

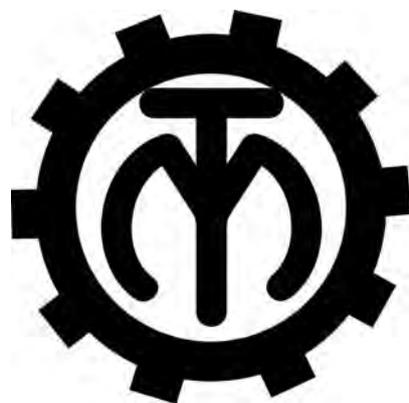
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

САПР РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.06 «Мехатроника и робототехника»
дневной формы обучения*

Часть 2



Могилев 2018

УДК 658.012.011.56
ББК 30.2-5-05
С 19

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «28» мая 2018 г.,
протокол № 11

Составители: канд. техн. наук, доц. М. Н. Миронова;
канд. техн. наук, доц. В. М. Шеменков;
С. Г. Черняков

Рецензент канд. техн. наук, доц. О. Е. Печковская

Методические рекомендации предназначены для выполнения лабораторных работ студентами направления подготовки 15.03.06 «Мехатроника и робототехника» дневной формы обучения.

Учебно-методическое издание

САПР РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

Часть 2

Ответственный за выпуск	В. М. Шеменков
Технический редактор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 46 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.

Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018



Содержание

1 Лабораторная работа № 9. Программирование технологической операции, выполняемой на токарном станке с ЧПУ	4
2 Лабораторная работа № 10. Программирование технологической операции, выполняемой на фрезерном станке с ЧПУ	8
3 Лабораторная работа № 11. Автоматизированное моделирование и программирование роботизированного технологического комплекса	10
4 Лабораторная работа № 12. Автоматизации расчета деталей узлов РТК в среде интегрированной системы программирования.....	14
5 Лабораторная работа № 13. Автоматизация параметрического черчения трехмерных геометрических моделей стандартных деталей.....	22
6 Лабораторная работа № 14. Автоматизированное геометрическое моделирование узлов РТК.....	36
Список литературы	41



1 Лабораторная работа № 9. Программирование технологической операции, выполняемой на токарном станке с ЧПУ

Цель работы: овладение практическими навыками автоматизированного проектирования управляющих программ для оборудования с ЧПУ с использованием современных САМ-систем.

Порядок выполнения первой части работы

Изучение проектных операций процедуры разработки управляющей программы для токарной обработки. Рассмотрим последовательность проектных операций процедуры разработки управляющей программы для токарной обработки с использованием модуля «LATHE» (токарная обработка) системы SprutCAM.

Предполагаемая последовательность переходов обработки рассматриваемой детали: подрезка правого торца; черновое точение наружного контура детали; чистовое точение наружного контура детали; точение канавок; нарезание резьбы; сверление отверстия.

Загрузить программу SprutCAM и учебник SprutCAM.

Из меню учебника SprutCAM (рисунок 1.1) выбрать раздел «Токарная обработка», «Обработка торца» (урок № 1).

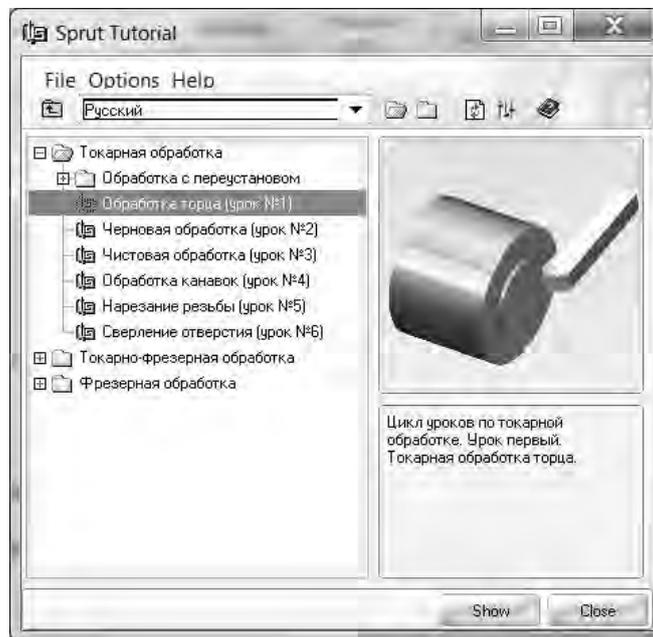
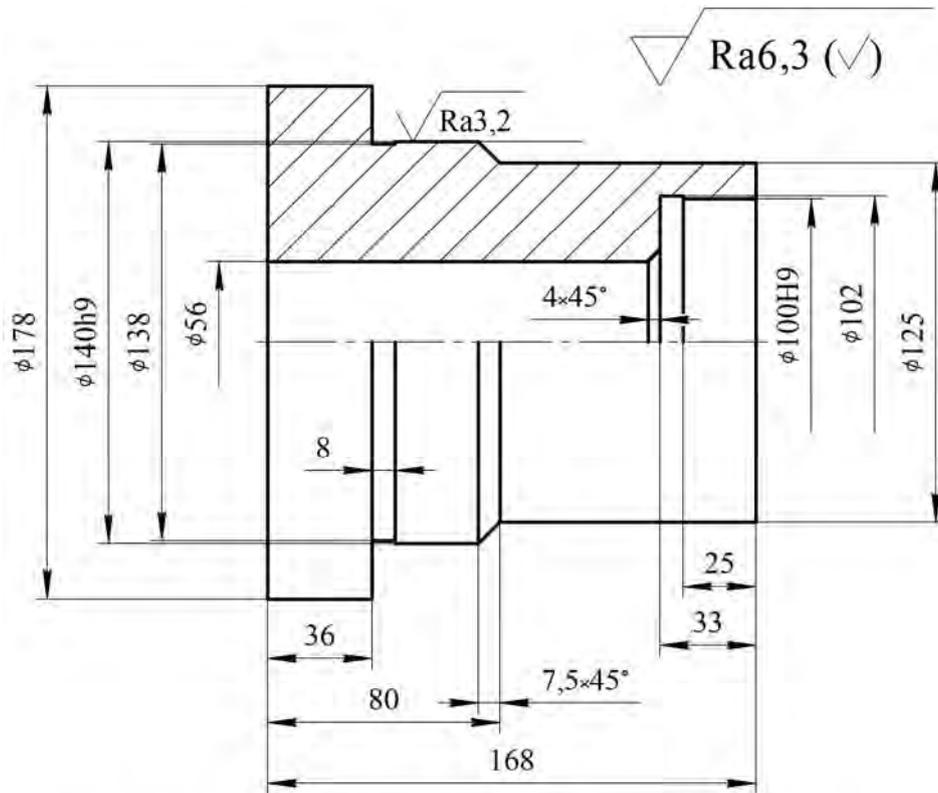


Рисунок 1.1 – Главное меню учебника SprutCAM

После выбора урока выполнить последовательно все предусмотренные в нем упражнения, перемещаясь по ним с помощью кнопок навигации. Завершив изучение упражнений урока № 1, перейти к уроку № 2 и выполнить остальные пять уроков.

Задание

С использованием САМ-системы SprutCAM спроектировать управляющую программу для токарной обработки детали, эскиз которой представлен на рисунке 1.2.



1. Общие допуски по ГОСТ 30893.1-2002: H14, h14, $\pm IT14/2$.
2. Остальные технические требования по СТБ 1014-95.

Рисунок 1.2 – Эскиз детали

Выполнение индивидуального задания состоит из следующих проектных процедур.

- 1 Создание 3D-модели детали и заготовки.
- 2 Задание технологической информации (определение последовательности технологических переходов, выбор инструментов и задание режимов резания).
- 3 Геометрическое моделирование технологических переходов и устранение возможных ошибок.
- 4 Генерация управляющей программы для станка с ЧПУ.

Содержание отчета по первой части

- 1 Задание и эскиз детали.
- 2 Эскиз профилей детали и заготовки с указанием зон обработки.
- 3 Краткое описание порядка выполнения работы.
- 4 Текст управляющей программы для станка с ЧПУ.

Контрольные вопросы

- 1 Назначение САМ-систем.
- 2 Каков общий порядок подготовки управляющих программ для оборудования с ЧПУ с использованием САМ-систем?
- 3 Какие существуют способы подготовки исходной геометрической информации об обрабатываемой детали?
- 4 Какая технологическая информация задается в процессе подготовки управляющей программы?

Порядок выполнения второй части работы

Во второй части будет рассмотрен порядок формирования программ для токарной обработки с использованием модуля «LATHE», который позволяет программировать следующие типы технологических переходов: подрезка торцов и отрезка; послойное черновое и черновое точение и растачивание наружных и внутренних поверхностей; точение и растачивание канавок; сверление осевых отверстий; сверление несоосных отверстий и фрезерование поверхностей на токарно-фрезерных обрабатывающих центрах; нарезание резьбы.

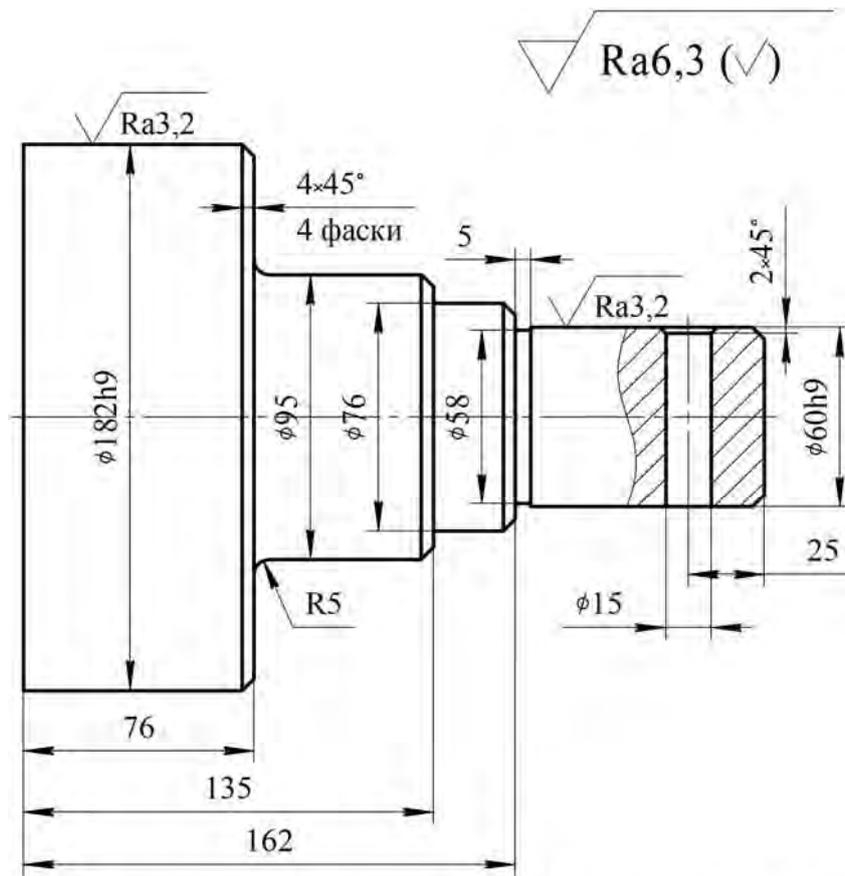
Порядок выполнения работы

- 1 Загрузить программу SprutCAM.
- 2 Загрузить учебник «Учебник SprutCAM» на рабочем столе Windows.
- 3 Из меню «Учебник SprutCAM» выбрать раздел «Токарно-фрезерная обработка», «Обработка хвостовика».
- 4 Выполнить последовательно все предусмотренные в нем упражнения, перемещаясь по ним с помощью кнопок навигации.
- 5 С использованием САМ-системы SprutCAM спроектировать управляющую программу для обработки детали (рисунок 1.3) на токарно-фрезерном обрабатывающем центре.

Содержание отчета по второй части

- 1 Задание и эскиз детали.
- 2 Эскиз профилей детали и заготовки с указанием зон обработки.
- 3 Краткое описание порядка выполнения работы.
- 4 Текст управляющей программы для станка с ЧПУ.





1. Общие допуски по ГОСТ 30893.1-2002: Н14, h14, ±IT14/2.
2. Остальные технические требования по СТБ 1014-95.

Рисунок 1.3 – Эскиз детали

Контрольные вопросы

- 1 Области применения САМ-систем.
- 2 Основные типы САМ-систем.
- 3 Программирование каких типов металлорежущих станков возможно с использованием САМ-системы SprutCAM?
- 4 Каковы отличительные особенности САМ-системы SprutCAM?
- 5 Что такое симуляция? Для чего используется этот процесс?
- 6 Что означает термин «постпроцессирование»?

2 Лабораторная работа № 10. Программирование технологической операции, выполняемой на фрезерном станке с ЧПУ

Цель работы: овладение практическими навыками автоматизированного проектирования управляющих программ для фрезерных станков с ЧПУ с использованием современных САМ-систем.

Порядок выполнения работы

- 1 Загрузить программу SprutCAM двойным щелчком по ярлыку SprutCAM на рабочем столе Windows.
- 2 Загрузить учебник двойным щелчком по ярлыку «Учебник SprutCAM» на рабочем столе Windows.
- 3 Из меню «Учебник SprutCAM» (рисунок 2.1) выбрать раздел «Фрезерная обработка», «2D-обработка», «Построение 2D-контуров».
- 4 После выбора урока выполнить последовательно предусмотренные в нем два упражнения «Построение 2D-контуров» и «Обработка 2D-контуров». Для перемещения по разделам использовать кнопки навигации.

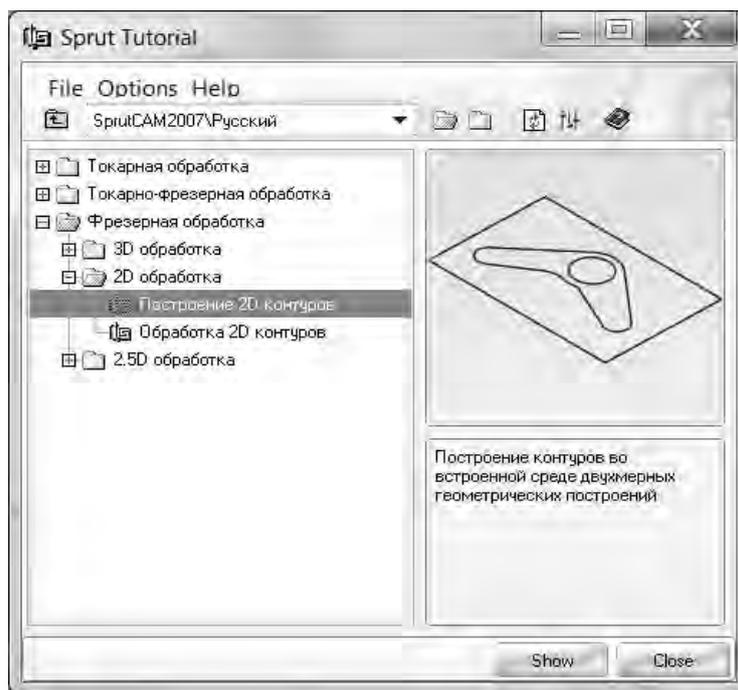


Рисунок 2.1 – Выбор урока из главного меню учебника по SprutCAM

5 После выполнения всех упражнений урока «2D-обработка» выполнить упражнения урока «2,5D-обработка».

6 С использованием САМ-системы SprutCAM спроектировать управляющую программу для фрезерной обработки детали, эскиз которой представлен на рисунке 2.2. Размеры поверхностей приведены в таблице 2.1.

Индивидуальное задание выполняется согласно заданному варианту.

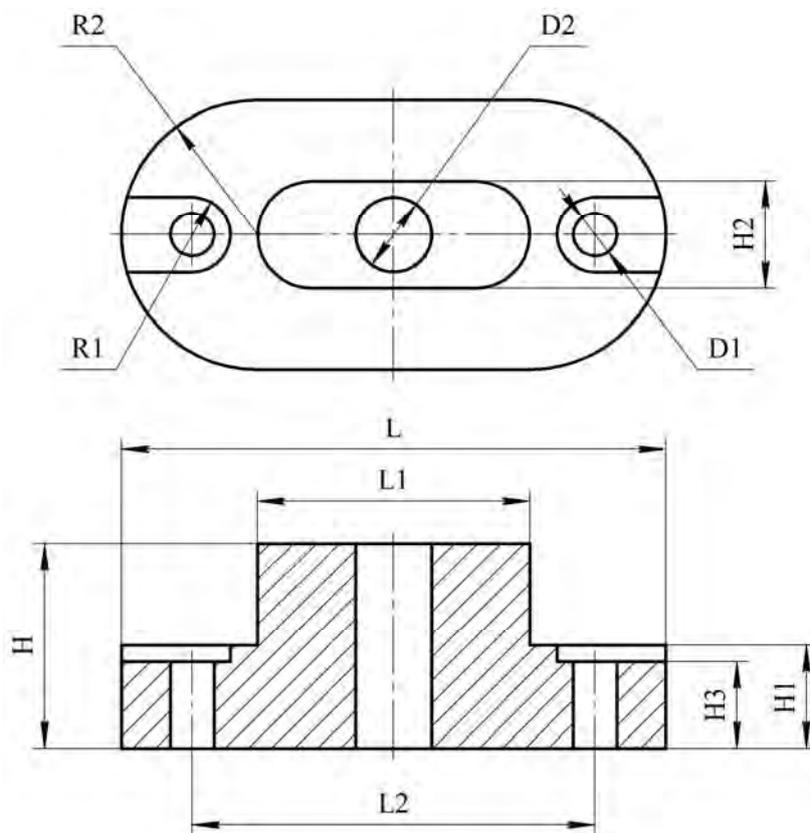


Рисунок 2.2 – Эскиз детали

Таблица 2.1 – Размеры поверхностей детали

Номер варианта	Значение размеров, мм											
	H3	H	L	L2	R1	R2	R3	D1	D2	H1	H2	L1
1	16	38	100	74	7	25	10	8	14	Произвольно		
2	24	57	150	111	10,5	37,5	15	12	21	Произвольно		

Порядок выполнения индивидуального задания

- 1 Проанализировать чертеж детали (см. рисунок 2.2). Выделить контуры заготовки и обрабатываемых поверхностей и присвоить им имена.
- 2 В режиме «2D» построить контуры заготовки и обрабатываемых поверхностей в плоскости X–Y.
- 3 Определить последовательность переходов фрезерной операции.
- 5 Задать технологические параметры переходов фрезерной операции (режущий инструмент, режимы резания, подвод, отвод инструмента, стратегию).
- 6 Выполнить генерацию программы в виде последовательности команд языка CLDATD.
- 7 Выбрать режим «Постпроцессор». Установить заданную систему ЧПУ и сгенерировать управляющую программу.

Содержание отчета

- 1 Задание и эскиз детали.
- 2 Эскиз профилей детали и заготовки с указанием зон обработки.
- 3 Краткое описание порядка выполнения работы.
- 4 Текст управляющей программы для станка с ЧПУ.

Контрольные вопросы

- 1 Назначение САМ-систем.
- 2 Каков общий порядок подготовки управляющих программ для оборудования с ЧПУ с использованием САМ-систем?
- 3 Какие существуют способы подготовки исходной геометрической информации об обрабатываемой детали?
- 4 Какая технологическая информация задается в процессе подготовки управляющей программы?

3 Лабораторная работа № 11. Автоматизированное моделирование и программирование роботизированного технологического комплекса

Цель работы: изучение методики создания моделей роботизированных технологических комплексов на основе цифровых прототипов основного и вспомогательного технологического оборудования и их офлайн-программирования на языке RAPID.

Методика выполнения работы

1 Создание компоновки РТК на основе цифровых прототипов технологического оборудования. Загрузить систему RobotStudio и создать РТК на базе шаблона IRB 2600_12kg_1.8m.

Установить силовой блок и блок контроллера IRC5 в пространстве станции на безопасном расстоянии от робота.

Из раздела меню «Home» командой «Import Library» из библиотеки цифровых прототипов «Equipment» на площадку РТК установить станок (файл Lathe.rslib). Командой «Set Position» определить положение станка с координатами $X = 1400$, $Y = 1152$, $Z = 0$, угол поворота вокруг оси $Z - 180^\circ$.

Командой «Import Library» из библиотеки цифровых прототипов «Equipment» загрузить файл схвата Gripper.rslib. Для точной установки схвата на фланец робота на вкладке «Layout» выбрать узел «Gripper» и из контекстного меню выполнить команду «Attach to...». В открывшемся списке нажать пункт меню «IRB 2600_12_1.8». Перемещением и вращением изображения РТК оценить точность установки схвата.



Для имитации секции шагового конвейера создать параллелепипед с размерами $4000 \times 200 \times 500$ мм. Из раздела меню «Modeling» командой «Solid» → «Box» построить параллелепипед, начиная из точки с координатами $X = 800$, $Y = -1000$, $Z = 0$ по заданным размерам $Length = 200$, $Width = 4000$, $Height = 500$. В закладке «Layout» на дереве проекта узел «Part_1» переименовать в «Conveyor».

Из библиотеки цифровых прототипов «Geometry» командой «Import Geometry» загрузить файл модели кассеты Cartridge.sat. Переместить модель кассеты на конвейер командой «Set Position», задавая координаты $X = 850$, $Y = 110$, $Z = 500$, угол поворота вокруг оси $X - 90^\circ$. Чтобы изображения конвейера и картриджа не сливались, перекрасить их командой «Set Color» из контекстного меню узлов «Conveyor» и «Cartridge» дерева проекта.

Аналогичным образом загрузить файл модели заготовки Rolling.sat и разместить заготовку в кассете с координатами $X = 885$, $Y = 100$, $Z = 550$, угол поворота вокруг оси $X - 90^\circ$.

Для создания второй загрузочной позиции на шаговом конвейере командой «Import Geometry» загрузить файл модели кассеты Cartridge.sat, а затем переместить модель кассеты на конвейер командой «Set Position», задавая координаты $X = 850$, $Y = 500$, $Z = 500$, угол поворота вокруг оси $X = 90^\circ$.

Аналогичным образом загрузить модель второй заготовки Rolling.sat и разместить заготовку в кассете с координатами $X = 885$, $Y = 490$, $Z = 550$, угол поворота вокруг оси $X - 90^\circ$. Заготовка должна разместиться между крайними упорами кассеты.

2 Создание опорных точек траектории для создания пути схвата. Для упрощения задачи определения координат при задании опорных точек (Target) пути схвата робота (Path) создать систему координат рабочего объекта Workobject для заготовки. Повернуть станцию в графическом окне так, чтобы ее конвейер был ориентирован, как показано на рисунке 3.1. Для точного определения системы координат объекта включить привязку к центру окружности «Snap Center» в верхней части графического окна системы и выбрать на закладке «Layout» узел «Rolling».

На вкладке «Home» в подразделе «Path Programming» нажать кнопки «Other» → «Create WorkObject». В появившемся окне в поле «Name» присвоить создаваемому рабочему объекту имя «Wobj_Rolling». В поле «Position X, Y» раздела меню «User Frame» нажать справа кнопку выпадающего списка, установить курсор в первую позицию поля «Position» (mm) для ввода координаты X (см. рисунок 3.1). На изображении заготовки, используя объектную привязку, найти центр торца и нажать левой клавишей мыши. Для завершения процедуры нажать кнопки «Accept», а затем «Create».

На основе компоновки РТК и размеров конвейера, картриджа, заготовки (рисунок 3.2) направления их осей координат определить координаты опорных точек (Target) пути схвата при загрузке и выгрузке заготовки со станка (таблица 3.1).



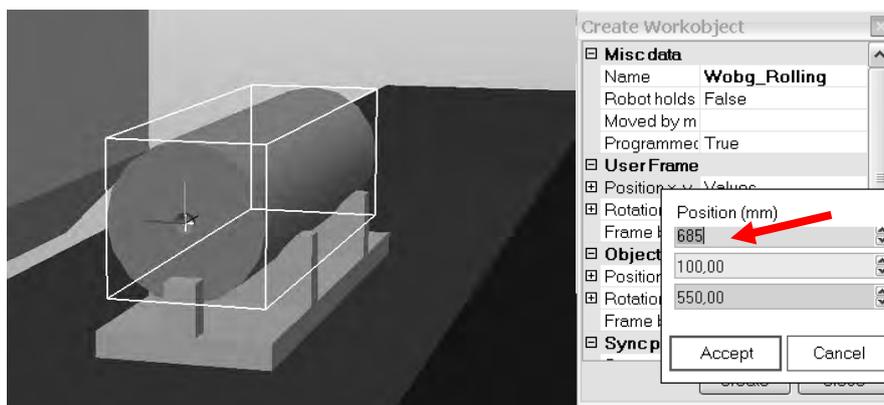


Рисунок 3.1 – Положение заготовки при создании системы координат Workobject

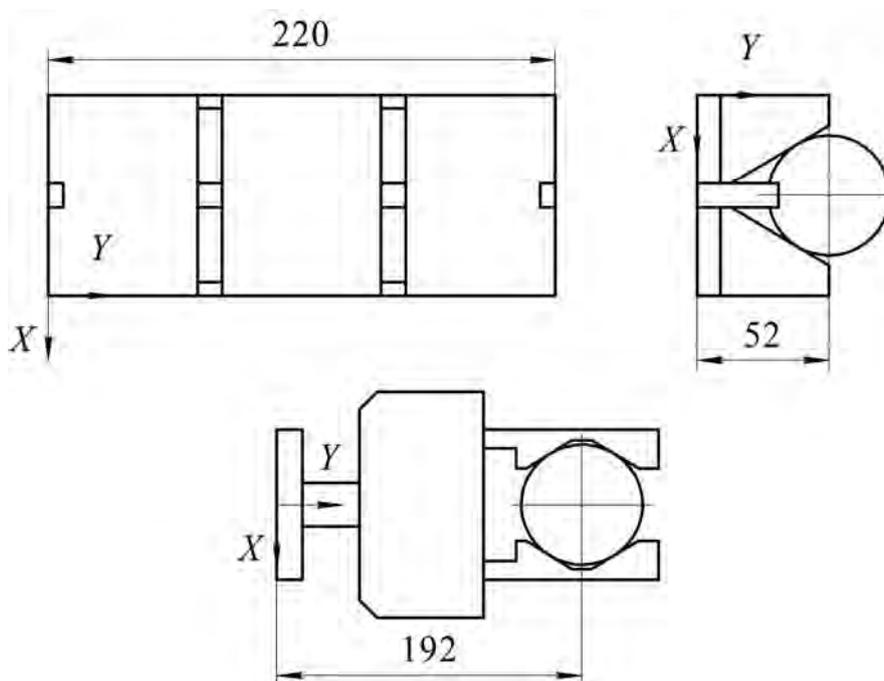


Рисунок 3.2 – Положение заготовки при создании рабочей системы координат

Таблица 3.1 – Координаты опорных точек пути схвата

Имя точки	Значение координаты в Wobj_Rolling, мм			Угол поворота вокруг оси объекта Wobj_Rolling, град		
	X	Y	Z	Rx	Ry	Rz
Target_10	0	-100	192	180	0	180
Target_20	0	-100	282	180	0	180
Target_30	720	-100	545	180	0	180
Target_40	620	-100	595	180	0	180
Target_50	20	-100	595	180	0	180

Для задания точек траектории выбрать рабочий объект «Wobj_Rolling» на закладке «Path&Targets», открыть вкладку «Home» главного меню системы. В разделе «Path Programming» нажать кнопки «Target» → «Create Target».

Вверху окна ввода данных «Create Target» в поле «Reference» из выпадающего списка выбрать в поле ссылку на системы координат рабочих объектов «WorkObject». Внизу окна в поле «WorkObject» выбрать рабочий объект «Wobj_Rolling», в системе координат которого определены опорные точки. Заполнить поля «Position» значениями координат целевой точки «Target_10». Нажать кнопку «Add» – «Добавить точку». После задания координат всех пяти точек нажать кнопку «Create». Закрыть окно «Create Target» нажатием кнопки «Close».

3 Создание пути схвата робота. Для создания пути схвата по ранее определенным точкам нажать в разделе меню «Path Programming» кнопки «Path» → «Empty Path».

На дереве путей и точек «Paths&Targets» в разделе «Path» появляется пустой путь «Path_10». Добавить в пустой путь «Path_10» целевые точки. Для этого выделить в меню «Paths&Targets» первую целевую точку «Target_20». В контекстном меню выбранной точки нажать пункт меню «Add to path» → «Path 10» → «<First>».

Создать полный путь командой «Add to path»→«Path 10»→«<Last>», добавляя точки в последовательности, показанной на рисунке 3.3.

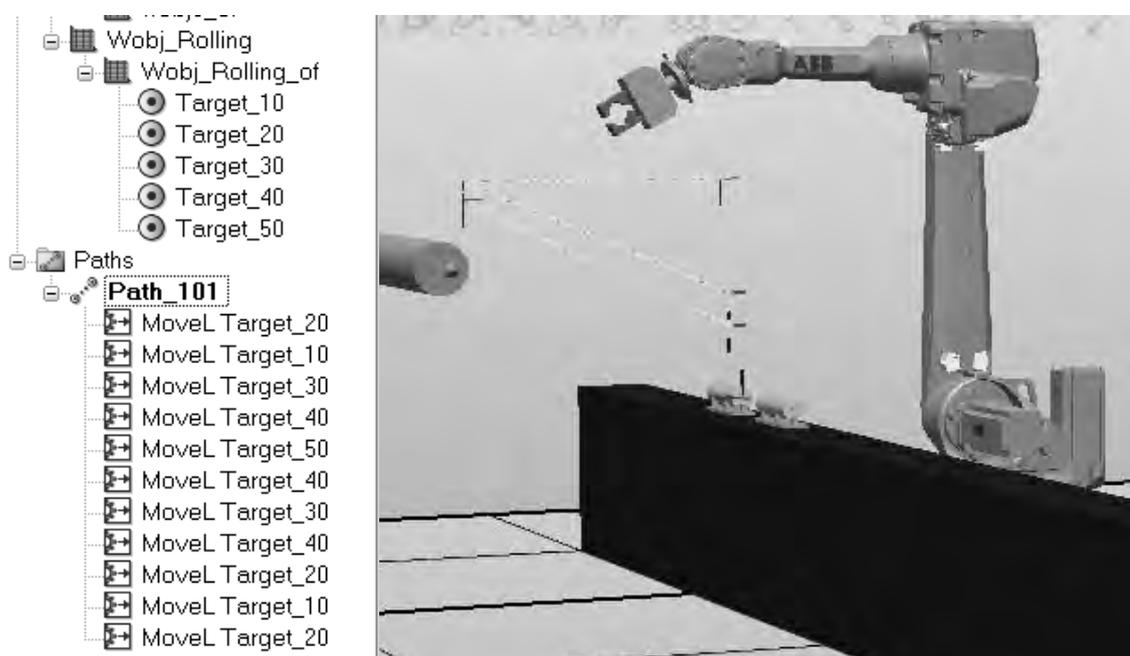


Рисунок 3.3 – Создание пути схвата Path_10

Для конфигурирования звеньев руки робота в каждой точке пути на дереве проекта выбрать путь «Path_10» и из контекстного меню выполнить команду «Auto Configuration».

Для проверки созданного пути инструмента выбрать вкладку «Simulation»,

раздел меню «Configure» и нажать пункт меню «Simulation Setup». Появляется окно «Setup Simulation». В списке «Available Procedures» выделить путь «Path_10» и нажать на стрелку влево, а затем кнопки «Apply» и «OK».

В результате выполнения этих команд будет сформирована Rapid-программа, управляющая движением робота вдоль заданной траектории.

Для моделирования движения робота в меню «Simulation Control» нажать кнопку  «Play». В графическом окне системы следует наблюдать, как робот перемещается по заданному пути.

Содержание отчета

- 1 Цель работы.
- 2 Текст Rapid-программы, управляющей движением робота вдоль заданной траектории.
- 3 Выводы.

Контрольные вопросы

- 1 Для чего предназначена программа RAPID?
- 2 Из каких частей состоит программа RAPID?
- 3 Какие функции содержит программирование высокого уровня RAPID?
- 4 Назовите специфические особенности языка RAPID.

4 Лабораторная работа № 12. Автоматизации расчета деталей узлов РТК в среде интегрированной системы программирования

Цель работы: изучение методики параметрического синтеза деталей и узлов РТК в среде САД-систем.

Программирование в среде САД-систем, таких как AutoCAD, Inventor и др., является мощным инструментарием для создания специализированных систем автоматизированного проектирования. Начиная с версии 2.15 в среду AutoCAD была интегрирована система программирования AutoLISP, а начиная с версии 14 – система программирования Microsoft Visual Basic for Applications (VBA).

Visual Basic – наиболее популярный контроллер автоматизации, поэтому в AutoCAD в качестве контроллера автоматизации используется Microsoft Visual Basic for Applications (VBA), который присутствует и в пакете приложений Microsoft Office.

VBA не рассматривается как замена AutoLISP или Visual LISP, а как альтернативное решение, дополняющее их.

Средства ActiveX Automation для AutoCAD стали доступны начиная с версии 14 (1997 г.). Большинство управляющих элементов ActiveX представляют собой серверы автоматизации, а любое приложение, поддерживаю-



щее VBA, может выступать в роли клиента автоматизации.

Клиенты и серверы автоматизации. Автоматизация (ранее известная как OLE-автоматизация – OLE Automation) – это одно из наиболее важных средств технологии ActiveX, позволяющее программно управлять объектами из других приложений. Это основное средство, с помощью которого можно интегрировать функциональные возможности различных приложений.

Приложения, поддерживающие автоматизацию, делятся на две категории: клиенты автоматизации и серверы автоматизации. Причем некоторые приложения могут быть только клиентами либо только серверами автоматизации, но есть и такие (к ним относится Microsoft Excel, Microsoft Access, AutoCAD), которые могут выступать и в том и в другом качестве.

При интеграции двух приложений одно предоставляет свои объекты для использования, а другое использует объекты первого приложения.

Приложение, объекты которого доступны для других приложений, называется **сервером автоматизации** (иногда его еще называют компонентом).

Приложение, которое использует объекты другого приложения, называется **клиентом автоматизации**. Объекты, которые доступны для других приложений, называют **объектами автоматизации**. Так, AutoCAD является служебным приложением по отношению к VBA или Visual Basic, которые действуют в качестве контроллера автоматизации.

Через объекты автоматизации приложение-сервер открывает доступ другим программам к тем своим функциям, которые могут быть им полезны. Например, текстовый редактор может открывать другим программам доступ к проверке орфографии, служба связи – доступ к созданию и отправке сообщений. Это позволяет разработчикам ускорить процесс создания своих приложений благодаря использованию готовых функций сервера.

Для программного управления объектом автоматизации из любого приложения AutoCAD или Microsoft Office необходимо:

- установить ссылку на библиотеку объектов приложения-сервера автоматизации;
- создать объект нужного класса;
- осуществить все необходимые действия над объектом, обращаясь к его свойствам и методам;
- закрыть объект.

Но можно использовать объекты сервера автоматизации, не устанавливая ссылку на конкретную библиотеку объектов, однако в этом случае программа будет работать медленнее. Это связано с тем, что после установки ссылки на библиотеку, при объявлении объектной переменной, можно указать ее конкретный тип, если же ссылка отсутствует, то для этой переменной необходимо указать обобщенный тип – Object.

Microsoft Excel как сервер автоматизации. Чтобы создать объект для использования в операциях автоматизации, нужно сначала создать экземпляр его класса и присвоить ссылку на него объектной переменной. Однако создать можно не любой объект из объектной модели приложения-сервера, а только глобальные объекты. Таким глобальным объектом для всех объектных моделей



приложений семейства Microsoft Office является объект Application, который находится на вершине иерархии объектов.

Существует несколько способов создания экземпляра класса объекта.

С помощью ключевого слова *New*. Это ключевое слово описывает переменную типа Object и одновременно устанавливает ссылку на новый экземпляр класса объекта.

```
Dim app As New Excel.Application
```

При использовании ключевого слова *New* для создания нового экземпляра класса Application запускается соответствующее приложение (например, Word или Excel). Если это приложение уже запущено, то, чтобы не запускать второй экземпляр, лучше использовать для создания нового экземпляра класса функцию **GetObject ()**.

Чтобы использовать ключевое слово *New* для создания экземпляра класса объекта автоматизации, необходимо предварительно добавить ссылку библиотеки объектов приложения сервера в диалоговом окне редактора VBA Tools (Инструменты + References (Ссылки)).

Пример использования

```
Sub ExcelOpenFile_Click()
On Error GoTo Err_ ' анализ ошибки открытия приложения
Dim strAppName As String
Dim app As New Excel.Application ' создать экземпляр класса
strAppName = CurrentProject.Path & "\Книга1.xls" ' путь к файлу электронной таблицы
With app
.Visible = True ' включаем видимость Excel
.Presentations.Open strAppName ' открываем файл Книга1.xls
End With
Set app = Nothing ' освобождаем переменную
Exit_
Exit Sub
Err_:
MsgBox Err.Description
Resume Exit_
End Sub
```

С помощью функции *CreateObject()*. Функция **CreateObject ()** создает ссылку на новый экземпляр класса автоматизированного объекта. Значение, возвращаемое этой функцией, должно быть присвоено объектной переменной с помощью оператора *Set*. Например, названным оператором открывается приложение Excel и ссылка на него присваивается объектной переменной. При этом даже если Excel уже был открыт, будет запущен еще один его экземпляр:

```
Set app = CreateObject ("Excel.Application")
```



Объектная переменная в данном случае может быть объявлена как Object, и тогда ссылку на библиотеку объектов Microsoft Word устанавливать не надо. А может быть объявлена так:

```
Dim app As Excel.Application
```

В этом случае должна быть обязательно установлена ссылка на библиотеку объектов Microsoft Excel *.0 Object Library (** показывает, что для разных версий офиса разные значения: 10.0, 11.0, 12.0 – для офисов XP, 2003, 2007 соответственно). Если на компьютере установлено несколько версий Microsoft Office, то можно указать номер версии приложения, которое будет использоваться при автоматизации, например:

```
Set app = CreateObject("Excel.Application.11")
```

Пример использования функции

```
Sub ExcelOpenFile()
On Error GoTo Err_
Dim strAppName As String
Dim app As Excel.Application
strAppName = CurrentProject.Path & "\Книга1.xls" ' указываем путь к файлу Книга1.xls
Set app = CreateObject("Excel.Application") ' создать экземпляр класса
With app
.Visible = True ' «проявляем» Excel
.Workbooks.Open strAppName ' открываем файл Книга1.xls
End With
Set app = Nothing ' закрываем переменную
Exit_:
Exit Sub
Err_:
MsgBox Err.Description
Resume Exit_
End Sub
```

С помощью функции *GetObject()*. Этот способ используется для того, чтобы получить ссылку на уже существующий экземпляр класса, т. е. для доступа к существующим документам, хранящимся в файлах, или для доступа к объекту Application уже запущенного приложения-сервера. Функция *GetObject()* имеет следующий синтаксис:

```
Set <объектнаяПеременная> = GetObject([<путь>][,<класс>])
```

Обязательно должен быть указан хотя бы один из аргументов функции!

Рассмотрим три варианта написания функции:

1) `Set app = GetObject("Excel.Application")` – здесь опускается первый параметр (тогда функция работает подобно двум вышеуказанным) – просто для создания экземпляра класса объекта;



2) `Set app = GetObject("C:\Baze\ Книга1.xls", "Excel.Application")` – здесь указаны оба параметра – путь к файлу и имя приложения. В этом случае запускается файл `Книга1.xls`;

3) `Set app = GetObject("C:\Baze\ Книга1.xls")` – здесь второй аргумент может быть опущен, если в реестре Windows существует связка расширения файла с объектом приложения. Например, в случае Microsoft Excel файлы `xls` по умолчанию имеют объектный тип `Excel.Workbook`. Это означает, что если не указывать аргумент `<класс>`, то будет создана ссылка на объект `Workbook`, представленный указанным в первом аргументе файлом.

Пример использования функции

```
Sub ExelOpenFile()
On Error GoTo Err_
Dim strAppName As String
Dim app As Excel.Application
    strAppName = CurrentProject.Path & "\Книга1.doc"
    Set app = GetObject("", "Excel.Application")
app.Visible = True
    Set app = Nothing
Exit Sub
Err_:
    MsgBox Err.Description
Resume Exit_
End Sub
```

При выполнении процедуры параметрического синтеза поршня пневмоцилиндра (рисунок 4.1) табличный процессор Excel является сервером автоматизации, т. к. он выполняет задачи определения параметров отдельных конструктивных элементов поршня. AutoCAD является клиентом автоматизации, т. к. он использует эти данные для геометрического синтеза конструкции поршня.

Методика выполнения работы

1 Скопировать в свою рабочую папку файл Microsoft Excel `Pr2_ПневмоцилиндрVBA.xls`.

2 Создать в своей рабочей папке файл AutoCAD под именем `Практика2.dwg`.

3 Выполнить ассоциативную загрузку AutoCAD (двойной «клик» мышкой по значку файла в своей рабочей папке).

4 Открыть редактор VBA клавиатурной командой `ALT + F11`.

5 Набрать текст программы приведенный ниже.

6 Выполнить программу клавиатурной командой `F5` (при наличии ошибок в программе устранить их).

7 Оценить правильность вычислений параметров поршня, просмотрев файл экспорта `PE.txt` и сравнив результаты с контрольными данными.

8 Составить отчет по результатам работы.



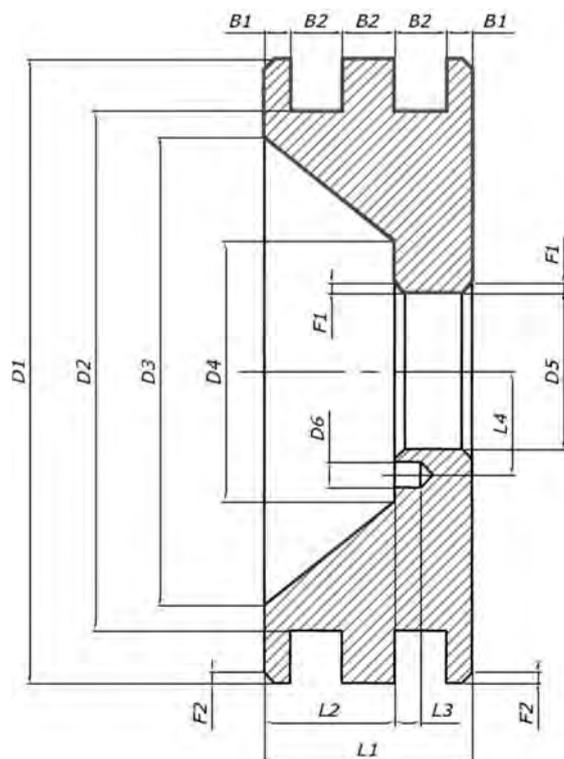


Рисунок 4.1 – Поршень пневмоцилиндра

Текст программы для выполнения работы

Public Sub PPSPor()

' ИСХОДНЫЕ ДАННЫЕ

Dim P1 As Integer ' Сила трения при контакте уплотнения с гильзой пневмоцилиндра, Н

Dim P2 As Integer ' Усилие при зажиме или транспортировании, Н

Dim P3 As Integer ' Вес поршня и перемещаемых частей привода, Н

Dim Pm As Integer ' Минимальное абсолютное давление в магистрали, Па (Н/м²)

Dim Pa As Long ' Атмосферное давление (101 325 Па)

Dim P0 As Integer ' Сила предварительного сжатия пружины, (Н)

Dim Pn As Integer ' Жесткость пружины, (Н/м)

Dim S As Integer ' Ход поршня,(М)

' ВЫХОДНЫЕ ДАННЫЕ (в мм.)

Dim Dp As Single ' Расчетный диаметр поршня

Dim D1 As Integer ' Принятый диаметр поршня

Dim Gk As Single ' Глубина канавок для уплотнений

Dim D2 As Single ' Диаметр дна канавок

Dim B1 As Single ' Ширина крайней щеки поршня

Dim B2 As Single ' Ширина канавок для уплотнений

Dim L1 As Single ' Длина поршня

Dim Ds As Integer ' Диаметр штока

Dim D5 As Single ' Диаметр отверстия для крепления штока

Dim Hg As Single ' Высота гайки

Dim Hs As Single ' Высота шайбы

Dim L2 As Integer ' Глубина поднутрения поршня

Dim M As Integer ' Диаметр резьбы гайки

```
Dim D6 As Integer ' Диаметр отверстия под лапку стопорной шайбы
Dim L3 As Single ' Глубина отверстия под лапку стопорной шайбы
Dim L4 As Integer ' Межосевое расстояние центрального отверстия и отверстия под лапку
стопорной шайбы
Dim D3 As Integer ' Большой диаметр поднутрения поршня
Dim D4 As Integer ' Меньший диаметр поднутрения поршня
Dim F1 As Integer ' Ширина фаски в центральном отверстии поршня
Dim F2 As Integer ' Ширина фаски на наружной поверхности поршня
```

```
Dim appExcel As Object
```

```
' Присваивает объектной переменной appExcel ссылку на файл Microsoft Excel
"ПневмоцилиндрVBA.xls"
' и тем самым обеспечивает доступ к листам и ячейкам электронной таблицы
Set appExcel = GetObject("Полный путь к рабочей папке\ПневмоцилиндрVBA.xls")
```

```
' ЧТЕНИЕ ИСХОДНЫХ ДАННЫХ
```

```
P1 = appExcel.Worksheets("Поршень").Range("B4").Value
P2 = appExcel.Worksheets("Поршень").Range("B5").Value
P3 = appExcel.Worksheets("Поршень").Range("B6").Value
Pm = appExcel.Worksheets("Поршень").Range("B7").Value
Pa = appExcel.Worksheets("Поршень").Range("B8").Value
P0 = appExcel.Worksheets("Поршень").Range("B10").Value
Cn = appExcel.Worksheets("Поршень").Range("B11").Value
S = appExcel.Worksheets("Поршень").Range("B12").Value
```

```
'Вычисление диаметра поршня
```

```
Dp = 1.13 * Sqr(P1 + P2 + P3 + P0 + cp * S)
```

```
'Запись расчетного диаметра поршня в ячейку электронной таблицы
```

```
appExcel.Worksheets("Поршень").Range("B16").Value = Dp
```

```
'Подбор ближайшего стандартного значения диаметра поршня по ГОСТ 15608-70
```

```
D1 = appExcel.Worksheets(" ГОСТ 15608 – 70 ").Range("B20").Value
```

```
'Запись принятого диаметра поршня в ячейку электронной таблицы
```

```
appExcel.Worksheets("Поршень").Range("B17").Value = Dp
```

```
'Выбор глубины канавки для уплотнения поршня
```

```
Gk = appExcel.Worksheets("Уплотнение поршня").Range("I26").Value
```

```
'Определение диаметра дна канавок для уплотнения поршня
```

```
D2 = D1 - Gk * 2
```

```
'Выбор ширины канавки для уплотнения поршня
```

```
B2 = appExcel.Worksheets("Уплотнение поршня").Range("I26").Value
```

```
'Ширина правой и левой щек поршня
```

```
B1 = Round(B2 / 1.5)
```

```
'Определение длины поршня
```

```
L1 = Round(B1 + B2 + B2 + B2 + B1)
```

```
'Подбор стандартного значения диаметра штока по ГОСТ 15608-70
```

```
Ds = appExcel.Worksheets(" ГОСТ 15608 – 70 ").Range("B22").Value
```

```
'Определение диаметра отверстия в поршне для крепления штока
```

```
D5 = Round(Ds / 2)
```

```
'Выбор размера резьбы гайки
```

```
M = appExcel.Worksheets("Гайки").Range("C27").Value
```

```
'Выбор высоты гайки
```

```
Hg = appExcel.Worksheets("Гайки").Range("D27").Value
```



```

'Выбор высоты шайбы
Hs = appExcel.Worksheets("Шайбы").Range("K30").Value
'Выбор диаметра отверстия под лапку стопорной шайбы
D6 = appExcel.Worksheets("Шайбы").Range("M30").Value
'Выбор глубины отверстия под лапку стопорной шайбы
L3 = appExcel.Worksheets("Шайбы").Range("N30").Value
'Определение межосевого расстояния центрального отверстия и
'отверстия под лапку стопорной шайбы
L4 = appExcel.Worksheets("Шайбы").Range("L30").Value
'Определение глубины поднутрения поршня для размещения
'элементов крепления штока
L2 = Round(Hg + Hs + 0.5 * Hg)
'Определение меньшего диаметра D4 поднутрения поршня
D4 = Round((L4 + D6) * 2)
'Определение большего диаметра D3 поднутрения поршня
D3 = Round(D1 - Gk * 2)
'Определение ширины фаски на наружной поверхности поршня
F2 = Round(B2 / 4)
'Определение ширины фаски в центральном отверстии поршня
F1 = appExcel.Worksheets("Гайки").Range("C27").Value
'Экспорт результатов параметрического синтеза поршня пневмоцилиндра в файл EP.txt
Open " Полный путь к рабочей папке \EP.txt" For Output Access Write As #1
Write #1, D1 '40 Контрольные значения
Write #1, D2 '33.5
Write #1, B2 '4.5
Write #1, B1 '3
Write #1, L1 '20
Write #1, D5 '6
Write #1, L2 '5
Write #1, D6 '4
Write #1, L4 '7
Write #1, L3 '5
Write #1, D4 '22
Write #1, D3 '34
Write #1, F1 '1
Write #1, F2 '1
Close #1
'Отключает объектную переменную appExcel и
закрывает связь с электронными таблицами
Set appExcel = Nothing
End Sub

```

Содержание отчета

- 1 Цель работы.
- 2 Текст программы параметрического моделирования поршня.
- 3 Выводы.



Контрольные вопросы

- 1 Для чего предназначен табличный процессор Excel при разработке программы параметрического моделирования?
- 2 Что позволяет создать функция CreateObject ()?
- 3 Какие функции содержит программирование высокого уровня?
- 4 Какой программный продукт является клиентом автоматизации при параметрическом проектировании?

5 Лабораторная работа № 13. Автоматизация параметрического черчения трехмерных геометрических моделей стандартных деталей

Цель работы: изучение методики параметрического черчения трехмерных геометрических моделей стандартных деталей в среде CAD-систем.

Программное обеспечение

- 1 Операционная система Windows XP.
- 2 Система управления базами данных Microsoft Access.
- 3 Пакет программ трехмерного геометрического моделирования AutoCAD.
- 4 Интегрированная система программирования Microsoft VBA.

В области проектирования автоматизированных производств объектами автоматизированного проектирования могут быть 3D-модели инструментальных наладок станков с ЧПУ или компоненты технологического оснащения роботизированных технологических комплексов, которые используются в среде таких систем, как SprutCAM, EdgeCAM, Cimatron, RobotStudio, RoboCAD и др.

Стандартные элементы конструкций средств автоматизации не требуют решения задачи параметрического синтеза. Их параметры определены соответствующими стандартами.

Создание библиотеки цифровых прототипов стандартных конструктивных элементов, которые часто встречаются в средствах автоматизации механо-сборочных операций, которые отличаются размерами, состоит из следующих этапов:

- 1) создание базы данных параметров стандартных деталей средств автоматизации;
- 2) создание макроса VBA для выбора и экспорта из базы данных параметров стандартных изделий;
- 3) геометрическое моделирование стандартного объекта на основе выбранных параметров.



Методика создания базы данных параметров стандартных деталей средств автоматизации

Таблицей Access является совокупность данных, объединенных общей темой. Для каждого объекта назначается отдельная таблица, чтобы не было повторений в данных (избыточности данных). Таблицы состоят из записей (строк) и полей (столбцов). Количество полей записи определяется на стадии проектирования таблицы, поэтому прежде чем создавать таблицу с помощью приложения Access, необходимо четко представлять ее структуру.

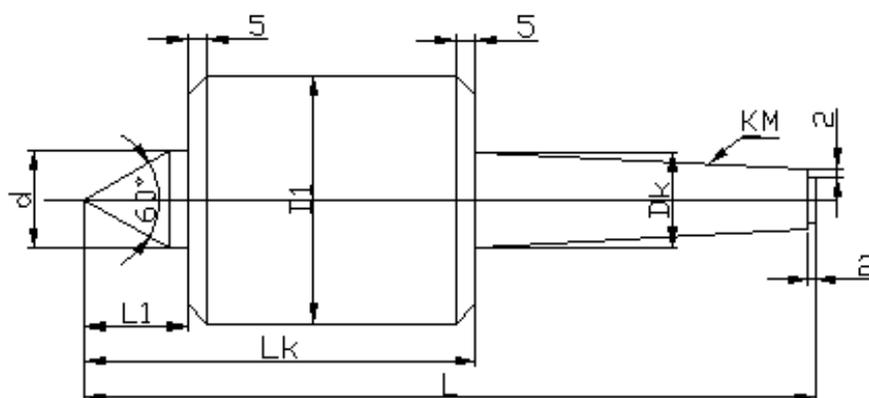
Рассмотрим создание таблицы базы данных вращающихся центров по ГОСТ 8745–75.

Исходная таблица стандарта ГОСТ 8742–75 *Центры вращающиеся для станков с ЧПУ* имеет следующий вид (таблица 5.1).

Таблица 5.1 – Параметры центров вращающихся по ГОСТ 8745–75

Конус Морзе	Серия	d	D	L	l	D_1	l_1
		Не более				Не менее	
2	Нормальная	22	56	160	90	56	24
3		25	63	180	94	63	26
4		28	71	210	101	71	30
5		32	80	240	104	80	34

Для создания базы данных уточнить состав параметров, необходимых для создания геометрической модели вращающегося центра (рисунок 5.1).



Конус Морзе	d	L	L_k	D_1	L_1	Kon
2	22	160	90	56	24	0,04995
3	25	185	99	63	26	0,05020
4	28	225	116	71	30	0,05194
5	32	260	124	80	34	0,05263

Рисунок 5.1 – Уточненные параметры центров вращающихся по ГОСТ 8745–75

В среде любой САД-системы создать параметризованный рисунок объекта и сохранить его в формате файла **.bmp**.

Загрузить СУБД Microsoft Access. После загрузки СУБД Microsoft Access и выполнения команд **Файл + Создать + Новая база данных** открывается окно создания таблиц (рисунок 5.2).

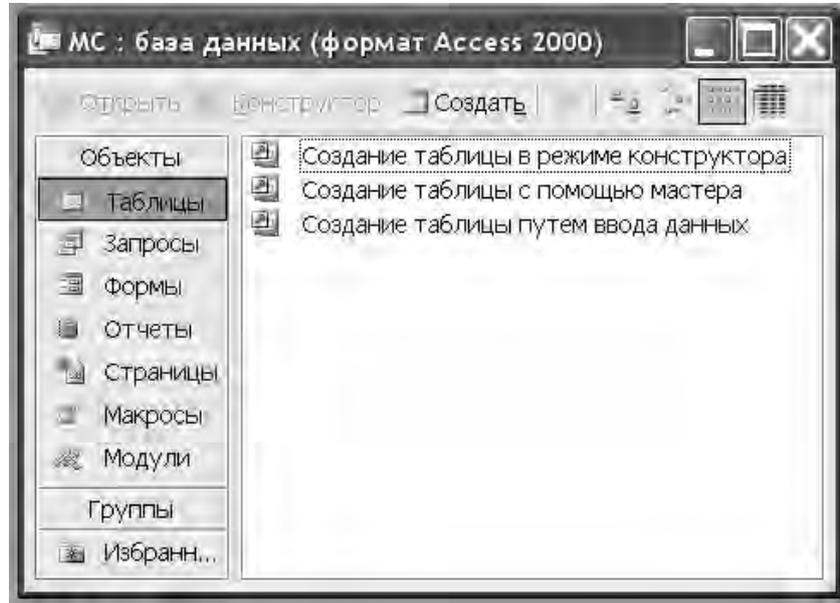


Рисунок 5.2 – Окно базы данных для выбора способа создания таблицы

Из трех возможных вариантов создания таблицы базы данных выбрать вариант **Создание таблицы в режиме конструктора**. Заполнить форму конструктора таблиц, присваивая имена и определяя типы данных полей в соответствии с таблицей стандарта (рисунок 5.3).

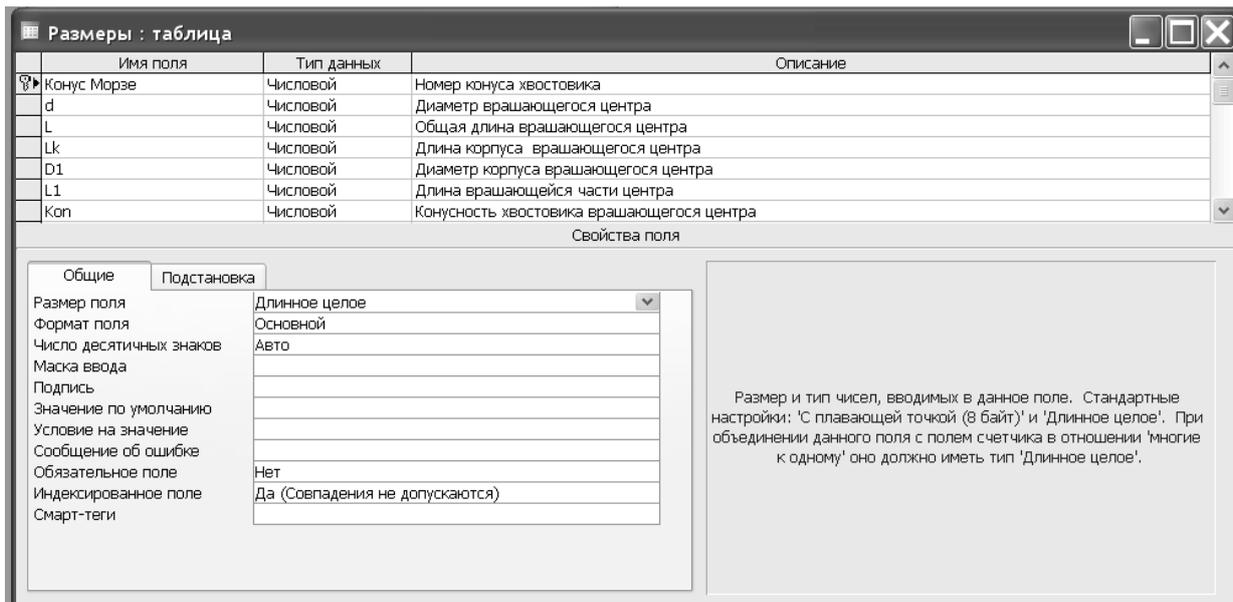


Рисунок 5.3 – Окно создания таблицы базы данных

Закрывать окно таблицы и сохранить таблицу под именем **Размеры**, предварительно определив поле **Конус Морзе** как ключевое. В окне базы данных (БД) открыть сохраненную таблицу и заполнить ее в соответствии с уточненными данными (рисунок 5.4).

Конус Морзе	d	L	Lk	D1	L1	Kon
2	22	160	90	56	24	0,04995
3	25	185	99	63	26	0,05020
4	28	225	116	71	30	0,05194
5	32	260	124	80	34	0,05263

Рисунок 5.4 – Окно таблицы БД в режиме ввода данных

Для организации процедуры экспорта данных, выбранных в таблице БД, создать форму командой **Формы** в окне базы данных (см. рисунок 5.2). Форма будет служить для просмотра таблицы, выбора и выполнения команды экспорта данных.

Для создания формы следует воспользоваться командой **Создание формы в режиме конструктора**. После выбора способа создания формы нажать кнопку **Создать** (рисунок 5.5).

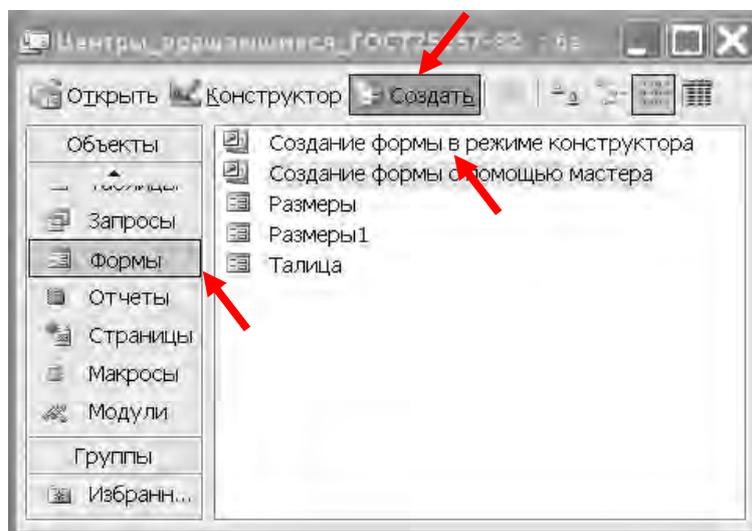


Рисунок 5.5 – Окно выбора способа создания формы

В появившемся окне **Новая форма** создать пустую форму без задания в качестве источника данных таблицы. Выбрать пункт **Конструктор** (рисунок 5.6) и нажать кнопку **ОК**.

На следующем шаге в открывшемся окне **Форма** нажать левой кнопкой мыши по полю формы и из контекстного меню выполнить команду **Заголовок/Примечание** формы для создания заголовка формы, в котором будет размещен рисунок детали и ссылка на стандарт (рисунок 5.7).

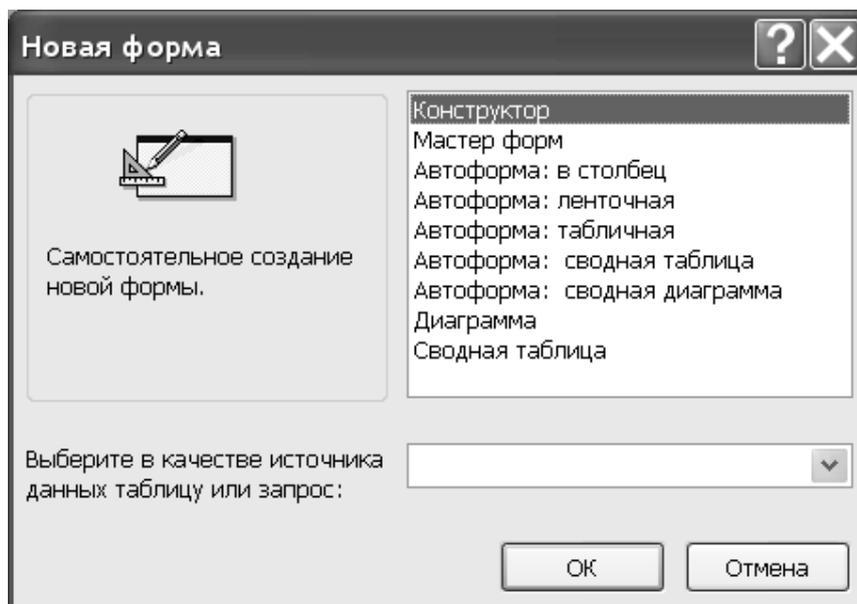


Рисунок 5.6 – Окно создания формы

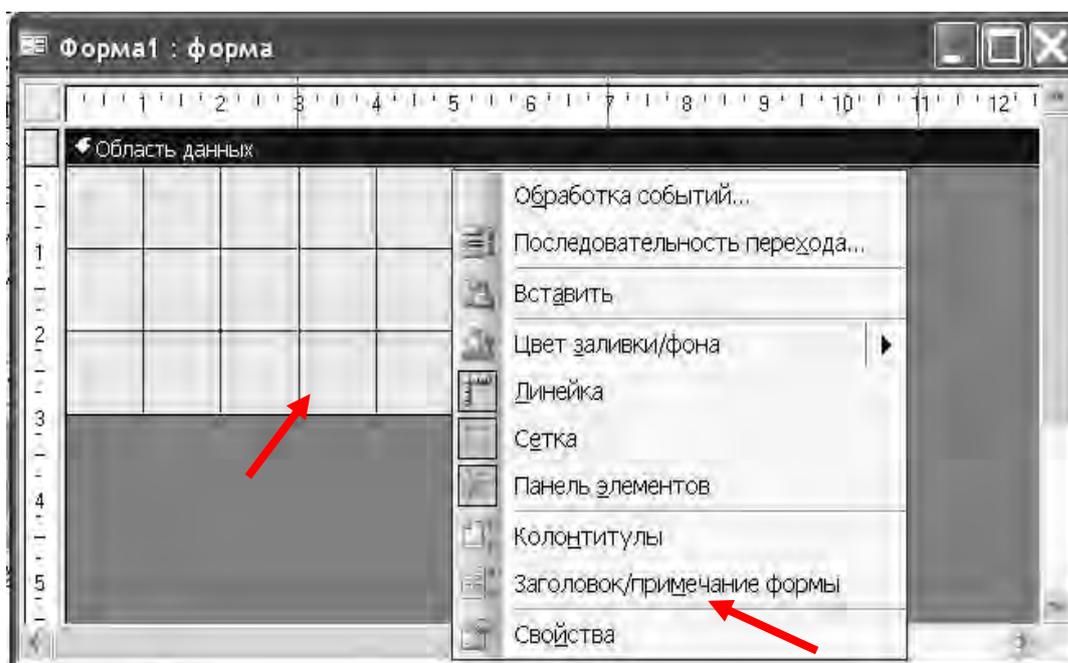


Рисунок 5.7 – Создание заголовка формы

На панели инструментов (рисунок 5.8) выбрать кнопку **Надпись** , нажать ее и вставить поле надписи в область заголовка формы. В созданном поле надписи записать название формы **Центры вращающиеся для станков с ЧПУ (ГОСТ 8742–75)**. Аналогичным образом создать подзаголовок **Тип А Исполнение 1**.

Ниже надписей вставить созданный ранее рисунок через буфер обмена или командой панели инструментов **Рисунок** . Для этого на панели инструментов нажать кнопку **Рисунок** , с помощью проводника Windows найти файл рисунка и выполнить команду **Открыть**.

Для выбора из базы данных требуемого типоразмера стандартной детали в области данных формы создать поле списка командой **Список** (рисунок 5.9).

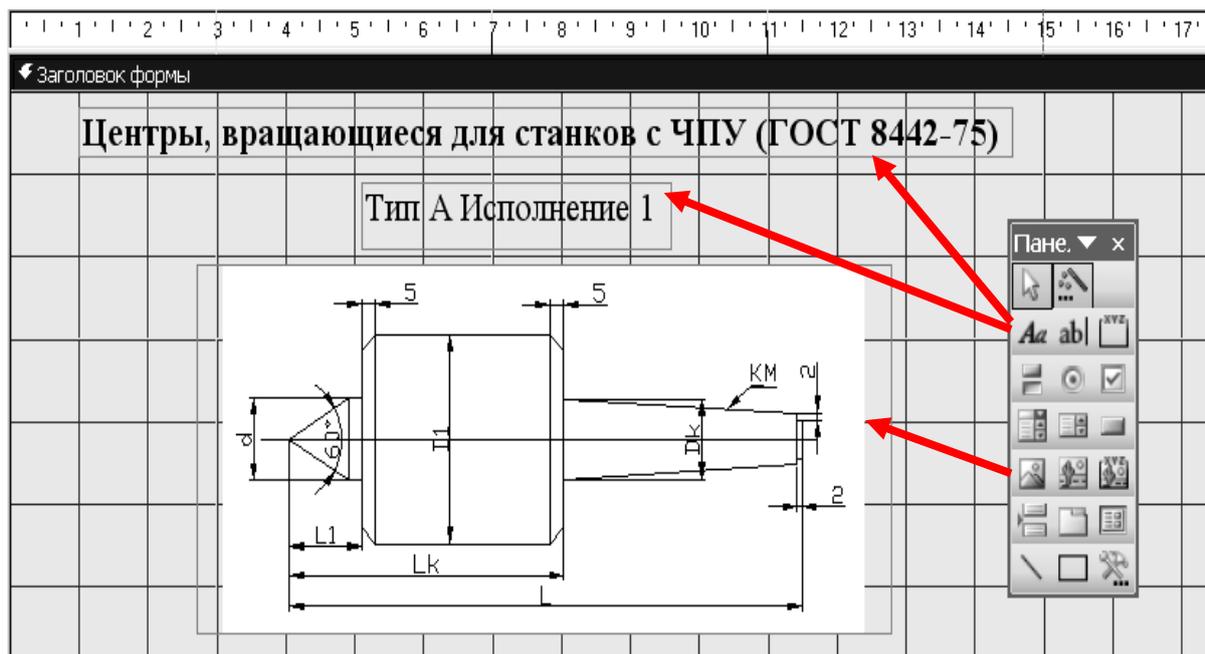


Рисунок 5.8 – Пример оформления заголовка формы



Рисунок 5.9 – Панель инструментов

В открывшемся окне мастера создания списков (рисунок 5.10) выбрать пункт «Объект «список»», использовать значения из таблицы или запроса и нажать кнопку **Далее**.

На следующем шаге создания списка выбрать требуемую таблицу (рисунок 5.11) и нажать кнопку **Далее**.

Для отбора данных, которые будут отображаться в списке, нажать кнопку выбора всех полей таблицы (рисунок 5.12), а затем кнопку **Далее**.

В следующем окне мастера создания списков (рисунок 5.13) выбрать порядок сортировки данных в списке **По возрастанию** и нажать кнопку **Далее**.

В следующем окне мастера создания списков (рисунок 5.14) отключить

флаг **Скрыть ключевой столбец**, перемещением границ определить ширину столбцов списка и нажать кнопку **Далее**.

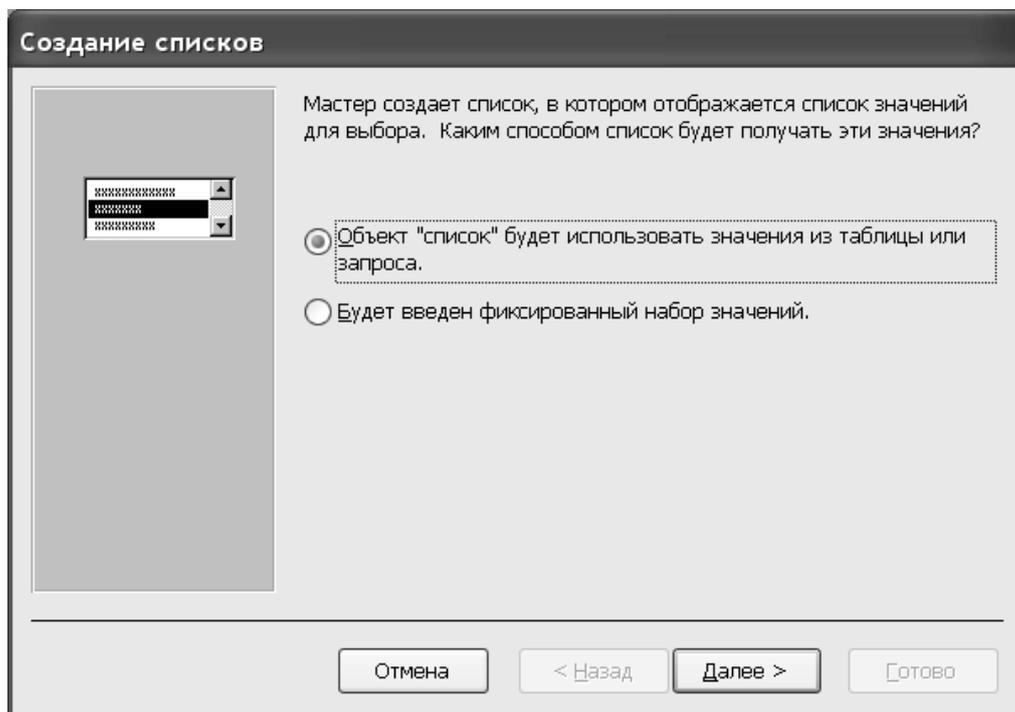


Рисунок 5.10 – Окно мастера создания списков

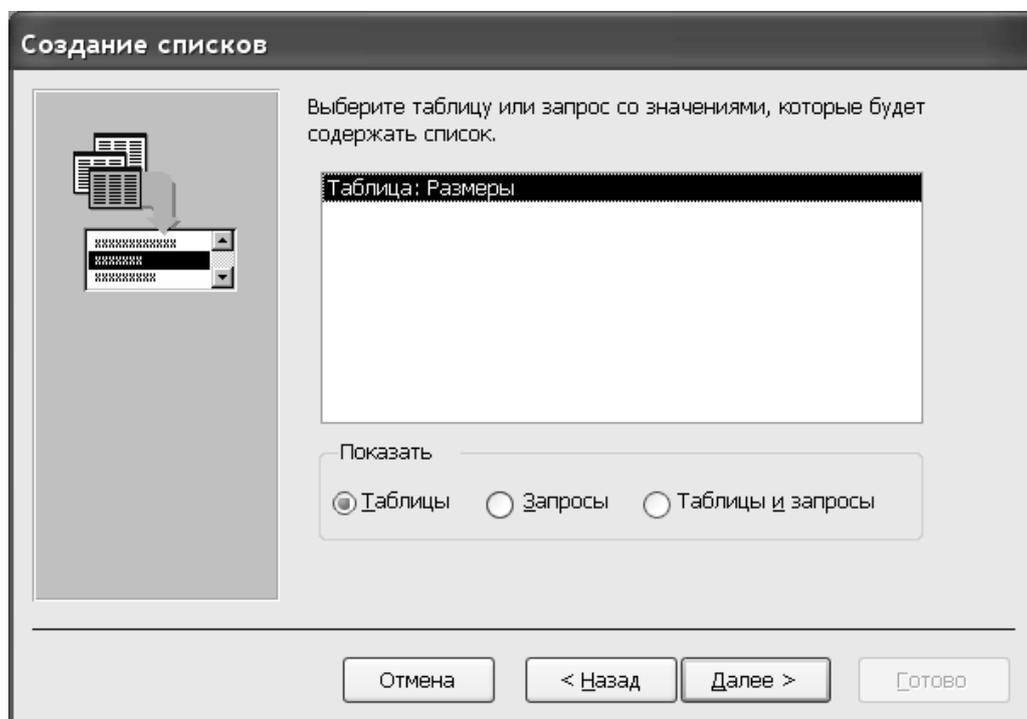


Рисунок 5.11 – Окно мастера создания списков на шаге выбора источника данных

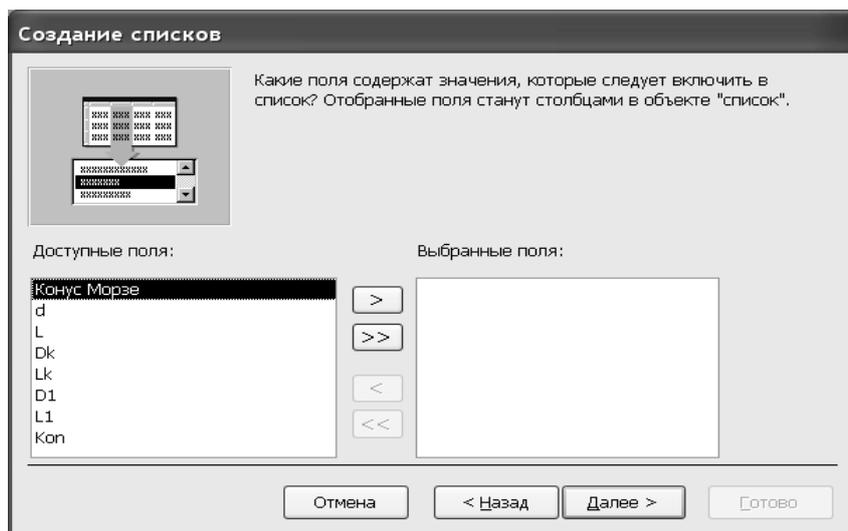


Рисунок 5.12 – Окно мастера создания списков на шаге выбора данных

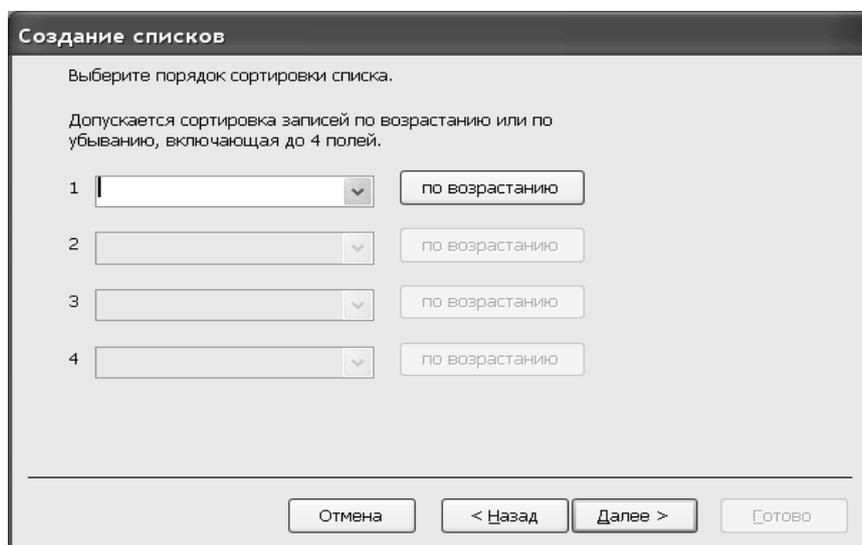


Рисунок 5.13 – Окно мастера создания списков на шаге выбора данных

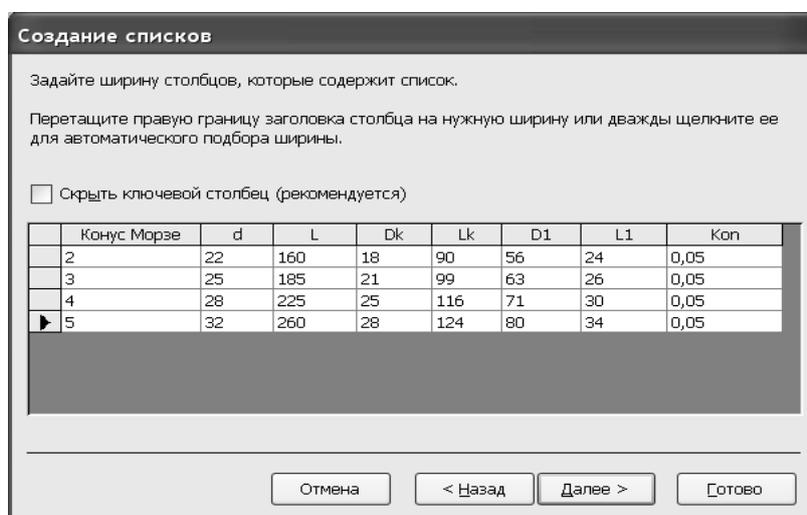


Рисунок 5.14 – Окно мастера создания списков на шаге выбора данных

На последнем шаге создания списка присвоить ему имя **Центры** и нажать кнопку **Готово** (рисунок 5.15).

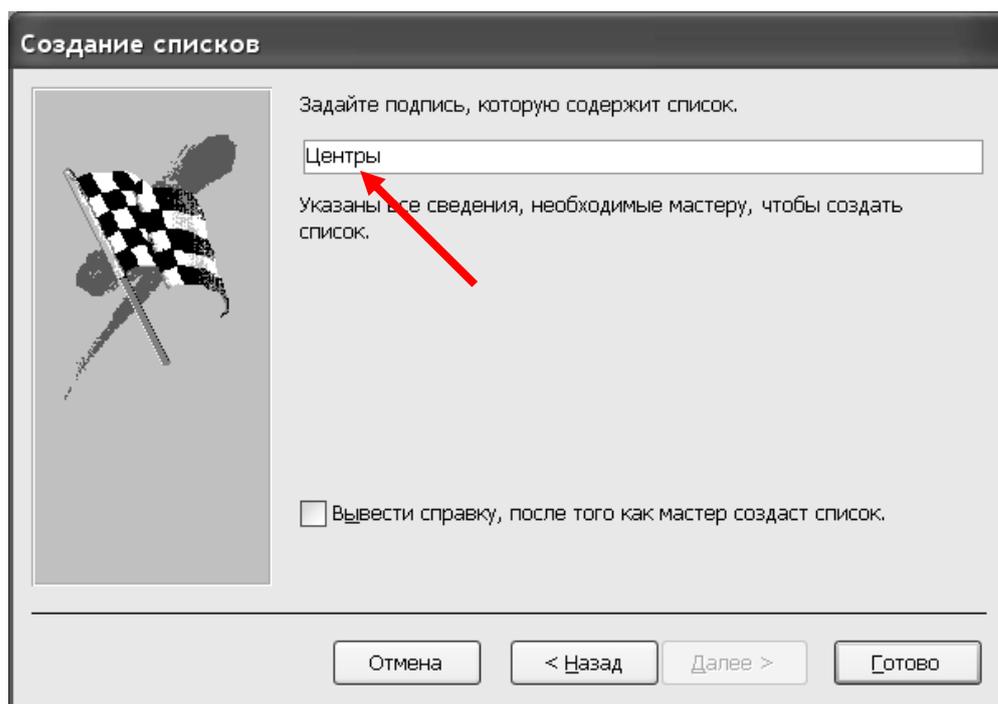


Рисунок 5.15 – Окно мастера создания списков на шаге выбора данных

Создать кнопку закрытия формы и выхода из базы данных. Для этого на панели инструментов (см. рисунок 5.9) выбрать команду **Кнопка**  и вставить изображение кнопки ниже таблицы (рисунок 5.16). В окне **Создание кнопок** из списка **Категории** выбрать **Работа с формой**, из списка **Действия** – «**Закрыть форму**», нажать кнопку **Готово**.

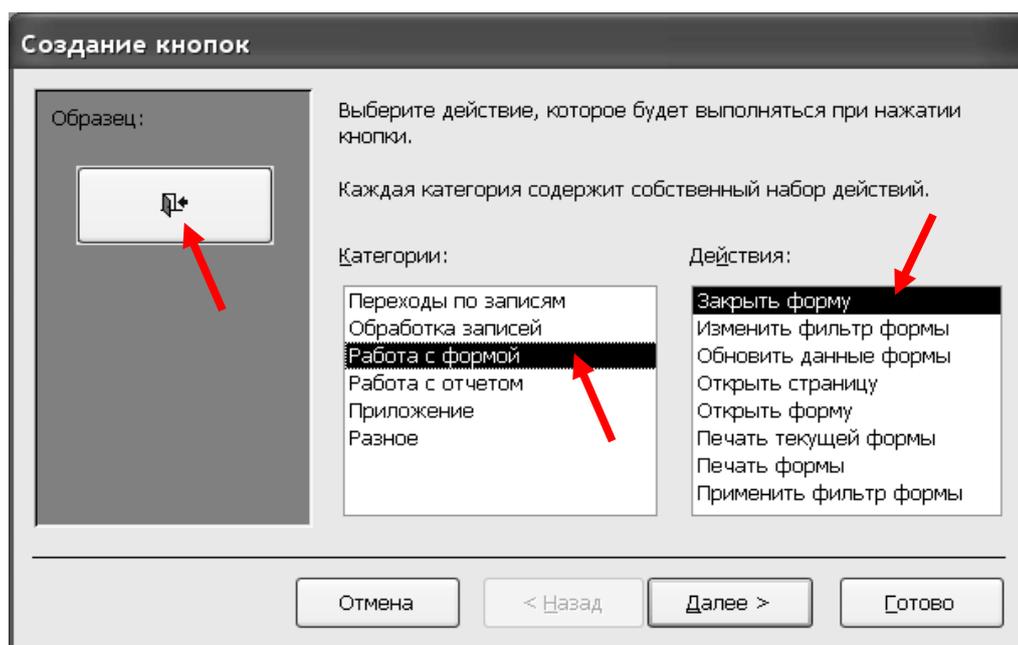


Рисунок 5.16 – Окно создания кнопок

Создать кнопку, которая в дальнейшем будет связана с макросом VBA для экспорта выбранных данных. Выбрать команду **Кнопка** и вставить изображение второй кнопки в нижней части формы.

Для задания имени и надписи кнопки щелчком правой клавиши мыши по изображению кнопки открыть контекстное меню, из которого выполнить команду **Свойства**. В окне свойств объекта открыть закладку **Все** и в полях **Имя** и **Подпись** записать слово **Экспорт** (рисунок 5.17).

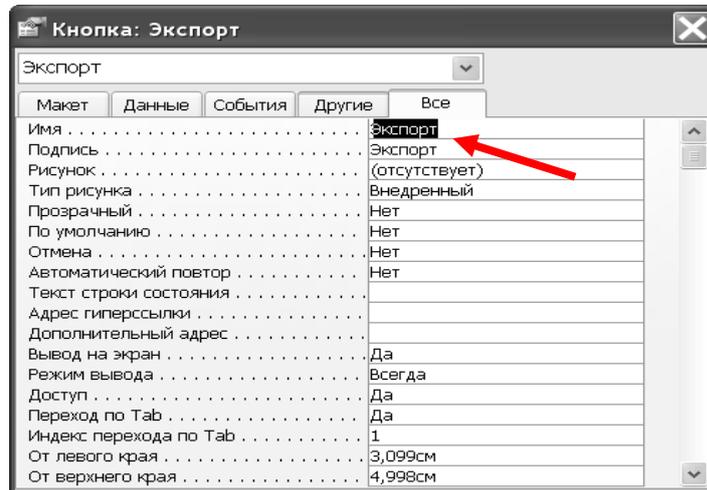


Рисунок 5.17 – Окно задания свойств кнопок

Для просмотра результатов создания формы выйти из режима конструктора нажатием кнопки **Вид**. Если создание формы было выполнено корректно (рисунок 5.18), то перейти к следующему этапу работы. В противном случае перевести форму в режим конструктора и сделать необходимые исправления.

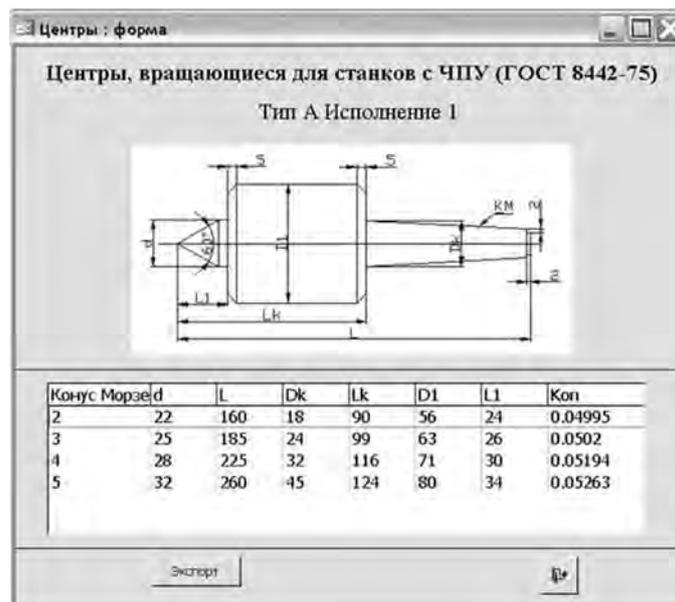


Рисунок 5.18 – Окно формы «Центры»

Для создания макроса экспорта данных открыть форму в режиме конструктора . Открыть контекстное меню кнопки **Экспорт** нажатием правой клавиши мыши по изображению кнопки. Из контекстного меню выполнить команду **Обработка событий** (рисунок 5.19).

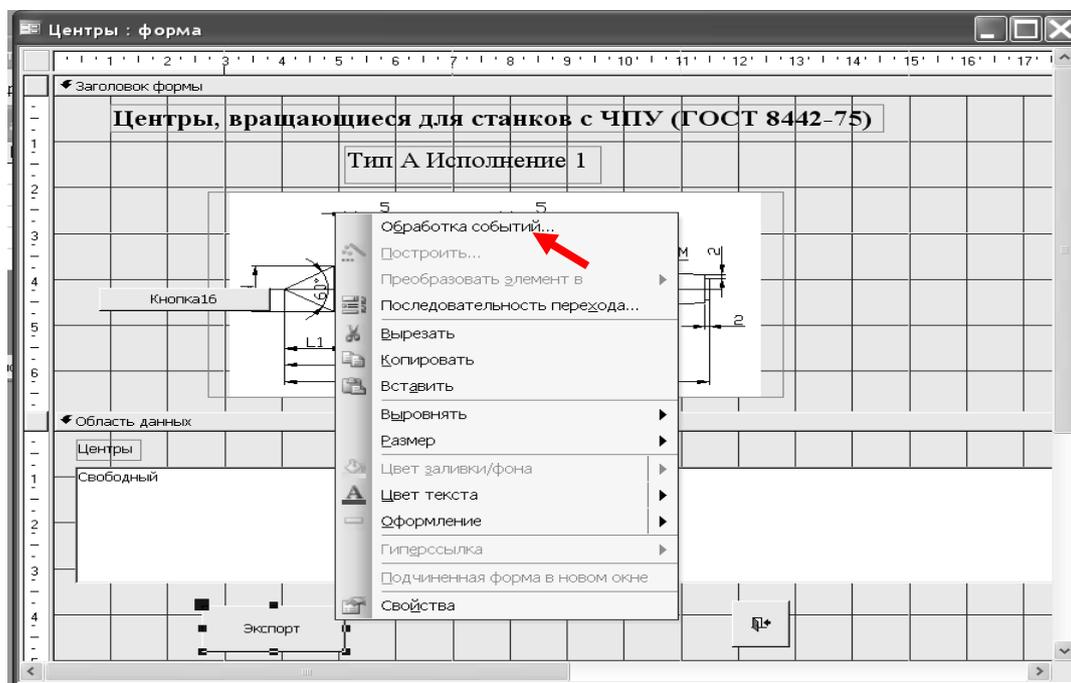


Рисунок 5.19 – Контекстное меню кнопки «Экспорт»

После выполнения команды **Обработка событий** открывается окно редактора Visual Basic, которое содержит уже готовую процедуру **Выход**, связанную с кнопкой закрытия формы, и операторы начала и конца процедуры, связанной с кнопкой **Экспорт** (рисунок 5.20).

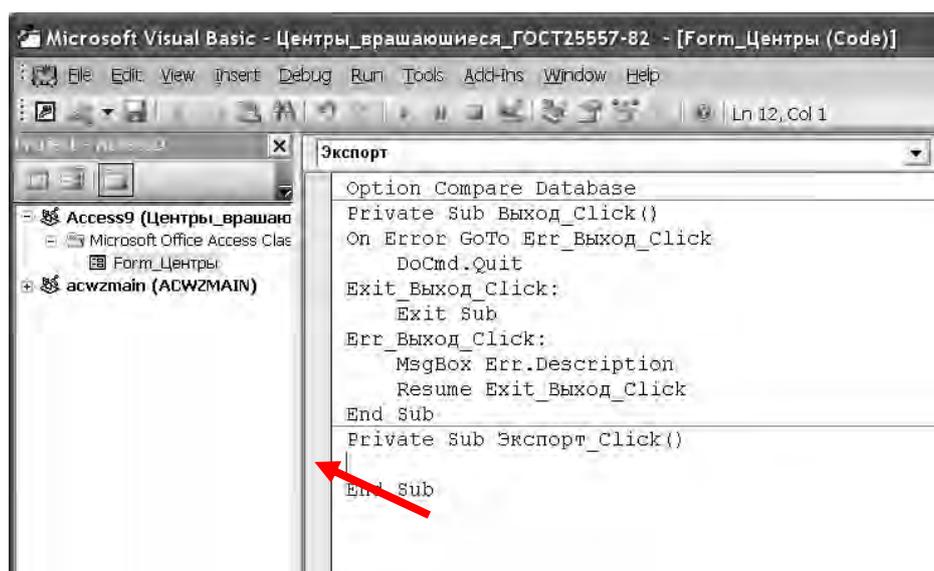


Рисунок 5.20 – Окно редактора Visual Basic

После оператора начала программы **Private Sub Экспорт_Click()** записать текст программы для чтения и экспорта параметров центров, который приведен ниже.

```
Private Sub Экспорт_Click()
Dim d, L, Dk, Lk, D1, L1, Kon As String
Dim Spis As Variant
'Поиск записи, выделенной в списке и чтение значений полей списка
For Each Spis In Me!Центры.ItemsSelected
d = Me!Центры.Column(1)
L = Me!Центры.Column(2)
Dk = Me!Центры.Column(3)
Lk = Me!Центры.Column(4)
D1 = Me!Центры.Column(5)
L1 = Me!Центры.Column(6)
Kon = Me!Центры.Column(7)
Next Spis
'Запись данных во временный файл
Open ".\FE.txt" For Output Access Write As #1
Print #1, d
Print #1, L
Print #1, Dk
Print #1, Lk
Print #1, D1
Print #1, L1
Print #1, Kon
Close #1
End Sub
```

После записи текста программы экспорта данных из базы данных стандартных изделий для проверки ее работы перейти в режим формы нажатием кнопки . Для проверки правильности выполнения программы в таблице формы выбрать любую запись и нажать кнопку **Экспорт**, после чего в папке **Мои документы** открыть файл экспорта данных **FE.txt**. Файл должен содержать список данных экспортированных из выбранной в таблице базы данных записи (рисунок 5.21). Повторить процедуру экспорта данных для различных записей таблицы и убедиться в корректности операции экспорта, просматривая файл **FE.txt**.

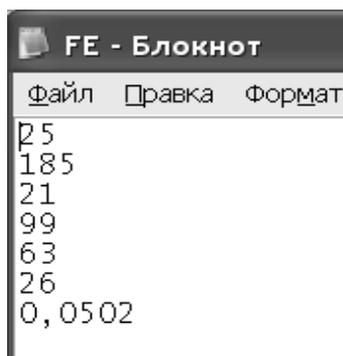


Рисунок 5.21 – Результаты экспорта данных из таблицы СУБД



На основе параметризованного изображения вращающегося центра (рисунок 5.22) построить контур, необходимый для создания 3D-модели методом вращения. На контуре определить точки, необходимые для его построения (рисунок 5.23).

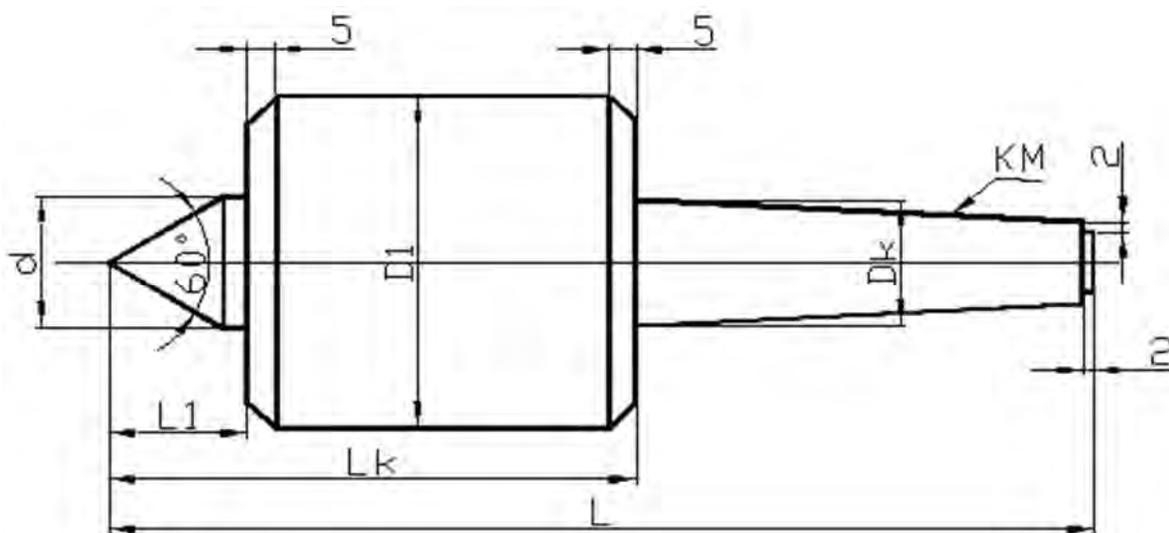


Рисунок 5.22 – Параметризованное изображение геометрической модели центра

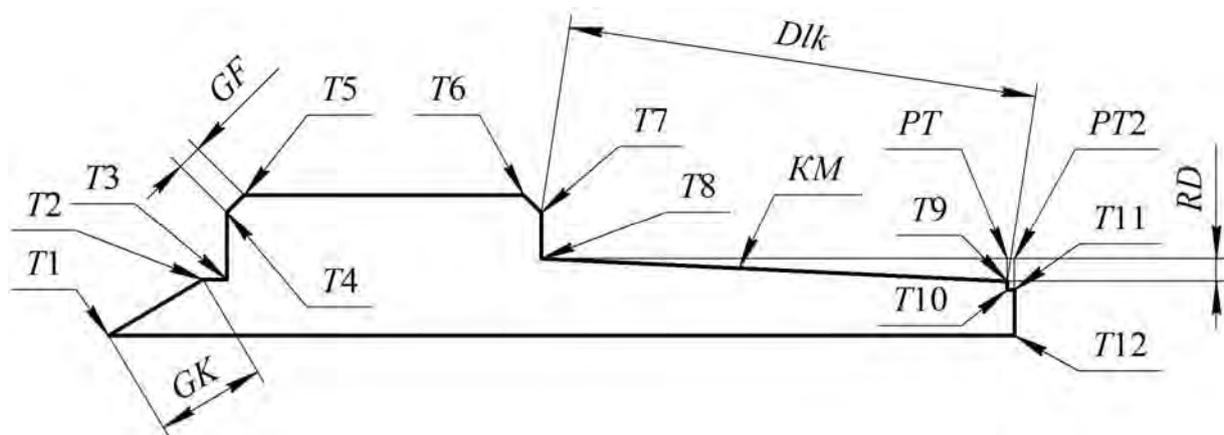


Рисунок 5.23 – Схема построения контура 3D-модели вращающегося центра

Для создания LISP-функции 3D-моделирования центров открыть редактор Visual LISP и записать приведенный ниже текст функции.

```

;*****
; LISP-функция 3D-моделирования вращающихся центров для станков с ЧПУ
; А. А. Иванов АТП-091                                     2012 г.
;*****
(Defun C:Центр_ТипА_Исп_1 ()
; Открытие файла исходных данных для чтения
(Setq F1 (Open "C:/TM/Мои документы/FE.txt" "r"))
  d (Atoi (Read-Line F1)) ;
  L (Atoi (Read-Line F1)) ;
  Dk (Atoi (Read-Line F1)) ; Чтение данных из файла

```

```

Lk (Atoi (Read-Line F1)) ; FE.txt и их преобразование
D1 (Atoi (Read-Line F1)) ; к числовому типу
L1 (Atoi (Read-Line F1)) ;
KM (Atof (Read-Line F1)) ;
)
(Close F1) ;Закрытие файла
;Вычисление координат точек для построения контура
(Setq T1 '(0 0) ; Начальная точка: X=0, Y=0
;Вычисление длины образующей конуса центра
GK (/ (/ d 2.0) (Sin (/ Pi 6.0)))
T2 (Polar T1 (/ Pi 6) GK)
T3 (Polar T2 0 (- 11 (* GK (Cos (/ Pi 6))))))
T4 (Polar T3 (/ Pi 2.0) (- (/ (- D1 d) 2.0) 5.0))
;Вычисление длины образующей фасок на корпусе
GF (/ 5.0 (Cos (/ Pi 4)))
T5 (Polar T4 (/ Pi 4) GF)
T6 (Polar T5 0 (- Lk 11 10))
T7 (Polar T6 (/ Pi -4) GF)
T8 (Polar T7 (* Pi 1.5) (- (/ (- D1 Dk) 2.0) 5.0))
;Вычисление длины конической части хвостовика
Dlk(- L LK 2.0)
RD (* Dlk KM)
PT (Polar T8 0 Dlk)
T9 (Polar PT (* Pi 1.5) RD)
T10 (Polar T9 (* Pi 1.5) 2.0)
T11 (Polar T10 0 2)
PT2 (Polar PT 0 2)
T12 (Polar PT2 (* Pi 1.5) (/ Dk 2.0))
)
;Визуализация модели
(Command "_-osnap" "" ;Отключение объектной привязки
;Построение контура
"_Pline" T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T1 ""
;Вращение контура
"_Revolve" "_All" "" T1 T12 360
;Раскрашивание модели по Гуро
"_Shademode" "_g"
))

```

Для проверки работы LISP-функции 3D-моделирования центров выполнить загрузку функции в среде AutoCAD командами **Сервис + Приложения + Загрузить**.

Далее в командной строке ввести название функции **Центр_ТипА_Исп_1** и пронаблюдать создание 3D-модели вращающегося центра.

Для тестирования разработанного программного комплекса выполнить последовательно экспорт из базы данных параметров для двух–трех центров и проверить выполнение команды **Центр_ТипА_Исп_1**. Результаты тестирования программного комплекса приведены на рисунке 5.24.



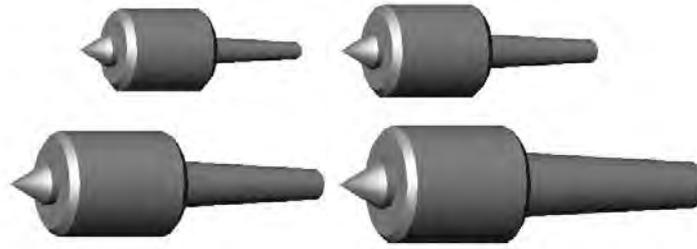


Рисунок 5.24 – Результаты тестирования LISP-функции

Содержание отчета

- 1 Цель работы.
- 2 Текст программы и результаты геометрического моделирования центра.
- 3 Выводы.

Контрольные вопросы

- 1 Для чего предназначена система управления базами данных Access?
- 2 Какие функции содержит программирование высокого уровня?
- 3 Какой программный продукт является клиентом автоматизации при параметрическом проектировании?

6 Лабораторная работа № 14. Автоматизированное геометрическое моделирование узлов РТК

Цель работы: изучение методики параметрического моделирования способом автоматизации параметрического и геометрического синтеза конструкций деталей РТК.

Простым и эффективным способом автоматизации параметрического и геометрического синтеза конструкций деталей средств автоматизации является организация взаимодействия различных систем автоматизированного проектирования посредством обмена данными через файлы экспорта.

Например, при структурно-параметрическом синтезе конструкции поршня (рисунок 6.1) его параметры ($D1$, $D2$ и т. д.) определяются в среде электронных таблиц.

После определения параметров поршня выполняется экспорт данных в файл экспорта с помощью макроса VBA. На этапе геометрического синтеза модели значения параметров импортируются из файла экспорта в среду системы геометрического моделирования для создания геометрической модели детали. Схема обмена данными приведена на рисунке 6.2.

В работе рассматривается реализация методики обмена данными между системами параметрического и геометрического синтеза деталей при создании 3D-модели поршня пневмоцилиндра.

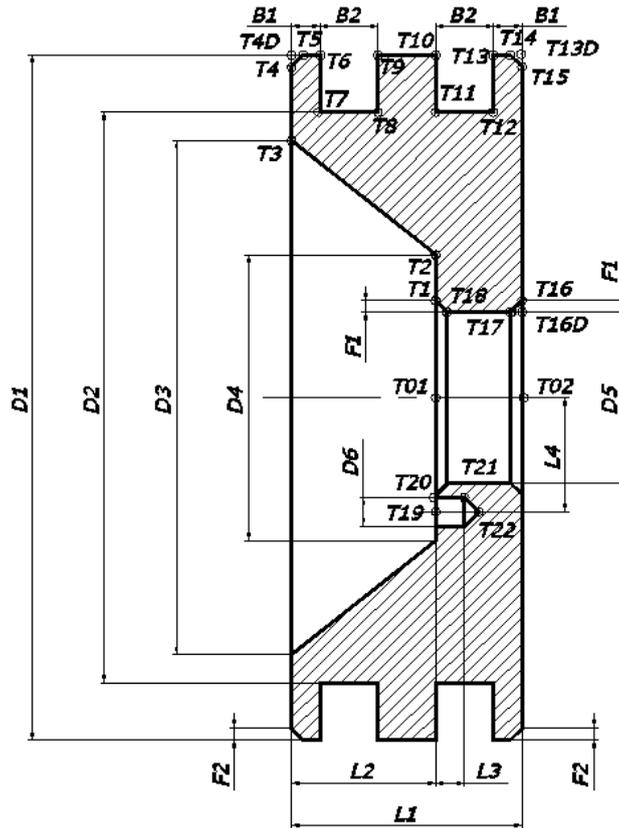


Рисунок 6.1 – Параметризованный чертеж поршня

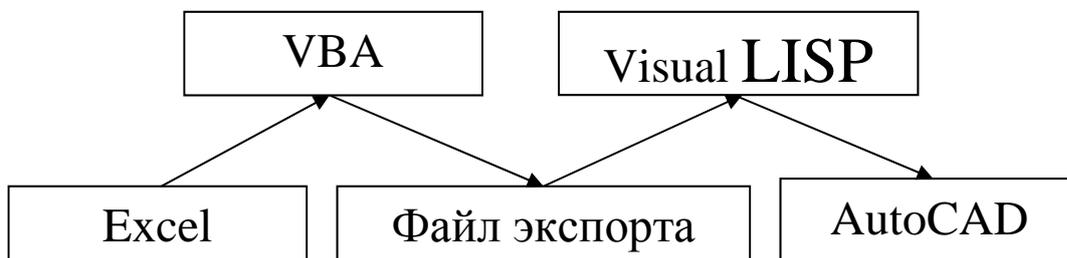


Рисунок 6.2 – Схема обмена данными между системами параметрического и геометрического синтеза

Организация экспорта данных из электронных таблиц Excel

Из папки методического обеспечения скопировать файлы ПоршеньПр3.xls и ПоршеньПр3.dwg в свою рабочую папку.

Открыть лист в режиме конструктора



Из контекстного меню кнопки «Экспорт результатов параметрического синтеза», расположенной в конце описания проектной процедуры параметрического синтеза поршня, выполнить команду **Исходный текст**. Ознакомьтесь с содержанием макроса VBA «Экспорт». Его содержание должно соответствовать тексту, расположенному ниже.

'Проектная процедура экспорта результатов параметрического синтеза поршня пневмоцилиндра

Private Sub Экспорт_Click()

Open "Путь к файлу в вашей рабочей папке \EP.txt" For Output Access Write As #1

Write #1, Worksheets(1).Range("B17").Value 'D1

Write #1, Worksheets(1).Range("B18").Value 'D2

Write #1, Worksheets(1).Range("B19").Value 'B2

Write #1, Worksheets(1).Range("B20").Value 'B1

Write #1, Worksheets(1).Range("B21").Value 'L1

Write #1, Worksheets(1).Range("B23").Value 'D5

Write #1, Worksheets(1).Range("B26").Value 'L2

Write #1, Worksheets(1).Range("B28").Value 'D6

Write #1, Worksheets(1).Range("B30").Value 'L4

Write #1, Worksheets(1).Range("B31").Value 'L3

Write #1, Worksheets(1).Range("B32").Value 'D4

Write #1, Worksheets(1).Range("B33").Value 'D3

Write #1, Worksheets(1).Range("B34").Value 'F1

Write #1, Worksheets(1).Range("B35").Value 'F2

Close #1

End Sub

Выйти из режима конструктора.

В ячейках B5, B7 и B8 ввести исходные данные (рисунок 6.3). Двойным щелчком по кнопке «Экспорт результатов параметрического синтеза» выполнить экспорт данных в файл EP.txt.

1	Подсистема проектирования пневмоцилиндров		
2	Проектная процедура вычисления значений параметров поршня		
3	Исходные данные		
4	Сила трения при контакте уплотнения с гильзой пневмоцилиндра, Н	$P_1 =$	0
5	Усилие при зажиме или транспортировании, Н	$P_2 =$	5000
6	Вес поршня и перемещаемых частей привода, Н	$P_3 =$	0
7	Минимальное абсолютное давление в магистрали, Па (Н/м ²)	$p_m =$	600000
8	Атмосферное давление (101 325 Па)	$p_a =$	101325

Рисунок 6.3 – Область ввода исходных данных

Выполнить ассоциативную загрузку системы геометрического моделирования AutoCAD двойным щелчком по значку файла ПоршеньПр3.dwg в своей рабочей папке.

Командой **Сервис + AutoLISP + Редактор Visual LISP** загрузить интегрированную систему разработки приложений Visual LISP. Создать файл под именем GSP.lsp (Геометрический синтез поршня) в своей рабочей папке. Набрать текст LISP-функции геометрического синтеза поршня, показанный ниже.

(DEFUN C : GSP ()

;Открытие файла экспорта данных из таблицы Excel

(Setq F1 (Open "Путь к вашей рабочей папке/EP.txt" "r")

;Чтение данных из Файла экспорта данных

D1 (Atoi (Read-Line F1))



;Диаметр дна канавок для уплотнения
 D2 (Atof (Read-Line F1))
 ;Ширина канавок для уплотнений
 B2 (Atof (Read-Line FI))
 ;Ширина крайней щеки поршня
 B1 (Atoi (Read-Line FI))
 ;Длина поршня
 LI (Atoi (Read-Line F1))
 ;Диаметр отверстия для крепления штока
 D5 (Atoi (Read-Line F1))
 ;Глубина поднутрения
 L2 (Atoi (Read-Line F1))
 ;Диаметр отверстия под лапку стопорной шайбы
 D6 (Atoi (Read-Line F1))
 ;Межосевое расстояние центрального отверстия и отверстия под лапку стопорной шайбы
 L4 (Atof (Read-Line F1))
 ;Глубина отверстия под лапку стопорной шайбы
 L3 (Atof (Read-Line F1))
 ;Меньший диаметр поднутрения поршня
 D4 (Atoi (Read-Line F1))
 ;Большой диаметр поднутрения поршня
 D3 (Atoi (Read-Line FI))
 ;Ширина Фаски в центральной отверстии
 Fs1 (Atoi (Read-Line F1))
 ;Ширина Фаски на наружной поверхности поршня
 Fs2 (Atoi (Read-Line F1))
)
 (Close F1)
 ;Вычисление координат расположения точек профиля поршня
 (Setq T02 (0 0 0))
 T01 (Polar T02 Pi (- LI L2))
 90G (/ PI 2.0)
 T1 (Polar T01 90G (+ (/ D5 2.0) Fs1))
 T2 (Polar T01 90G (/ D4 2.0))
 T2D (Polar T2 Pi L2)
 T3 (Polar T2D 90G (/ (- D3 D4) 2.0))
 T4 (Polar T3 90G (- (/ (- DI D3) 2.0) Fs2))
 T4D (Polar T4 90G Fs2)
 T5 (Polar T4D 0 Fs2)
 T6 (Polar T4D 0 B1)
 270G (* PI 1.5)
 T7 (Polar T6 270G (/ (- D1 D2) 2.0))
 T8 (Polar T7 0 B2)
 T9 (Polar T8 90G (/ (- DI D2) 2.0))
 T10 (Polar T9 0 (- L1 B1 B1 B2 B2))
 T11 (Polar T10 270G (/ (- 01 D2) 2.0))
 T12 (Polar T11 0 B2)
 T13 (Polar T12 90G (/ (- DI D2) 2.0))
 T13D (Polar T13 0 B1)
 T14 (Polar T13D Pi Fs2)
 T15 (Polar T13D 270G Fs2)



```

T16D (Polar T13D 270G (/ (- D1 D5) 2.0))
T16 (Polar T16D 90G Fsl)
T17 (Polar T16D Pi Fsl)
T18 (Polar T17 Pi (- LI L2 Fsl Fsl))
; Вычисление координат расположения точек профиля отверстия для
; лапки стопорной шайбы
T19 (Polar T01 270G L4)
T20 (Polar T19 90G (/ D6 2.0))
T21 (Polar T20 0 L3)
LG (/ (/ D6 2.0) (COS (/ Pi 6.0)))
T22 (Polar T21 (/ Pi -3) LG)
(Command "_-OSNAP ; Отключение объектной привязки
; Построение контура ПРОФИЛЯ поршня
"_Pline" T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15 T16 T17 T18 T1 ""
; Вращение контура поршня
"_Revolve" "_last T01 T02 360
; Построение контура ПРОФИЛЯ отверстия для лапки стопорной шайбы
"_Pline" T19 T20 T21 122 T19 :Вращение контура поршня
"_Revolve" ".last T19 T22 360
; Вычитание из объема поршня объема отверстия
"_Subtract" T3 "" T22 ""
; Раскрашивание 3D-модели по Гуро
"_Shademode" "_g"
" zoom" " all" ;Показать всё
)
(Prompt "\n Укажите требуемое положение объекта")
; Перемещение объекта пользователей в требуемую точку
(Command "_Move" T02 "_last" T02 Pause)

```

Сохранить набранный текст LISP-функции. Командой **Сервис + Приложение** загрузить файл LISP-функции GSP.lsp.

Для выполнения функции в командной строке AutoCAD набрать имя загруженной LISP-функции GSP.

Оценить полученный результат. При наличии ошибок устранить их.

Содержание отчета

- 1 Цель работы.
- 2 Скриншот области листа электронной таблицы с исходными данными и результатами параметрического синтеза поршня.
- 3 Текст файла экспорта данных.
- 4 Параметризованный чертеж поршня.
- 5 Текст LISP-функции.
- 6 Скриншот 3D-модели поршня.
- 7 Выводы



Контрольные вопросы

- 1 Области применения САМ-систем.
- 2 Основные типы САМ-систем.
- 3 Особенности программирования с использованием LISP-функций.

Список литературы

- 1 **Берлинер, Э. М.** САПР в машиностроении : учебник для вузов / Э. М. Берлинер, О. В. Таратынов. – Москва : Форум, 2011. – 448 с.
- 2 **Бунаков, П. Ю.** Технологическая подготовка производства в САПР : учебное пособие / П. Ю. Бунаков, Э. В. Широких. – Москва : ДМК Пресс, 2012. – 208 с.
- 3 **Акулович, Л. М.** Основы автоматизированного проектирования технологических процессов в машиностроении : учебное пособие для вузов / Л. М. Акулович, В. К. Шелег. – Минск : Новое знание, 2016. – 488 с.
- 4 Основы построения САПР ТП в многономенклатурном машиностроительном производстве : учебник / Г. Б. Бурдо [и др.]. – Старый Оскол : ТНТ, 2013. – 280 с.
- 5 **Берлинер, Э. М.** САПР технолога-машиностроителя : учебник / Э. М. Берлинер, О. В. Таратынов – Москва : Форум ; ИНФРА-М, 2015. – 336 с.
- 6 Основы автоматизированного проектирования : учебник / Под ред. А. П. Карпенко. – Москва : ИНФРА-М, 2017. – 329 с.
- 7 **Рязанцев, А. Н.** Автоматизация проектирования технологических процессов. Сборник задач : учебное пособие / А. Н. Рязанцев, А. А. Жолобов. – Минск : Дизайн ПРО, 1997. – 121 с.
- 8 Классификатор технологических операций машиностроения и приборостроения : 1 85 151. – Москва : Изд-во стандартов, 1987. – 72 с.
- 9 САПР технологических процессов : методические рекомендации к аудиторной контрольной работе для студентов специальности 1-36 01 01 «Технология машиностроения» / Сост. М. Н. Миронова. – Могилев : Беларус.-Рос. ун-т, 2017. – 16 с.