
  (progn
    (setq r_point (GETPOINT "\n Базовая точка: ") )
    ;(setq r_angle (GETREAL "\n Угол: ") )
    ;(command "_ .rotate" ПусуНоК "" r_point r_angle "")
    (command "_ .rotate" ПусуНоК "" r_point pause)
  )
)
)

```

```

(if ( OR ( = doac "m" ) ( = doac "M" ) )
  (command "_ .move" ПусуНоК "" "0,0" pause)
)
)
)

```

;Основная функция запуска программы без диалога

;(для тестирования рисования)

```

(defun RunDraw ()
  (DrawDetail 56 50 62 72 90 16 58 10 10 5 8 0)
)

```

;Установка остальных полей по 1

```

(defun set_fields(val / d1 d2 d3 bd2 bd3 h bh tt b e n)

```

```
(setq d1 (atof val) )
(setq d2 (* d1 0.89) )
(setq d3 (* d1 1.14) )
(setq bd2 (* d1 1.27) )
(setq bd3 (* d1 1.6) )
(setq h (* d1 0.3) )
(setq bh (* d1 1.01) )
(setq tt (* d1 0.16) )
(setq b (* d1 0.18) )
(setq e (* d1 0.07) )
(setq n (* d1 0.125) )
(setq d2 (rtos d2) )
(setq on_d2 d2)
(setq d3 (rtos d3) )
(setq on_d3 d3)
(setq bd2 (rtos bd2) )
(setq on_bd2 bd2)
(setq bd3 (rtos bd3) )
(setq on_bd3 bd3)
(setq h (rtos h) )
(setq on_h h)
(setq bh (rtos bh) )
(setq on_bh bh)
(setq tt (rtos tt) )
(setq on_tt tt)
(setq b (rtos b) )
(setq on_b b)
(setq e (rtos e) )
(setq on_e e)
(setq n (rtos n) )
```

```

(setq on_n n)
; установка значений в поля формы
(set_tile "d2edit" d2)
(set_tile "d3edit" d3)
(set_tile "bd2edit" bd2)
(set_tile "bd3edit" bd3)
(set_tile "hedit" h)
(set_tile "bhedit" bh)
(set_tile "tedit" tt)
(set_tile "bedit" b)
(set_tile "eedit" e)
(set_tile "nedit" n)
)

```

; Функция, выполняющаяся по нажатию кнопки "OK" в основном диалоге

```

(defun ok_tab ( / )
  (setq d1 (atoi on_d1)
        d2 (atoi on_d2)
        d3 (atoi on_d3)
        bd2 (atoi on_bd2)
        bd3 (atoi on_bd3)
        h (atoi on_h)
        bh (atoi on_bh)
        tt (atoi on_tt)
        b (atoi on_b)
        e (atoi on_e)
        n (atoi on_n)
  )
)
)

```

```

; Функция выбора формата листа
; соответствует 4 блоку графической схемы
(defun dd_radio ( / ret_value1 dcl_id what_next on_rad)
; функция, вызываемая по ОК (формирование списка данных на
  выход)
  (defun ok_radio_tab ( / )
    (setq ret_value1 on_rad)
  )
; загрузка файла-диалога
  (setq dcl_id (load_dialog "рамка.dcl"))
; проверка существования диалога, если не существует —
  возвращаем
; заведомо неправильное значение.
; Тогда рамка не будет рисоваться вообще
  (if (not (new_dialog "dd_radio" dcl_id)) (exit)
    (progn
; начальные установки переменных и элементов
      (set_tile "radio1" "1")
      (setq on_rad "radio1")
      (setq what_next 8)
; цикл
      (while (< 2 what_next)
        (action_tile "radio1" "(setq on_rad $key)")
        (action_tile "radio2" "(setq on_rad $key)")
        (action_tile "radio3" "(setq on_rad $key)")
        (action_tile "radio4" "(setq on_rad $key)")
        (action_tile "accept" "(done_dialog 1) (ok_radio_tab)"
          )
        (action_tile "cancel" "(done_dialog 1) (exit)"))

```

```

    (setq what_next (start_dialog))
  )
; выгрузка диалога из памяти
(unload_dialog dcl_id) ; Unload the DCL file
ret_value1
)
)
)

; Функция запуска визуального диалога. Она же основная
(defun run_dialog ( / ret_value1 dcl_id1 what_next on_ed2
  on_ed1)
; начало программы (блок 1)
  ( SETQ flag T ) ; блок 2 графической схемы
( WHILE flag ; блок 3 графической схемы
  ; узнаём формат листа
  (setq f_format (dd_radio) ) ; блок 4 графической схемы
  ; загрузка основного диалога
  (setq dcl_id1 (load_dialog "form.dcl"))
  ; проверка существования диалога, если не существует —
  выход
  (if (not (new_dialog "dd_param" dcl_id1))(exit))
; начальные установки переменных и элементов
(set_tile "d1edit" "")
(set_tile "d2edit" "")
(set_tile "d3edit" "")
(set_tile "bd2edit" "")
(set_tile "bd3edit" "")
(set_tile "hedit" "")

```

```

(set_tile "bredit" "")
(set_tile "tredit" "")
(set_tile "bredit7" "")
(set_tile "eedit" "")
(set_tile "nedit" "")
(setq on_d1 ""
      on_d2 ""
      on_d3 ""
      on_bd2 ""
      on_bd3 ""
      on_h ""
      on_bh ""
      on_tt ""
      on_b ""
      on_e ""
      on_n ""
      )
( SETQ flag1 T )
( WHILE flag1

(start_image "ff")
(slide_image 0 0 (dimx_tile "ff") (dimy_tile "ff") "
  screenshot" )
(end_image)
(setq what_next 8)
; цикл для обработки ввода данных пользователем.
; Соответствует 5 блоку графической схемы
(while (< 2 what_next)
  ; дух машины, мой бубен сильнее твоей тупости!!!
  (action_tile "d1edit" "(setq on_d1 $value)(set_fields

```

```

    $value)")
  (action_tile "d2edit" "(setq on_d2 $value)")
  (action_tile "d3edit" "(setq on_d3 $value)")
    (action_tile "bd2edit" "(setq on_bd2 $value)")
    (action_tile "bd3edit" "(setq on_bd3 $value)")
  (action_tile "hedit" "(setq on_h $value)")
  (action_tile "bhedit" "(setq on_bh $value)")
  (action_tile "bedit" "(setq on_b $value)")
  (action_tile "ttedit" "(setq on_tt $value)")
  (action_tile "eedit" "(setq on_e $value)")
  (action_tile "nedit" "(setq on_n $value)")
  (action_tile "accept" "(done_dialog 1) (ok_tab)")
    (action_tile "cancel" "(done_dialog 1) (exit)")
  (setq what_next (start_dialog))
)

```

*; Проверка введённых данных. Соответствует 6 блоку
графической схемы*

```
(if ( or ( <= bd2 d3) ( <= d3 d1 ) ( <= bh h) (< d1 1) )
```

```
(progn
```

*; Параметры заданы неверно — ошибка. Соответствует 7
блоку*

```

  (alert "Неверно заданы параметры" )
    (new_dialog "dd_param" dcl_id1)
  (set_tile "d1edit" on_d1)
  (set_tile "d2edit" on_d2)
  (set_tile "d3edit" on_d3)
  (set_tile "bd2edit" on_bd2)
  (set_tile "bd3edit" on_bd3)

```

```

(set_tile "hedit" on_h)
(set_tile "bhedit" on_bh)
(set_tile "bedit" on_b)
(set_tile "tedit" on_tt)
(set_tile "eedit" on_e)
(set_tile "nedit" on_n)
)
( progn
  (setq flag1 nil)
  ; рисование детали — запуск параметрического
  ; чертежа
  ; все данные корректны. Соответствует 8 блоку
  ; графической схемы
  (DrawDetail d1 d2 d3 bd2 bd3 h bh tt b e n
    f_format)
)
)
)
)
; ввод данных, необходимых для повторения всех действий
; соответствует 10 и 11 блокам графической схемы
( SETQ ans ( GETSTRING "\nПовторить<Д/Н>: " ) )
( SETQ flag ( OR ( = ans "Д" ) ( = ans "д" )
  ( = ans "Y" ) ( = ans "y" ) ) )
); конец WHILE
; выгрузка диалога из памяти
(unload_dialog dcl_id1)
( PRIN1 )
; конец — 12 блок графической схемы алгоритма
) ;Аминь!

```


Приложение В Листинг DCL-файла

```
// Диалог параметрического чертежа детали
dd_param :
dialog {
label = "Параметрический чертеж детали";
fixed_height = true;
: row {
:image {
key="ff";
color =0;
    fixed_height = true;
fixed_width = true;
aspect_ratio=0.5;
width=120;
height=30;
alignment = center;
}
:column {
:column{
: edit_box {
key = "d1edit";
value = "0";
fixed_height = true;
fixed_width = true ;
width =20 ;
label = "d1= ";
}
: edit_box {
key = "d2edit";
```

```

value = "0";
fixed_height = true;
fixed_width = true ;
width = 20 ;
label = "d2= ";
}
: edit_box {
key = "d3edit";
value = "0";
fixed_height = true;
fixed_width = true ;
width = 20 ;
label = "d3=";
}
: edit_box {
key = "bd2edit";
value = "0";
fixed_height = true;
fixed_width = true ;
width = 20 ;
label = "D2= ";
}
: edit_box {
key = "bd3edit";
value = "0";
fixed_height = true;
fixed_width = true ;
width = 20 ;
label = "D3= ";
}

```

```
: edit_box {
key = "hedit";
value = "0";
fixed_height = true;
fixed_width = true ;
width = 20 ;
label = "h=";
}

: edit_box {
key = "bhedit";
value = "0";
fixed_height = true;
fixed_width = true ;
width = 20 ;
label = "H=";
}

: edit_box {
key = "ttedit";
value = "0";
fixed_height = true;
fixed_width = true ;
width =20 ;
label = "t= ";
}

: edit_box {
key = "bedit";
value = "0";
fixed_height = true;
fixed_width = true ;
```

```

width =20 ;
label = "b= ";
}
: edit_box {
key = "eedit";
value = "0";
fixed_height = true;
fixed_width = true ;
width =20 ;
label = "e= ";
}
: edit_box {
key = "nedit";
value = "0";
fixed_height = true;
fixed_width = true ;
width =20 ;
label = "n= ";
}

}
: row {
ok_button;
cancel_button;

}
}
}
}

```

Приложение Г Результат (Параметрический чертеж А4)

Приложение Д Результат (Параметрический чертеж АЗ)

Приложение Е Результат (Параметрический чертеж А2)