

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«Брестский государственный технический университет»**

Кафедра информатики и прикладной математики

Методические рекомендации

по выполнению курсовых работ по дисциплине «Информатика»
для специальностей

36 01 01 «Технология машиностроения»,

36 01 03 «Технологическое оборудование машиностроительного производства»

Издание четвертое переработанное, дополненное

УДК 681.3

Методические рекомендации содержат сведения о требованиях к содержанию и оформлению курсовых работ по дисциплине «Информатика», структуре программы, базовых структурах программирования, примеры решения некоторых типовых задач и стандартные процедуры. Примеры приведены на языке Visual Basic 6.0 и Visual Basic for Applications для Microsoft Excel.

Предназначены для студентов первого и второго курсов специальности 36 01 01 «Технология машиностроения» и 36 01 03 «Технологическое оборудование машиностроительного производства» по дисциплине «Информатика» очной формы обучения и имеют целью оказать помощь студентам в подготовке и оформлении курсовых работ по названным дисциплинам. Может быть полезна также студентам других специальностей, аспирантам и магистрантам при выполнении выпускных и дипломных работ.

Издание четвертое переработанное, дополненное.

Составитель: В.Л. Быков, к. т. н., доцент

Рецензент: Котов И.В., зав. кафедрой информационных технологий Брестского государственного университета им. А.С. Пушкина, к. ф.-м. н., доцент

Содержание

1. Общие положения.....	4
1.1. Выбор варианта задания.....	4
1.2. Требования к содержанию курсовой работы.....	4
1.3. Примерные варианты типовых тем курсовых работ.....	4
1.4. Содержание пояснительной записки.....	5
1.5. Требования к оформлению пояснительной записки.....	6
2. Основы программирования.....	8
2.1. Этапы разработки программы.....	8
2.2. Структура программы.....	9
3. Типовые алгоритмы и процедуры.....	12
3.1. Паспорт программы.....	12
3.2. Исследование функции на отрезке.....	12
3.3. Решение алгебраических и трансцендентных уравнений.....	13
Отделение корней уравнения.....	13
Уточнение значения корня на отрезке отделения.....	14
3.4. Вычисление определенного интеграла.....	14
3.5. Построение изображения детали.....	15
3.6. Моделирование движения механизма.....	17
А. Моделирование движения механизма с целью определения зон действия звеньев механизма.....	17
Б. Исследование кинематики управления механизмом.....	21
3.7. Методические рекомендации по выполнению курсовой работы № 2.....	24
3.7.1. Разработка меню пользователя.....	25
3.7.2. Функции пользователя.....	26
3.7.3. Разработка окна диалога.....	28
Приложение 1.....	30
Пример оформления курсовой работы.....	30
Приложение 2.....	34
Рисунки для расчета движения механизма.....	34
Приложение 3.....	38
Рисунки для расчета манипулятора.....	38
Приложение 4.....	42
Таблица исходных данных для расчета манипулятора.....	42
Приложение 5.....	43
Рисунки для моделирования движения механизма.....	43
Список рекомендуемой литературы.....	47

1. Общие положения

Курсовые работы по дисциплине Информатика выполняются студентами в третьем и четвертом семестрах на основе изученного материала. В третьем семестре курсовая работа выполняется на языке программирования Visual Basic 6.0, а в четвертом семестре в электронной таблице Excel средствами электронной таблицы с использованием языка программирования Visual Basic for Applications в объеме учебной программы. Эта особенность определила структуру и содержание методических рекомендаций.

1.1. Выбор варианта задания

Вариант выполнения курсовой работы выбирается по двум последним номерам зачетной книжки. Если номер зачетной книжки больше числа вариантов, то берется вариант по последнему номеру зачетной книжки.

Например, если у студента Жука В. Д. номер зачетной книжки 280305, тогда номер задания 5, если у студента Мамонова И. Б. номер зачетной книжки 289037, тогда номер задания 7.

Порядок выбора варианта может быть изменен по указанию преподавателя.

1.2. Требования к содержанию курсовой работы

Курсовая работа выполняется студентом самостоятельно с использованием языка программирования Visual Basic или в электронной таблице средствами Excel с использованием Visual Basic for Application.

Объем и содержание курсовой работы должны быть достаточными для проверки знаний студента по программированию в объеме учебного материала и оформлению документов в среде Microsoft Word в объеме курса. Работа может быть расчетной, графической или смешанной расчетно-графической. Общий объем работы, в зависимости от содержания, должен составлять 15 - 20 листов машинописного текста. Программа должна быть выполнена с использованием принципов структурного программирования.

Задание на выполнение курсовой работы выдается преподавателем или выбирается студентом самостоятельно по одной из профилирующих кафедр и согласовывается с преподавателем. Допускается в качестве темы курсовой работы выбирать выполненные или выполняемые расчетно-графические работы по другим дисциплинам.

1.3. Примерные варианты типовых тем курсовых работ

А. Расчетные:

разработать программу для ввода блока данных, сохранения их на диске, загрузки, обработки и вывода результатов вычислений на экран в виде таблиц и печать в виде отформатированных таблиц;

Б. Расчетно-графические:

разработать программу для ввода блока данных, сохранения их на диске, загрузки, обработки данных и вывода результатов на экран и печать в виде графиков функций и диаграмм. Например:

разработать демонстрационную программу для решения линейных и трансцендентных уравнений тремя способами;

разработать демонстрационную программу для анализа функций одной и двух переменных.

В. Графические:

разработать программу для моделирования изображения детали (аксонометрия и проекции). Программа должна обеспечивать изменение масштаба, поворот изображе-

ния, перемещение изображения в любую точку экрана с контролем выхода изображения за границы экрана;

разработать программу моделирования движения механизма. Программа должна обеспечивать изображение механизма в статике, динамике, построение зон действия звеньев механизма, построение траектории движения заданной точки механизма;

и др.

1.4. Содержание пояснительной записки

Пояснительная записка должна включать: титульный лист, бланк задания, содержание, введение, основную часть, заключение, список использованной литературы, приложения.

Во введении излагается тема, цель работы, ее актуальность, анализ известных методов решения задачи, преимущества предлагаемого решения.

Основная часть должна содержать постановку задачи, исходные данные, математическую модель (расчетные формулы), краткие теоретические сведения об используемых методах решения задачи, рисунок механизма или детали, описание форм, переменных, описание свойств объектов, установленных на форму, схемы алгоритмов. В пояснительную записку целесообразно включать общую схему алгоритма и полную схему алгоритма одного из блоков программ по указанию преподавателя.

Таблица 1.1

Описание переменных ← заголовок таблицы

Имя программы	Назначение программы		Лист
			Листов
Переменная			Комментарий
обозначение в формуле	имя переменной	тип переменной	
X _n	a	Single	Начало отрезка табулирования

Описание переменных и свойств объектов представляется в виде таблиц (табл.1.1, табл. 1.2).

Таблица 1.2.

Свойства элементов управления формы Статика

Наимен. программы	Тема проекта:	
	Лист 2	Листов 1
<i>Mechanizm</i>	<i>Моделирование движения механизма</i>	
Объект	Свойство	Значение
Форма	Name	frmStatica
	Caption	Статика
	AutoRedraw	True
	MDIChild	True
Командная кнопка 1	Name	cmdStatica
	Caption	Старт
	Height	375
	Width	1335

В приложение включаются распечатки паспорта программы, меню, алгоритма программы, таблиц результатов вычислений, графиков функций и диаграмм, иллюстрирующих возможности программы, перечень использованных операторов и команд без комментариев.

Распечатки программ не должны содержать никаких комментариев к тексту программы.

1.5. Требования к оформлению пояснительной записки

Пояснительная записка и приложения оформляются в соответствии с требованиями ГОСТ 2.105-79. ЕСКД: Общие требования к текстовым документам. В университете разработаны требования к оформлению дипломных и курсовых проектов и работ [1], которыми следует пользоваться при выполнении работы. Титульный лист, рамки, таблицы и

<p style="text-align: center;">Министерство образования Республики Беларусь Учреждение образования «Брестский государственный технический университет» Кафедра информатики и прикладной математики</p> <p style="text-align: center;">Тема: Исследование функции одной переменной</p> <p style="text-align: right;">Выполнил: студент группы № _____ _____ (Фамилия, инициалы)</p> <p style="text-align: right;">Консультант: _____ _____ (Фамилия, инициалы)</p> <p style="text-align: right;">Проверил: _____ _____ (Фамилия, инициалы)</p> <p style="text-align: center;">г. Брест, 2009 г.</p>

Рис. 1.1. Пример оформления титульного листа

рисунки оформляются средствами текстового процессора. Текст пояснительной записки пишется от руки на стандартных листах формата А4. Текст должен иметь поля: слева - 30 мм, справа - 10 мм, сверху - 20 мм, снизу - 20 мм. Средняя плотность текста по вертикали 28 - 30 строк на страницу. Титульный лист оформляется на ЭВМ с использованием одного из редакторов текста. Пример оформления титульного листа приведен на рис.1.1. Приложения распечатываются на принтере.

На первом листе пояснительной записки изображается угловой штамп для документов (рис. 1.2.), на последующих листах – небольшой угловой штамп (рис. 1.3).

Текст курсовой работы разделяется на разделы и подразделы. Разделы нумеруются арабскими цифрами с точкой, например, 1., 2. и так далее. Если в разделе есть подразделы, то они нумеруются в пределах данного раздела (1.1., 1.2. ...). Заголовки разделов и подразделов записываются заглавными буквами. Точка в конце заголовка не ставится. Заголовки отделяются от текста одной свободной строкой.

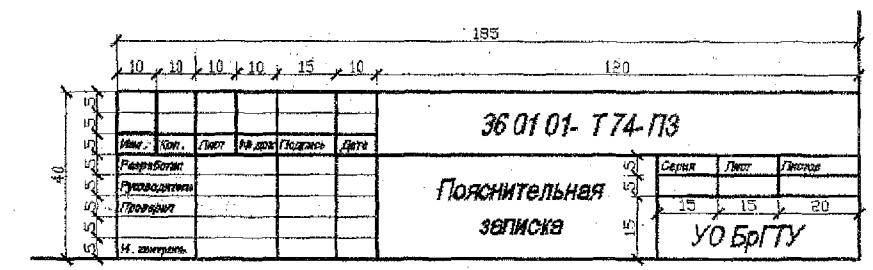


Рис. 1.2. Пример углового штампа для первого листа

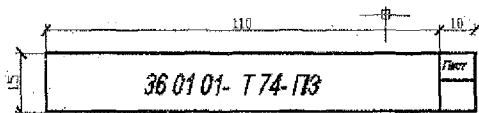


Рис.1.3. Угловой штамп на последующих страницах

Все страницы курсовой работы нумеруются, на первой странице номер не ставится. Нумерация страниц должна быть сквозная, включая приложения.

Формулы, используемые в пояснительной записке, записываются в общем виде в отдельной строке и нумеруются. Номер формулы записывается арабскими цифрами у правого края листа и заключается в круглые скобки. Ниже формулы приводится пояснение имеющихся в ней символов. Описание каждого символа приводится в отдельной строке. Пояснения к символам, кроме последнего, заканчиваются точкой с запятой, а пояснения к последнему символу - точкой.

Ссылки на формулы в тексте пояснительной записки указываются в виде их номеров в круглых скобках, например, (3.5). Формулы нумеруются в пределах раздела.

Таблицы, если их несколько, нумеруются в пределах раздела. Боковые и нижние ограничивающие линии не изображаются (Табл. 1.1.). Если таблица разделена на несколько частей, то номер таблицы и заголовок пишутся только над первой частью, а над последующими частями пишут "Продолжение таблицы" и ее номер. Ссылки на номера таблиц указываются следующим образом: Табл. 3.1.

Рисунки выполняются на отдельных листах. Все рисунки, так же как и формулы, нумеруются в пределах раздела.

Различная ориентация текста, таблиц и рисунков не допускается.

В тексте пояснительной записки разрешается использовать без расшифровки только общепринятые сокращения. Если используются нестандартные сокращения, то они должны быть расшифрованы в тексте при первом упоминании. Использовать сокращения отдельных слов не разрешается.

Схемы алгоритмов выполняются в соответствии с требованиями ГОСТ 19.003-80. Соотношение между геометрическими элементами схем установлены стандартом: размер a выбирается из ряда 10, 15, 20 мм. Допускается увеличивать размер a на число кратное 5. Размер b равен $1,5a$, допускается устанавливать размер b равным $2a$. В настоящее время размеры блоков не нормируются.

Список использованных источников составляется в алфавитном порядке или в порядке ссылок на источники в тексте. Примеры правильной записи некоторых источников приведены в списке использованной литературы настоящих рекомендаций. Номер источника проставляется в конце фразы или данных, заимствованных из соответствующего источника, и заключается в квадратные скобки, например [5].

Приложение должно начинаться с новой страницы. Если приложений несколько, то каждое из них должно начинаться с новой страницы. В верхнем правом углу пишется заглавными буквами слово "ПРИЛОЖЕНИЕ" и его номер, а в центре следующей строки пишется заглавными буквами название приложения.

2. Основы программирования

2.1. Этапы разработки программы

Всякая программа проходит от момента своего появления до списания определенный жизненный цикл. Основными этапами этого цикла являются: содержательная постановка задачи, математическая постановка задачи, выбор метода решения, разработка математической модели, разработка схемы алгоритма, написание программы на одном из языков программирования, отладка программы, сдача программы в эксплуатацию и научно-техническое сопровождение программы.

Содержательная постановка задачи заключается в формулировке задачи на естественном языке.

Математическая постановка задачи сводится к точному описанию исходных данных, условий задачи и целей ее решения с использованием математических выражений в общем виде.

Математическая модель задачи представляет собой совокупность математических выражений, описывающих данную задачу.

Метод решения задачи выбирается из известных методов. При отсутствии известных методов, разрабатывается свой, оригинальный метод решения. После выбора метода решения осуществляется **формализация** задачи, или, иными словами, описание исходных данных и условий задачи в виде, удобном для ввода в ЭВМ, неформализованные условия задачи представляются в математической форме.

Схема алгоритма является одним из способов наглядного представления алгоритма с помощью специальных элементов, предусмотренных ЕСКД.

Алгоритм - точное, однозначное предписание последовательности действий (операций), приводящее к решению задач данного класса за конечное число шагов или заданное время.

Написание программы на одном из языков программирования заключается в описании схемы алгоритма с помощью операторов и команд соответствующего языка высокого уровня.

Отладка программы заключается в проверке правильности функционирования алгоритма решения задачи с помощью контрольных примеров, результаты решения которых заведомо известны, устранение обнаруженных синтаксических и логических ошибок.

Научно-техническое сопровождение программы предусматривает контроль за результатами работы программы и устранение ошибок, обнаруженных в процессе эксплуатации, доработке программы и ее совершенствовании в соответствии с требованиями заказчика.

2.2. Структура программы

При разработке программы необходимо руководствоваться принципами структурного программирования. К ним относятся *нисходящая разработка, модульное проектирование и использование базовых структур*.

При реализации принципа нисходящей разработки программа разрабатывается в следующей последовательности.

1. На основе анализа задача разбивается на подзадачи, выделяются уровни и подуровни, составляется иерархическая структура программы.

2. Определяются глобальные переменные, процедуры и функции.

3. Для каждого уровня:

разрабатывается математическая модель и выбирается метод решения задачи; определяются основные блоки программы и разрабатывается укрупненная схема алгоритма; определяются входные и выходные переменные, общие для данного уровня; разрабатываются схемы алгоритмов для реализации основных блоков программы, определяются входные и выходные переменные соответствующего уровня.

Этот процесс продолжается, пока схема алгоритма не будет доведена до базовых структур соответствующего языка программирования.

Базовые структуры для всех языков программирования в основном совпадают. Отличия могут заключаться только в форматах используемых операторов. Основные базовые элементы, используемые для изображения схем алгоритмов, приведены в табл. 2.1.

Для каждого проекта составляется описание переменных по форме, приведенной в табл. 1.1. Это позволит избежать ошибок при программировании, например, повторного использования имен переменных, систематизировать исходные данные и результаты вычислений, облегчит составление отчетной документации.

Свойства формы и объектов управления, установленных на форме, описывается по форме, приведенной в табл. 1.2.

Принцип модульного проектирования заключается в следующем: при программировании различных задач всегда встречаются фрагменты программы, которые используются в теле программы неоднократно, например, вычисление факториала, вычисление производной, построение графика функции, ввод данных в массивы и так далее. В этих случаях такие типовые блоки программы могут быть оформлены в виде подпрограмм (процедур) или функций пользователя и использоваться не только в разрабатываемой программе, но и в других программах. Такие процедуры и функции следует объявлять в стандартных модулях.

При создании пользовательских процедур, ссылающихся на объекты, установленные на форме, или саму форму, целесообразно использовать переменные типа Object. Рассмотрим порядок объявления и использования переменных типа Object на примере процедуры построения опор.

1. Объявим в разделе общие модуля переменную Obj:

```
Public Obj As Object
```

2. В глобальной процедуре модуля используем объект Obj для описания процедуры построения опоры. Опора в последующем будет строиться непосредственно в форме.

```
Public Sub Pscop ( ) 'процедура построения опоры в модуле
```

```
Obj.PSet (x0, y0), QBColor(6) 'коричневый
```

```
Obj.DrawWidth = 2
```

```
Obj.Line -(x0 - 10, y0 - 10), QBColor(6)
```

```
Obj.Line -(x0 + 10, y0 - 10), QBColor(6)
```

```
Obj.Line -(x0, y0), QBColor(6)
```

```
End Sub
```

Таблица 2.1

Основные графические элементы схем алгоритмов

Наименование	Обозначение	Функция
Процесс		Выполнение операций или группы операций, в результате которых изменяется значение, форма представления или расположение данных.
Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий.
Модификация		Выполнение операций, меняющих команды, или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы).
Предопределенный процесс		Использование созданных ранее или отдельно описанных алгоритмов или программ.
Ввод – вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).
Линии потока		Указание на последовательность связи между символами. Можно без стрелки, если линия направлена слева направо или сверху вниз, со стрелкой в остальных случаях.
Соединитель		Указание на связь между прерванными линиями потока, соединяющими операторы в пределах листа.
Соединитель		Указание на связь между прерванными линиями потока, соединяющими операторы на разных листах.
Пуск-останов		Начало, конец, прерывание процесса обработки данных или выполнения программы.
Комментарий		Связь между элементами схемы с пояснениями

3. В обработчике события Load формы, в которой предполагается использовать процедуру построения опоры, присвоим переменной Obj имя формы.

```
Private Sub Form_Load() ' обработчик события Load формы Статика
    Set Obj = frmStatika
```

```
...
End Sub
```

4. В обработчике события Click командной кнопки Построение вызовем процедуру построения опор.

```
Private Sub cmdPostroenie_Click() ' обработчик события кн. Построение
    Cls
    Ft = Val(txtStatic.Text) ' угол в градусах
    fi = Ft / 180 * pi ' угол в радианах
    Raschet ' Вызов глобальной процедуры расчет
    Postroenie ' Вызов глобальной процедуры построения механизма
    Pspop ' Вызов глобальной процедуры построения опор
```

```
End Sub
```

Программа для курсовой работы должна включать в общем случае следующие программные модули (блоки): паспорт программы, блок ввода данных, меню программы, блок вычислений, блок вывода результатов на печать, блок построения графиков и диаграмм, блоки сохранения результатов вычислений на диске и загрузки результатов предыдущих вычислений с диска, блок выхода из программы.

Паспорт программы содержит наименование программы, ее назначение и область применения, сведения об авторе и дате разработки или последней модификации, адрес и номера телефонов, факсов для связи с автором или фирмой – разработчиком программы, обладателем лицензии.

Блок ввода данных содержит объявления типов переменных, массивов, присвоение начальных значений переменным, элементам массивов, строковым константам.

Блок вычислений служит для реализации вычислений в соответствии с разработанной математической моделью. Он может содержать также модули сохранения результатов расчетов в файле на диске, вывода предварительных и итоговых результатов на экран в табличной форме, возврата в главное меню.

Блоки вывода результатов на форму и вывода результатов на печать должны содержать подменю и позволять выводить результаты вычислений на экран или на печатающее устройство в табличной форме и/или в виде графиков и диаграмм по желанию пользователя. Выводиться должны текущие результаты или результаты последних вычислений без предварительного расчета. Необходимо предусмотреть защиту от ошибочных ситуаций, например отсутствия данных.

Перед началом работы необходимо первоначально определить состав и назначение форм и модулей. Составить их эскизы и определить функции, выполняемые ими. Если форм в проекте больше одной, то для обеспечения взаимодействия форм необходимо включить в проект MDI- форму и разработать для нее пользовательское меню.

Стандартные модули, как известно, не содержат визуальных элементов и предназначены для объявления глобальных переменных, процедур и функций пользователя.

Форма содержит визуальные элементы и программный модуль, который содержит обработчики событий элементов управления и формы. В разделе Общие (General) объявляются переменные уровня формы, процедуры и функции пользователя, доступные только в данной форме.

3. Типовые алгоритмы и процедуры

3.1. Паспорт программы

Паспорт программы можно выполнить различными способами: написать в объекте Надпись (Label) или в окне ввода (TextBox), оформить в текстовом процессоре Word и затем вставить его в форму или объект PictureBox, используя свойство Picture или механизм вставки и внедрения OLE.

Пример 3.1. Паспорт программы.

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра информатики и прикладной математики
КУРСОВАЯ РАБОТА

тема: Моделирование изображения детали

Программа позволяет строить аксонометрическое изображение детали, и ее проекции, изменять масштаб изображения, перемещать деталь и поворачивать изображение на произвольный угол в режиме диалога с пользователем. Угол вводится в градусах

Исполнитель: Баценко Д.Л. гр.Т-39

Дата сдачи работы: 1.06.97

Брест 2009

3.2. Исследование функции на отрезке

Исследование функции на отрезке включает следующие этапы (процедуры):

1. Анализ функции. Определение области определения функции.
2. Поиск точек разрыва функции, точек пересечения ее с осями координат и вертикальных асимптот, если они существуют.
3. Установление четности (нечетности), периодичности функции.
4. Определение критических точек. К критическим точкам относятся граничные точки и точки экстремумов.
5. Исследование функции на монотонность.
6. Определение интервалов выпуклости и вогнутости, точек перегиба.
7. Поиск асимптот графика функции.
8. Построение графика функции.

Первый этап неформализованный. По виду функции устанавливают область ее определения и точки разрыва, а также периодичность функции. Остальные этапы могут быть выполнены с помощью программы.

Вопросы исследования функций подробно рассмотрены в учебной литературе по высшей математике, например [4]. Поэтому рассмотрим лишь один вопрос - численное определение производной.

Производная от функции может быть определена численными методами [6, 8]. Известно, что

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + o(x) \quad (3.1)$$

Отсюда вытекает способ численного дифференцирования. Если заменить предел Δx его конечным значением h , то получим приближенные формулы для вычисления первой и второй производных:

$$f'(x) \approx \frac{f(x+h) - (f(x-h))}{2h} \quad (3.2)$$

$$f''(x) \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} \quad (3.3)$$

Эти выражения представляют собой усеченные интерполяционные многочлены (многочлен Стирлинга). Одной из серьезных проблем в данном случае является выбор величины шага h . При уменьшении шага уменьшается ошибка усечения, но возрастает ошибка округления при вычислении производной. Поэтому стремятся выбрать оптимальную величину шага, при которой ошибка усечения и ошибка округления будут примерно равны. Для формулы (3.2) оптимальный шаг определяется из выражения [8]:

$$h = \sqrt[3]{\frac{3\varepsilon}{M_3}} \quad \text{или} \quad \frac{1}{6h} |\Delta^3 y| \approx \frac{1}{2} \frac{\varepsilon}{h} \quad (3.4)$$

где h - шаг, $\Delta^3 y$ - конечная разность 3-го порядка, ε - абсолютная погрешность вычисления функции, M_3 - максимальное значение конечной разности 3-го порядка. Таким образом, ошибка усечения равна примерно половине ошибки округления. Полная погрешность не превышает при этом $0.5 \varepsilon/h$.

Для формулы (3.3) величина шага определяется из следующего соотношения

$$\frac{1}{12h^2} |\Delta^4 y| \approx 4 \frac{\varepsilon}{h^2} \quad (3.5)$$

Ввиду сложности самой процедуры отыскания оптимальной величины шага при вычислении производной функции в курсовом проекте рекомендуется принять значение шага в пределах от 0.1 до 0.01 без вычислений.

Чтобы определить точки пересечения графика функции с осями координат, необходимо принять $y=0$ для определения точек пересечения с осью X и $x=0$ - для определения точек пересечения с осью Y .

При определении наличия четности (нечетности) следует исходить из того, что у четной функции $F(x)=F(-x)$ для любого x .

Для исследования функции по п. 5 - 6 необходимо протабулировать на заданном отрезке функцию, ее первую и вторую производные. Результаты табулирования рекомендуется записать вначале в массив (массивы).

Вычисление пределов для определения асимптот можно проводить в одной из математических систем, например, Matlab.

3.3. Решение алгебраических и трансцендентных уравнений

Алгоритм решения алгебраических и трансцендентных уравнений включает два самостоятельных этапа:

1. Отделение корней на заданном интервале или области определения функции;
2. Уточнение значения корней на отрезках отделения.

Отделение корней уравнения

Общая схема отделения корней уравнения на заданном интервале включает следующую последовательность операций:

- определить критические точки: точки экстремумов и граничные точки отрезка;
- вычислить значения функции в критических точках и составить таблицу смены знака функции;
- определить отрезки отделения корня.

При решении задачи с помощью ЭВМ алгоритм должен быть несколько иным: выполнить табулирование функции на заданном интервале. В процессе табулирования необходимо:

- на каждом шаге вычислять значения функции в текущей $F(x_i)$ и следующей $F(x_{i+1})$ точках и проверять условие знакопостоянства функции $F(x_i) \cdot F(x_{i+1}) > 0$;
- зафиксировать точки x_i и x_{i+1} , в интервале которых произведение функций будет отрицательным или равным нулю. Эти точки и будут границами отрезков отделения. Так как отрезков отделения может быть несколько, то их необходимо записать в массив.

Уточнение значения корня на отрезке отделения

Уточнение значения корня на отрезке отделения осуществляется одним из методов одномерной поисковой оптимизации, например, методом итераций, методом дихотомии или методом Ньютона (метод касательных). Алгоритмы реализации этих методов рассмотрены в [6,2,3].

Самым простым методом уточнения значения корня является метод деления отрезка пополам (метод дихотомии):

1. Вычислить значение функции в точке a : $Y_a = F(a)$.
2. Найти середину отрезка $[a, b]$ - точку c . Вычислить значение функции в точке c : $Y_c = F(c)$.
3. Проверить длину отрезка $[a, b]$. Если длина отрезка $[a, b]$ меньше или равна требуемой точности ε , то точка c есть корень. Вывести результат на печать. Иначе уточнить, где находится корень по отношению к точке c : слева от точки c или справа от точки c .
4. Если произведение значений функций Y_a и Y_c меньше нуля, то корень находится слева от точки c , тогда точку b надо перенести в точку c : $b = c$, - и повторить операции по п. п. 2, 3, 4. Иначе корень находится справа от точки c . Тогда необходимо точку a перенести в точку c : $a = c$, а также $Y_a = Y_c$, так как значение функции в точке c нам уже известно, и повторить операции по п. п. 2, 3, 4.

Примечание. В качестве условия окончания процесса деления отрезка можно использовать выражение $Y_c \leq 0$, однако точность поиска значения аргумента будет в этом случае сильно зависеть от вида функции и принятого значения шага табулирования функции при отделении корня. Для медленно изменяющихся функций длина отрезка уточнения корня будет больше, чем для быстро изменяющихся функций.

3.4. Вычисление определенного интеграла

При выполнении курсовой работы по данной теме необходимо вычислить определенный интеграл с заданной точностью двумя или тремя способами и сравнить полученные результаты по точности достигаемых результатов, при заданном числе шагов, или по числу шагов, необходимых для достижения заданной точности. Для вычисления интеграла можно использовать методы средних, правых или левых прямоугольников, метод трапеций или метод Симпсона, которые подробно описаны в литературе [6,2,3].

Вычисление интегралов осуществляется по следующим формулам:

для метода средних прямоугольников

$$\int_a^b f(x) dx = \sum_{i=1}^n h y_i(x_{i-1/2}), \quad (3.6)$$

где x_i - текущее значение аргумента, $(x_{i-1/2})$ - значение аргумента в середине отрезка разбиения (полуцелая точка); y_i - значение функции при соответствующем значении аргумента; h - шаг разбиения отрезка интегрирования (шаг разбиения отрезка интегрирования выбирается, обычно, постоянным), $h = (b-a)/N$, N - число отрезков разбиения интервала интегрирования. Начальное значение N должно быть четным, например, 4.

Для метода трапеций

$$\int_a^b f(x) dx = \left(\frac{1}{2}(y_0 + y_n) + \sum_{i=1}^{n-1} y_i\right) * h \quad (3.7)$$

Для метода Симпсона

$$\int_a^b f(x) dx \approx \frac{h}{3} [y_0 + 4(y_1 + y_3 + \dots + y_{n-1}) + 2(y_2 + y_4 + \dots + y_{n-2}) + y_n] \quad (3.8)$$

Погрешность численного интегрирования R равна сумме погрешностей численного интегрирования R_i на каждом отрезке $[x_{i-1}, x_i]$, которые определяются по следующим формулам:

$$R_i = \frac{1}{24} h_i^3 f^{(4)}(x_{i-1/2}) \quad \text{для формулы средних прямоугольников} \quad (3.9)$$

$$R_i = \frac{1}{12} h_i^3 f^{(2)}(x_i) \quad \text{для формулы трапеций} \quad (3.10)$$

$$R_i = \frac{h_i^4}{180} f^{(4)}(x) \quad \text{для формулы Симпсона} \quad (3.11)$$

Вычисление погрешности интегрирования можно оформить в виде функций, определяемых пользователем. Однако вычисление погрешностей интегрирования по данным формулам достаточно громоздко и трудоемко. Поэтому на практике для определения погрешности вычисления интеграла применяют метод двойного прохода и эмпирические формулы [8]:

$$(I_1 - I_2) / 3 < \epsilon, \quad \text{для метода трапеций} \quad (3.12)$$

$$(I_1 - I_2) / 15 < \epsilon, \quad \text{для метода Симпсона} \quad (3.13)$$

Здесь I_1 и I_2 - значение интеграла на текущем и предыдущем шаге, соответственно.

Суть метода двойного прохода состоит в следующем. Вначале задают небольшое число отрезков разбиения (для данных методов число отрезков разбиения должно быть четным), вычисляют значение интеграла (I) и запоминают его ($I_1 = I$). Увеличивают число отрезков разбиения в два раза и снова вычисляют значение интеграла (I). После этого вычисляют значение ошибки интегрирования по формулам (3.12, 3.13). Вычислительный процесс прекращается, когда величина ошибки интегрирования будет меньше или равна заданному значению.

При вычислении определенного интеграла с заданной точностью необходимо организовать два вложенных цикла. Внутренний цикл обеспечивает вычисление площади криволинейной трапеции при текущем значении шага разбиения h отрезка интегрирования, а внешний цикл служит для вычисления ошибки численного интегрирования и обеспечивает достижения требуемой точности.

3.5. Построение изображения детали

Курсовая работа на данную тему не содержит обычно математических моделей и вычислительных алгоритмов и с этой точки зрения является достаточно простой. При выполнении данной работы студент должен продемонстрировать умение использовать графические возможности языка программирования, создания анимационных эффектов. Программа построения изображения детали должна содержать следующие подпрограммы:

- изображение детали в аксонометрии;
- изображение детали в аксонометрии с разрезом;
- проекции детали;
- перемещение детали в указанную точку экрана;
- поворот детали на заданный угол;
- изменение масштаба изображения.

Для обеспечения универсальности программы ввод данных о размерах детали рекомендуется организовать с помощью окна диалога, данные записывать в массивы переменных и сохранять их в файле на диске.

Все подпрограммы по изменению масштаба, повороту, перемещению детали должны иметь режим диалога для запроса требуемых параметров и контроля выхода изображения за границы экрана. Каждая экранная форма должна содержать текстовое сообщение о порядке действий пользователя и, при необходимости, подменю для управления изображением на экране (контекстное меню).

Порядок создания анимационных эффектов рассмотрен в [3].

Одним из сложных алгоритмов является алгоритм поворота детали на заданный угол с переносом в указанную точку. Ниже приводится алгоритм поворота детали вокруг оси Z. Поворот детали вокруг осей X или Y требует более сложных алгоритмов.

Пример 3.2. Алгоритм поворота изображения на заданный угол и с переносом в заданную точку

```
500 Rem Алгоритм поворота изображения с переносом в заданную точку
510 Call Poworot : ' Вызов подпрограммы построения изображения
520 x1=190: y1=80: x2=400: y2=310: ' Ввод координат изображения на экране
530 REM Определение размерности массива
n=0: ' текущая переменная, размерность массива
CurrentX=100: CurrentY=100: PRINT "ждите, идет считывание изображения"
For i=x1 To x2
For j=y1 To y2
IF Point (i,j)=1 Then n=n+1
Next j
Next i
ReDim ax(n) As Integer, ay(n) As Integer
540 Rem Считывание данных в массив
l=0
For i=x1 To x2
For j=y1 To y2
IF Point (i,j)=1 Then l=l+1: ax(l)=i-x1:ay(l)=j-y1
Next j
Next i
550 Rem Воспроизведение изображения
Cls
x0=Val(InputBox("Введите координату X точки переноса"))
y0=Val(InputBox("Введите координату Y точки переноса"))
f1= Val(InputBox("Введите угол поворота в градусах"))
f=f1*pi/180 ' перевод градусов в радианы
Cls
Pset(x0,y0)
For i=1 To n
bx=ax(i): by=ay(i)
a=x0 + bx*cos(f)- by*sin(f)
b=y0 + bx*sin(f)+ by*cos(f)
Pset (a,b)
Next i
```

Изменение масштаба изображения реализуется легко путем переопределения математических координат с помощью оператора Scale (0,0)-(640*mx,200*my), где mx и my - масштабы по соответствующим осям.

Чаще всего расстояние между пикселями на экране монитора по горизонтали и по вертикали разные, поэтому при изображении дуг и окружностей возникают искажения. Чтобы избавиться от этих искажений, необходимо установить строгое соответствие между значениями размеров экрана по горизонтали и по вертикали: $X=4/3Y$. С учетом сказанного оператор Scale следует применять в следующем виде: $Scale(0,y)-(x,0)$.

В программе можно предусмотреть демонстрацию возможностей программы.

3.6. Моделирование движения механизма

Под моделированием движения механизма понимается воспроизведение положения механизма в последовательные моменты времени. При этом возможны две постановки задачи:

А. Моделирование движения механизма с целью определения зон действия звеньев механизма;

Б. Исследование кинематики управления механизмом.

А. Моделирование движения механизма с целью определения зон действия звеньев механизма

При выполнении курсового проекта на данную тему необходимо решить следующие вопросы:

1. Разработать математическую модель механизма.
2. Разработать алгоритмы, обеспечивающие:
 - воспроизведение положения механизма в исходном состоянии;
 - воспроизведение положения механизма в нескольких промежуточных точках;
 - определение зон движения механизма;
 - определение области, занимаемой механизмом в процессе движения;
 - построение траектории движения заданной точки.

При наличии у механизма только ползунов или катков, у которых звенья механизма присоединены к центру катка, никаких принципиальных трудностей в разработке математической модели не возникает. Для вывода математических зависимостей используются теоремы синусов, косинусов, свойства прямоугольных треугольников, подобия фигур и тому подобные соотношения, известные из школьного курса математики.

Указания к решению задач, подобных механизму, изображенному на рис.9 приложения 2

Сложнее обстоит дело с механизмами, имеющими катки, у которых звенья механизма присоединены к точкам на поверхности катка. Рассмотрим в качестве примера механизм, представленный на рис. 3.1.

1. Выберем масштаб изображения из условий, что размер по оси X больше размера по оси Y в 4/3 (иначе возникают нелинейные искажения, связанные с параметрами экрана) и изобразим механизм в масштабе.

2. Определим параметры узлов и составим таблицу исходных данных:

№ узла	X	Y	L	ϕ
O	200	150	L1=90	55
A	Xa	Ya	L2=240	
B	Xb	Yb	R=75	
C	150	Yc	L3=150	
D	Xd	170		

3. Составим математическую модель

$$x_a = x_0 + l_1 \cos(\phi_1) \quad (3.14)$$

$$y_a = y_0 + l_1 \sin(\phi_1) \quad (3.15)$$

На данном рисунке каток имеет две степени свободы. Он может вращаться вокруг своей оси и перемещаться в горизонтальном направлении. В исходных данных высота расположения катка и его радиус заданы, а следовательно и вертикальная координата центра катка Y_d известна. Горизонтальное положение катка X_d - неизвестно. Наибольшие трудности возникают при определении значения угла φ_2 .

Для определения значения угла φ_2 необходимо составить и решить систему из пяти уравнений:

$$\begin{cases} x_b = x_a + l_2 \cos \varphi_2 & (3.16) \\ x_b = x_d + r \cos \varphi_3 & (3.17) \\ y_b = y_a + l_2 \sin \varphi_2 & (3.18) \\ y_b = y_d + r \sin \varphi_3 & (3.19) \\ (x_b - x_d)^2 + (y_b - y_d)^2 = r^2 & (3.20) \end{cases}$$

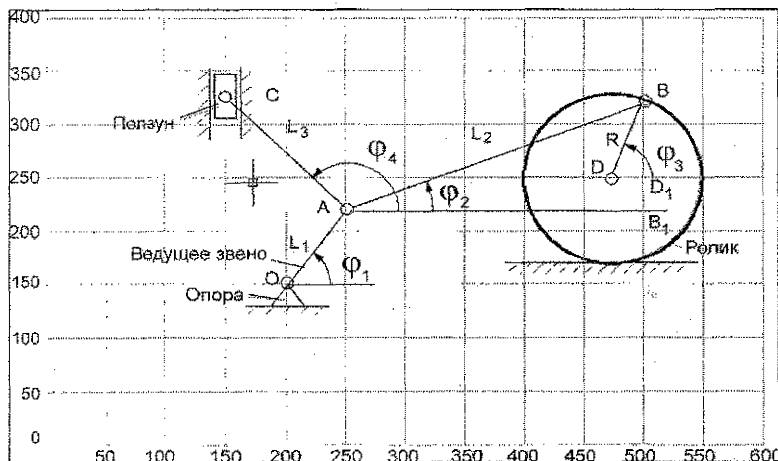


Рис. 3.1. Механизм

Здесь l_2 - длина шатуна AB, r - радиус катка. Первые четыре уравнения получают из рассмотрения прямоугольников ABB_1 и DBD_1 . Пятое уравнение составляется из следующих соображений: координаты точки B лежат на поверхности окружности, поэтому можно записать уравнение окружности с центром в точке D и радиусом, равным радиусу катка r .

Из уравнений 3.16 и 3.17 находим значение X_d :

$$x_d = x_a + l_2 \cos \varphi_2 - r \cos \varphi_3 \quad (3.21)$$

Из уравнений 3.18 и 3.19 находим значение угла φ_3 :

$$\varphi_3 = \arcsin((y_a + l_2 \sin \varphi_2 - y_d) / r) \quad (3.22)$$

Заменив в выражении (3.19) угол φ_3 на его значение из (3.22) и подставив значения x_b (3.18), x_d и y_b (3.19) в (3.22) получим уравнение

$$\begin{aligned} & ((x_a + l_2 \cos \varphi_2 - (x_a + l_2 \cos \varphi_2 - r \cos(\arcsin((y_a + l_2 \sin \varphi_2 - y_d) / r)))^2 + \\ & + (y_a + l_2 \sin \varphi_2 - y_d)^2) - r^2 = 0. \end{aligned} \quad (3.23)$$

В выражении 3.23 один неизвестный параметр - угол φ_2 . Найти значение угла φ_2 из данного уравнения можно с помощью одного из методов параметрической оптимизации,

например методом итераций. Фрагмент программы, реализующий данный метод приведен ниже.

Для вычисления координат точки С следует воспользоваться теоремой Пифагора, так как после определения координат точки А будут известны катет и гипотенуза треугольника АСС₁.

Пример 3.3. Программа для подбора значения угла φ_2

МЕХанизм построения механизма

SCREEN 9

Y=400: X=4/3*Y

SCALE (0,Y) - (X,0): REM Переопределение координат на математические

REM Ввод исходных данных

pi=ATN(1)*4

e=1 *Точность поиска

x0=100: y0=100

yd=150

l1=40: l2=175:l3=50

i1=2*pi/3

REM Цикл изменения значения угла φ_1 .

for i=0 to i1 step 0.3

f1=i

xa=x0+l1*cos(f1): ya=y0+l1*sin(f1)

s1=e+1:df=.01: f2=pi/10

REM Цикл изменения значения угла φ_2

while abs(s1)>e and f2<pi/2

a=(ya-yd+l2*sin(f2))/l3

b=a*l3

if (1-a^2)<=0 then goto m1

s1=l3^2-((xa+l2*cos(f2)-(xa+l2*cos(f2)-l3*cos(atn(a/sqr(1-a^2))))^2+b^2)

goto m2

m1:

?нет решения, F2="f2*180/pi,"F1="f1*180/pi

m2:

ft=f2

f2=f2+df

wend

REM Расчет координат

f2=ft

a=(ya-yd+l2*sin(f2))/l3

if (1-a^2)<0 then goto m3

f3=atn(a/sqr(1-a^2))

xb=xa+l2*cos(f2)

yb=ya+l2*sin(f2)

xd=xb-l3*cos(f3)

m3: REM Построение рисунка

circle (x0,y0),4

line (x0,y0)-(xa,ya)

circle (xa,ya),4

line (xa,ya)-(xb,yb)

circle (xb,yb),2

circle (xd,yd),l3

next i

end

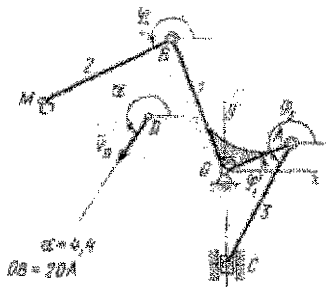


Рис. 3.2. Механизм

Указания к решению задач, подобных механизму, приведенному на рис. 1 приложения 3

В этих задачах (рис. 3.2.) необходимо в исходных данных указать положение точки М. Написать уравнение прямой, проходящей через эту точку, в заданном направлении:

$$y=kx+b \quad (3.24)$$

Для точки М в этом уравнении известны значения X и Y, $k=\operatorname{tg}(\alpha)$. Поэтому можно вычислить значение свободного члена b.

Точка М в процессе движения должна постоянно находиться на заданной прямой. Ее уравнение также будет иметь вид $Y_m=kX_m+b$, в котором будет две неизвестных X_m и Y_m . Кроме того, в любой момент времени ее расстояние от точки В должно быть равно длине звена ВМ, длина которого может быть определена через координаты точек В и М по теореме Пифагора. Таким образом получаем систему уравнений с двумя неизвестными:

$$\begin{cases} Y_m=kX_m+b \\ BM^2=(X_m-X_b)^2+(y_m-Y_b)^2 \end{cases}$$

которая легко решается.

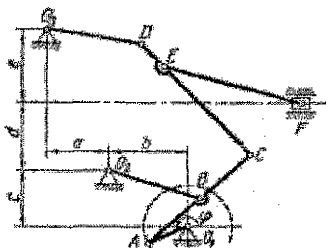


Рис. 3.3. Механизм

Указания к решению задач приложения 5

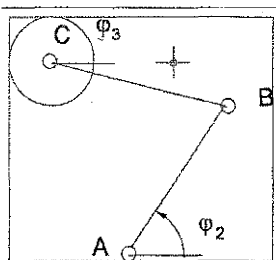


Рис.3.4. Фрагмент механизма рис. 3.3.

Рассмотрим порядок нахождения координат точки В на рис. 3.3. У точки В неизвестны ни x ни y. Но известны координаты точек А и О₂.

Определим координаты точки В через координаты точек А и С (рис. 3.4).

$$x_b = x_c + BC \cos(\varphi_3) \quad (1)$$

$$y_b = y_c - BC \sin(\varphi_3) \quad (2)$$

$$x_b = x_a + AB \cos(\varphi_2) \quad (3)$$

$$y_b = y_a + AB \sin(\varphi_2) \quad (4)$$

Решим эту систему уравнений.

Освободимся от переменных x_b и y_b .

Из уравнений (1) и (3) получим

$$BC \cos(\varphi_3) = x_a - x_c + AB \cos(\varphi_2) \quad (5)$$

Из уравнений (3) и (4) получим

$$BC \sin(\varphi_3) = y_a - y_c + AB \sin(\varphi_2) \quad (6)$$

Возведем выражения (5) и (6) в квадрат и сложим их, тогда получим следующее выражение:

$$BC^2 \cos^2(\varphi_3) + BC^2 \sin^2(\varphi_3) = (x_a - x_c + AB \cos(\varphi_2))^2 + (y_a - y_c + AB \sin(\varphi_2))^2 \quad (7)$$

Вспользуемся известным выражением $\cos^2(x) + \sin^2(x) = 1$, раскроем скобки и приведем подобные члены:

$$AB^2 - BC^2 + (x_a - x_c)^2 + (y_a - y_c)^2 + 2AB(x_a - x_c) \cos(\varphi_2) + 2AB(y_a - y_c) \sin(\varphi_2) = 0 \quad (8)$$

Введем вспомогательную переменную $U = (AB^2 - BC^2 + (x_a - x_c)^2 + (y_a - y_c)^2) / (2AB)$, повторно разделим неизвестные переменные $\cos(\varphi_2)$ и $\sin(\varphi_2)$ и повторим операцию по п. 6 и 8. В результате получим:

$$\begin{aligned} U^2 + 2U(x_a - x_c) \cos(\varphi_2) + (x_a - x_c)^2 \cos^2(\varphi_2) &= (y_a - y_c)^2 \sin^2(\varphi_2) \\ U^2 - (y_a - y_c)^2 + 2U(x_a - x_c) \cos(\varphi_2) + \{(x_a - x_c)^2 + (y_a - y_c)^2\} \cos^2(\varphi_2) &= 0 \end{aligned} \quad (9)$$

Выражение (9) является квадратным уравнением относительно $\cos(\varphi_2)$ и может быть решено аналитически. После вычисления значения $\cos(\varphi_2)$ находим значение $\sin(\varphi_2)$ и координаты точки В согласно выражениям (3), (4).

Аналогично вычисляются значения координат точки D на рис. 3.3.

Другой подход к решению данной задачи состоит в использовании свойства точки, лежащей на окружности. Геометрическое место точек, лежащих на окружности, центр которой не совпадает с началом координат, определяется выражением:

$$(x - x_0)^2 + (y - y_0)^2 = R^2. \quad (3.25)$$

Точка В принадлежит сразу двум окружностям с координатами в точках А и С. Записав соответствующие выражения для этих окружностей, получим следующую систему уравнений:

$$(x_b - x_c)^2 + (y_b - y_c)^2 = BC^2 \quad (3.26)$$

$$(x_b - x_a)^2 + (y_b - y_a)^2 = AB^2$$

Решая эту систему уравнений, приходим к тем же результатам.

Б. Исследование кинематики управления механизмом

Описание задания

Цель расчета - освоение методики аналитического исследования кинематики управляемого движения манипулятора.

Рассматриваемый манипулятор является плоским механизмом с двумя степенями свободы. Следовательно, его захвату, точке М, разрешается произвольное движение в плоскости по двум координатам. Управление должно совместить захват с движущейся деталью, точкой D. Варианты кинематических схем манипуляторов представлены в приложении 3. Деталь D движется с постоянной скоростью V_0 в указанном на рисунке направлении. Координаты точки D изменяются по закону

$$x_D = x_D(0) + V_{Dx}t, \quad y_D = y_D(0) + V_{Dy}t \quad (3.27)$$

Управление движением захвата М осуществляется по линейной комбинации рассогласований координат точек D и М, а также их производных. Рассогласование координат точек D и М в момент времени $t = \tau$ должно составлять величину δ от начальных рассогласований.

Исходные данные определяются по формулам (3.28) и таблице приложения 4.

$$r_i = r_{1r} + 0,001n, \quad \tau_i = \tau_i + 0,001n$$

$$\varphi_i(0) = \varphi_{1r} + 0,01n, \quad \varphi_i(0) = \varphi_{1r} + 0,01n \quad (i=2, 3); \quad (3.28)$$

$$\varphi_D = \varphi_{Dr} - 0,003n, \quad \tau = 1, 2(1+0,02n).$$

Здесь r_i - длина i-го звена, $OA=r_1$, $AB=r_2$, $BC=r_3$, $BM=2BC$; φ_i - угловое положение звеньев механизма.

Требуется: выбрать управление, решающее поставленную задачу.

Методику расчета рассмотрим на примере.

Постановка задачи

Управление манипулятором (рис. 3.5) должно обеспечить за время t сближение захвата М с движущейся деталью D. Деталь движется прямолинейно с постоянной скоростью \vec{v} в указанном на рис. 3.5 направлении. Начальное положение манипулятора задано углами поворота звеньев $\varphi_i(0)$,

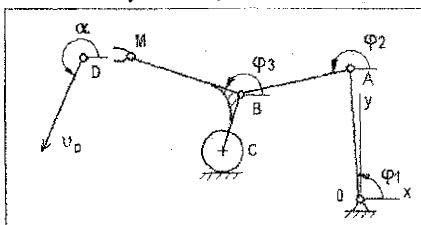


Рис.3.5. Манипулятор

$\varphi_2(0)$, $\varphi_3(0)$. К моменту времени $t=\tau$ требуется относительная точность δ совмещения точек M и D . Управление манипулятором осуществляется по линейной комбинации рассогласований и их производных.

Числовые значения исходных данных: $v_D=0,304$ м/с – скорость детали D ; $\alpha=4,35$ рад – направление движения детали D ; $OA=0,953$ м; $AB=0,847$ м; $BC=0,457$ м; $BM=2BC$ м; $\varphi_1(0)=1,63$ рад; $\varphi_2(0)=3,37$ рад; $\varphi_3(0)=2,87$ рад; $x_D(0)=-2,16$ м; $y_D(0)=1,18$ м, x_D, y_D – координаты детали D ; $\delta=0,01$; $\tau=1,37$ с; $\Delta t=0,057$ с.

Здесь α – угол между осью x и вектором v_D , Δt – шаг интегрирования.

Требуется:

1. Составить уравнения управляемого движения точки M , уравнения углового движения звеньев манипулятора и уравнение для скорости точки C .

2. Выбрать параметры управления, обеспечивающего сближение точек M и D с заданной точностью.

3. Проинтегрировать с помощью ЭВМ уравнения движения на интервале времени τ .

4. Построить траектории сближения точек M и D и графики $\varphi_1(t)$, $\omega_{1z}(t)$, $v_{Cx}(t)$.

Составление уравнений движения

Уравнения движения детали D имеют вид

$$\begin{aligned} x_D &= x_D(0) + v_{Dx}t; & v_{Dx} &= v_D \cos \alpha = -0,108 \text{ м/с;} \\ y_D &= y_D(0) + v_{Dy}t; & v_{Dy} &= v_D \sin \alpha = -0,284 \text{ м/с.} \end{aligned} \quad (3.29)$$

Предположим, что координаты захвата M известны в процессе движения. Тогда можно вычислить рассогласования координат точек D и M :

$$\Delta x = x_D - x_M; \quad \Delta y = y_D - y_M. \quad (3.30)$$

Управление движением захвата осуществляется по сигналам управления u_x, u_y , образованным линейной комбинацией рассогласований и их производных:

$$u_x = \Delta x + T^* \frac{d}{dt} \Delta x; \quad u_y = \Delta y + T^* \frac{d}{dt} \Delta y. \quad (3.31)$$

При управлении с большими коэффициентами усиления k с погрешностью порядка $1/k$ выполняются соотношения

$$u_x = 0; \quad u_y = 0. \quad (3.32)$$

Тогда из (3.30, 3.31, 3.32) получим:

$$\frac{dx_M}{dt} = v_{Dx} + \frac{1}{T^*} (x_D - x_M); \quad \frac{dy_M}{dt} = v_{Dy} + \frac{1}{T^*} (y_D - y_M) \quad (3.33)$$

Манипулятор является механической системой с двумя степенями свободы, поэтому движение по двум координатам x_M, y_M , найденное по (3.33), однозначно определяет движение всех его звеньев. Кинематические уравнения, описывающие изменение углов поворота и угловых скоростей звеньев, составляются по методике, известной из курса теоретической механики.

Выбор коэффициента управления T^*

Из (3.31) с учетом (3.32) получим:

$$T^* \frac{d}{dt} \Delta x + \Delta x = 0; \quad T^* \frac{d}{dt} \Delta y + \Delta y = 0 \quad (3.34)$$

Решение этих уравнений однотипно:

$$\Delta x = \Delta x(0) e^{-\frac{t}{T^*}}; \quad \Delta y = \Delta y(0) e^{-\frac{t}{T^*}}. \quad (3.35)$$

По условию, к концу интервала времени τ рассогласования $\Delta x, \Delta y$, должны составлять величину δ от начальных рассогласований.

Из (3.35) имеем $\delta = \Delta x / \Delta x(0) = e^{-\frac{\tau}{T^*}}$, откуда $T^* = -\tau / \ln \delta$.

При заданных исходных данных $T^* = -1,37 / \ln 0,01 = 0,297$ с.

По определению

$$v_{Mx} = \frac{dx_M}{dt}; \quad v_{My} = \frac{dy_M}{dt} \quad (3.36)$$

Если систему уравнений (3.33) и кинематических уравнений движения звеньев привести к форме Коши, то она будет иметь вид

$$\begin{aligned} \dot{x}_M &= v_{Mx}(x_n, t) & \dot{y}_M &= v_{My}(y_n, t) \\ \dot{\varphi}_i &= \omega_i(v_{Mx}, v_{My}, t) \quad (i = 1, 2, 3). \end{aligned} \quad (3.37)$$

Эти уравнения манипулятора, являющегося системой с двумя степенями свободы, записаны в избыточном наборе пяти переменных — $x_M, y_M, \varphi_1, \varphi_2, \varphi_3$. Отсюда следует, что из начальных значений этих переменных независимо могут задаваться только два. В таблице приложения 4 независимыми задаются величины $\varphi_1(0), \varphi_2(0)$, значения $\varphi_3(0)$, указанные в таблице, вычислены по $\varphi_1(0), \varphi_2(0)$ для рассматриваемой конструктивной схемы манипулятора (рис. 3.5). Значения $x_M(0), y_M(0)$ следует найти самостоятельно по значениям $\varphi_1(0), \varphi_2(0), \varphi_3(0)$.

Из (3.29), (3.33) и (3.36) получим:

$$\begin{aligned} v_{Mx} &= v_{Dx} + (x_D(0) + v_{Dx}t - x_M) / T^*; \\ v_{My} &= v_{Dy} + (y_D(0) + v_{Dy}t - y_M) / T^*; \end{aligned} \quad (3.38)$$

Угловое движение звеньев манипулятора и скорость точки C однозначно определяются движением точки M и внешними связями, налагаемыми в точках O и C . Составим выражения для проекций скоростей точек C и M .

В соответствии с графом $C \xrightarrow{1} B \xrightarrow{2} M$ запишем:

$$\begin{aligned} v_{Mx} &= v_{Cx} - \omega_{3z} r_3 \sin(\varphi_3 - \pi/2) - \omega_{3z} 2r_3 \sin(\varphi_3) \\ v_{My} &= \omega_{3z} r_3 \cos(\varphi_3 - \pi/2) + \omega_{3z} 2r_3 \cos(\varphi_3) \end{aligned} \quad (3.39)$$

В соответствии с графом $O \xrightarrow{1} A \xrightarrow{2} B \xrightarrow{3} C$

$$\begin{aligned} v_{Cx} &= -\omega_{1z} r_1 \sin \varphi_1 - \omega_{2z} r_2 \sin \varphi_2 - \omega_{3z} r_3 \sin(\varphi_3 + \pi/2); \\ v_{Cy} &= -\omega_{1z} r_1 \cos \varphi_1 + \omega_{2z} r_2 \cos \varphi_2 + \omega_{3z} r_3 \cos(\varphi_3 + \pi/2) = 0. \end{aligned} \quad (3.40)$$

Из уравнения (3.39), (3.40) получим

$$\begin{aligned} \omega_{3z} &= v_{My} / [r_3 (2 \cos \varphi_3 + \sin \varphi_3)]; \\ v_{Cx} &= v_{Mx} + \omega_{3z} r_3 (2 \sin \varphi_3 - \cos \varphi_3); \\ \omega_{1z} &= (v_{Cx} \cos \varphi_2 + \omega_{3z} r_3 \cos(\varphi_3 - \varphi_2)) / [r_1 \sin(\varphi_2 - \varphi_1)] \\ \omega_{2z} &= (\omega_{3z} r_3 \sin \varphi_3 - \omega_{1z} r_1 \cos \varphi_1) / (r_2 \cos \varphi_2). \end{aligned} \quad (3.41)$$

Уравнения (3.41) дополним дифференциальными соотношениями

$$\dot{\varphi}_1 = \omega_{1z}, \quad \dot{\varphi}_2 = \omega_{2z}, \quad \dot{\varphi}_3 = \omega_{3z} \quad (3.42)$$

Решение задачи на ЭВМ и обработка результатов

Систему уравнений (3.34), (3.37), (3.38) интегрируем с помощью ЭВМ на интервале времени $\tau = 1,37$ с, используя конечно разностную схему Эйлера. Шаг интегрирования $\Delta t = 0,057$ с

Начальные условия по переменным $\varphi_1, \varphi_2, \varphi_3$ приведены в исходных данных, а по переменным x_M, y_M вычисляются по данным длин звеньев механизма и рисунку (рис. 3.5):

$$\begin{aligned} x_M &= r_1 \cos \varphi_1 + r_2 \cos \varphi_2 + 2 r_3 \cos \varphi_3; \\ y_M &= r_1 \sin \varphi_1 + r_2 \sin \varphi_2 + 2 r_3 \sin \varphi_3; \end{aligned} \quad (3.43)$$

Подставив в (3.38) числовые значения $r_i, \varphi_i(0)$, получим $x_M(0) = -1,76$ м, $y_M(0) = 1$ м.

Программа счета

Обозначение переменных в программе:

Переменные	t	Δt	x	y	φ_1	φ_2	φ_3	ω_{1z}	ω_{2z}	ω_{3z}	V_{Mx}	V_{My}	V_{Cx}
Обозначение	T	DT	X	Y	F1	F2	h3	OM1	OM2	OM3	VX	VY	VC

Начальные значения переменных $t, x, y, \varphi_1, \varphi_2, \varphi_3$, длины звеньев r_i , начальные значения координат $x_D(0), y_D(0)$, скорости V_{Dx}, V_{Dy} , параметр управления T^* вводятся как числовые константы с клавиатуры с помощью оператора InputBox, из файла последовательного доступа или с помощью окон вода (TextBox). На печать выводятся переменные $t, \Delta t, x, y, \varphi_1, \varphi_2, \varphi_3, \omega_{1z}, \omega_{2z}, \omega_{3z}, V_{Mx}, V_{My}, V_{Cx}$.

По результатам счета строятся графики функций $V_i(t), \omega_{iz}(t), V_{Cx}(t)$ и траектории сближения точек M и D .

Текст программы:

```

DT= 0,057 : T= 0.456 : x=-2.16 : y=1.18 : F1=1.63 : F2=3.37 : F3=2.87
VDx=-0.108 : VDy= 0.284 : XD0=-2.16 : YD0=1.18 : TU=0.297
R1=0.953 : R2=0.847 : R3=0.457
N=Int(T/DT)+1
For k=1 TO N
    VX=-VDx-(XD(0)+VDx*T+x)/TU
    VY=-VDy+(YD(0)-VDy*T-y)/TU
    OM3= VY/(SIN(F3)+2*COS(F3))/R3
    RO=R3*OM3
    VC=VX+RO*(2*SIN(F3)-COS(F3))
    OM1=(VC*COS(F3)+RO*COS(F3-F2))/SIN(F2-F1)/R1
    OM2=(RO*SIN(F3)-R1*OM1*COS(F1))/COS(F2)/R2
    x=x+VX*DT
    y=y+VY*DT
    F1=F1+OM1*DT
    F2=F2+OM2*DT
    F3=F3+OM3*DT
    Print T, x, y, VX, VY, VC, F1, F2, F3, OM1, OM2, OM3
    T=T+DT
Next k
    
```

3.7. Методические рекомендации по выполнению курсовой работы №2

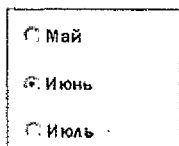
Курсовая работа № 2 выполняется в Электронной таблице с применением языка программирования Microsoft Visual Basic Application (VBA).

При запуске программы должен открываться лист с элементами управления проектом: окна диалога, меню и т. п. Процедуры построения графиков, оформления таблиц должны быть по возможности автоматизированы.

3.7.1. Разработка меню пользователя

Примеры разработки меню пользователя рассмотрим на примере меню для перехода по листам:

Меню с помощью Переключателей



Меню с помощью кнопок

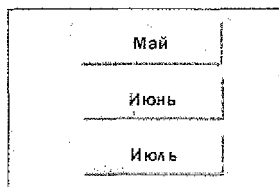


Рис. 3.6. Меню на основе переключателей

Рис. 3.7. Меню на основе кнопок

Создадим четыре листа. Листам присвоим имена Оглавление, Май, Июнь, Июль. На листе Оглавление разместим меню. Наиболее просто создать меню с помощью кнопок или переключателей (Рис. 3.6, 3.7):

- Перейдем на лист Оглавление;
- Откроем панель инструментов Элементы управления и установим на лист три переключателя. Настроим их свойства. Кнопка Свойства находится на панели элементов управления. Свойствам Caption присвоим имена "Май", "Июнь", "Июль". Свойству Value первого переключателя присвоим значение True;
- Перейдем в режим конструктора, активизировав верхнюю левую кнопку на панели элементов управления;
- Выделим первый элемент управления и активизируем на панели элементов управления кнопку "Исходный текст" – откроется окно редактор VBA. Внесем текст в обработчик события Click объекта OptionButton1, как показано на рис. 3.8;
- Внесем соответствующие команды в обработчики событий Click объектов OptionButton2 и OptionButton3;
- Для удобства поместим переключатели в рамку. Для этой цели можно воспользоваться инструментом Прямоугольник на панели инструментов Рисование. Прямоугольник переместим на нижний уровень (чтобы объекты были над прямоугольником).

```
Private Sub OptionButton1_Click()  
    Имя_листа = OptionButton1.Caption  
    Sheets(Имя_листа).Activate  
End Sub  
Private Sub OptionButton2_Click()  
    Имя_листа = OptionButton2.Caption  
    Sheets(Имя_листа).Activate  
End Sub  
Private Sub OptionButton3_Click()  
    Имя_листа = OptionButton3.Caption  
    Sheets(Имя_листа).Activate  
End Sub
```

Рис. 3.8. Исходный текст

- Выйдем из режима конструктора щелчком мыши по кнопке Режим конструктора. Теперь при щелчке мышью по одной из кнопок будет открываться соответствующий лист. Аналогично создается меню на основе кнопок CommandButton. Текст программы в обработчике событий будет иметь следующий вид:

```
Private Sub CommandButton1_Click()  
    Sheets("Май").Activate  
End Sub
```

3.7.2. Функции пользователя

Рассмотрим несколько вариантов создания и использования функций пользователя. Указанные варианты отличаются способом ввода данных.

Табулирование функции одной переменной. Вариант 1

Исходные данные внесены в ячейки B1, B2, B3. Для применения функции выделите область A5:B11 и вызовите функцию Tab11. В окна ввода введите адреса Хнач, Хкон, и шага. Нажмите комбинацию клавиш [Ctrl+Shift+Enter]. Функция возвращает значения аргумента и функции. Если размер выделенной области мал, то массив урезается. Если размер выделенной области больше требуемого, то в лишние ячейках будет записано сообщение "#Н/Д" – недействительные данные. Напомним, что динамический массив объявляется дважды: первый раз в разделе общие без указания размерности оператором Dim, а второй раз в процедуре оператором ReDim с указанием размерности.

Option Base 1 ' нумерация элементов массива
будет начинаться с нуля

Dim ay() As Single ' объявление массива

Public Function Tab11(a, b, shag)

Dim i As Single, x As Single

Dim n As Integer

n = Int((b - a) / shag) + 1

ReDim ay(n, 2) As Single

x = a

For i = 1 To n

ay(i, 1) = x: ay(i, 2) = Round(Sin(x), 2)

x = x + shag

Next i

Tab11 = ay

End Function

	A	B
1	Хнач=	1,5
2	Хкон=	6
3	Шаг=	0,5
4		
5	1,5	1
6	2	0,91
7	2,5	0,6
8	3	0,14
9	3,5	-0,35
10	4	-0,76
11	4,5	-0,98

Рис.3.9. Табулирование функции одной переменной

Табулирование функции одной переменной. Вариант 2

Функция Tab12 (Рис. 3.10) возвращает массив. Входными данными также является массив.

Массив А	Результат
1	2
2	4
3	6
4	8
5	10
6	12
7	14

Внесите данные исходного массива в столбец А. Выделите область, куда будут помещены результаты. Вызовите функцию Tab12. Укажите область данных и нажмите комбинацию клавиш [Ctrl+Shift+Enter]. Особенность данной функции заключается в том, что переменная, объявленная в заголовке функции как переменная типа Variant, может принимать данные как массив. Возвращаются данные с помощью массива Z, который должен быть объявлен как двумерный массив.

Табулирование функции одной переменной. Вариант 3

Данная функция отличается от функции Tab11 тем, что исходные данные вводятся из заранее известных ячеек: А44 – начальное значение отрезка табулирования, А45 – конечное значение отрезка табулирования, А46 – шаг табулирования.

Public Function Tab13() As Variant

Dim a As Single, b As Single,
s As Single

Dim i As Integer, x As Single

a = Range("A44")

b = Range("A45")

s = Range("A46")

n = Int((b - a) / s) + 1

ReDim ay(n, 2) As Single

x = a

For i = 1 To n

ay(i, 1) = Round(x, 2): ay(i, 2) =

Round(Sin(x), 2)

x = x + s

Next i

Tab13 = ay

End Function

Public Function Tab12(xm As Variant) As Variant

Dim i As Integer

n = xm.Rows.Count

' определение числа строк в массиве

ReDim z(n, 1) As Variant

For i = 1 To n

z(i, 1) = Cos(xm(i))

Next i

Tab12 = z

End Function

Рис. 3.10. Табулирование функции одной переменной

Табулирование функции двух переменных

Принцип работы функции двух переменных подобен принципу работы функции Tab12, только вводить требуется два массива аргументов. Расположение массивов исходных данных не имеет значения (рис. 3.11).

X=	Y=	Выходной массив Z(x,y)			
1	0	1,00	2,00	3,00	4,00
1,2	0,5	1,20	2,20	3,20	4,20
1,4	1	1,40	2,40	3,40	4,40
1,6	1,5	1,60	2,60	3,60	4,60

Рис.3.11. Табулирование функции двух переменных

Функция пользователя для табулирования функции двух переменных

Public Function Tab22(xm As Variant, ym As Variant) As Variant

Dim i As Integer, j As Integer, n As Integer, m As Integer

n = xm.Rows.Count

m = ym.Rows.Count

ReDim z(n, m) As Variant

For i = 1 To n

For j = 1 To m

z(i, j) = xm(i) + 2 * ym(j)

Next j

Next i

Tab22 = z

End Function

3.7.3. Разработка окна диалога

Порядок разработки окна диалога

Большинство элементов управления в редакторе VBA аналогичны по своему назначению соответствующим элементам в Visual Basic 6.0. Поэтому комментарии будут даны только к новым объектам.

1. Введите команду **Сервис, Макрос, Редактор Visual Basic** - открывается окно редактора VBA.

2. Введите команду **Insert, UserForm**.

3. Откройте панель инструментов ToolBox (если она отсутствует на экране) командой **View, ToolBox** или щелчком мыши по соответствующей кнопке на панели инструментов Стандартная.

4. Установить на форму четыре метки Label (Надпись) для внесения подписей к окнам ввода (рис. 3.12): Хнач, Хкон, Шаг и Выходной массив.

5. Установите на форму три объекта: TextBox (Начало, Конец, Шаг) для ввода данных и один объект RefEdit (RefEdit1). Объект RefEdit позволяет указать область данных путем выделения мышью на рабочем листе, или вводом диапазона с клавиатуры, так же как и в стандартных функциях Excel.

6. Установите на форму две командные кнопки CommandButton (Вычисления - cmdCalc и Выход - cmdExit).

7. Объявите переменные, общие для формы в разделе General.

8. Объявите функцию пользователя для вычисления заданной функции.

9. Запишите текст программы в обработчике событий Click командных кнопок.

10. Сохраните рабочую книгу командой **File, Save <имя рабочей книги>**.

11. Перейдите на рабочий лист командой **File, Close and Return to Microsoft Excel** или с помощью кнопок на панели задач (предпочтительнее).

12. Установите на рабочий лист кнопку:

- откройте панель инструментов Элементы управления;
- активизируйте кнопку "Режим конструктора" (в верхнем левом углу панели элементов управления). Имя кнопки меняется на "Выход из режима конструктора";
- установите на рабочий лист кнопку (имя кнопки по умолчанию CommandButton1);
- вызовите окно свойств кнопки через контекстное меню или с панели элементов управления и измените надпись на кнопке, например, "Окно диалога";
- вызовите двойным щелчком по кнопке окно программы (Code) и запишите в обработчик события Click кнопки команду для открытия окна диалога:

```
UserForm1.Show;
```

- вернитесь на рабочий лист и деактивируйте кнопку "Выход из режима конструктора".

13. Сохраните рабочую книгу и проверьте работу программы.

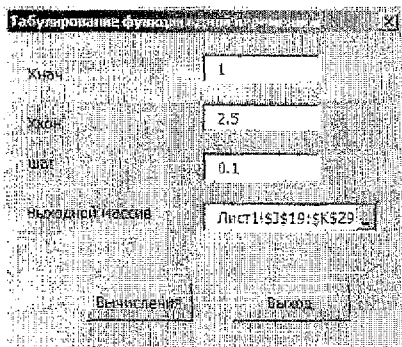


Рис. 3.12. Окно диалога для табулирования функции одной переменной

Текст программы

Option Explicit ' раздел General

Option Base 1

Dim D() As Variant

Dim a As Single, b As Single, h As Single, n As Integer, m As Integer

Public Function TabOD(a, b, h)

'Функция пользователя для табулирования функции одной переменной с выводом

данных в массив

Dim i As Single, x As Single

ReDim ay(n, m) As Single

n=Int((b-a)/h)+1

x = a

For i = 1 To n

ay(i, 1) = x: ay(i, 2) = Round(Sin(x), 2)

x = x + h

Next i

TabOD = ay

End Function

Private Sub cmdExit_Click()

' Обработчик события кнопки Выход

Unload Me

End Sub

Private Sub cmdCalc_Click() ' обработчик события кнопки Вычисления

a = Val(Начало.Text) ' начало отрезка табулирования

b = Val(Конец.Text) ' конец отрезка табулирования

h = Val(Шаг.Text) ' шаг табулирования

' D - выходной массив

m = Range(RefEdit1.Value).Columns.Count ' число столбцов

n = Range(RefEdit1.Value).Rows.Count ' число строк

ReDim D(n, m) As Variant ' переобъявление массива

Range(RefEdit1.Value).Select

Selection = TabOD(a, b, h)

End Sub

Пример оформления курсовой работы:

П1.1. Краткие теоретические сведения

Под движением механизма понимается воспроизведение положения механизма в последовательные моменты времени. При моделировании движения механизма рассматривают следующие вопросы:

1. Воспроизведение положения механизма в исходном состоянии.
2. Воспроизведение положения механизма в нескольких промежуточных положениях.
3. Определение зон движения звеньев механизма.
4. Определение области, занимаемой механизмом в процессе движения.
5. Построение траектории движения заданной точки.

Основными элементами механизма являются (рис.П1.1): кривошип (1), шатун (2, 3), ролик (4) с радиусом (5), ползун (6).

П1.2. Исходные данные

К исходным данным относятся координаты опор, ползуна, ролика, длины звеньев механизма. Исходные данные удобно задавать в пикселях, тем более что координаты опор на экране иначе задать невозможно. Масштаб преобразования для линейных величин примем равным 100:1, то есть 100 пикселей равны 1 м.

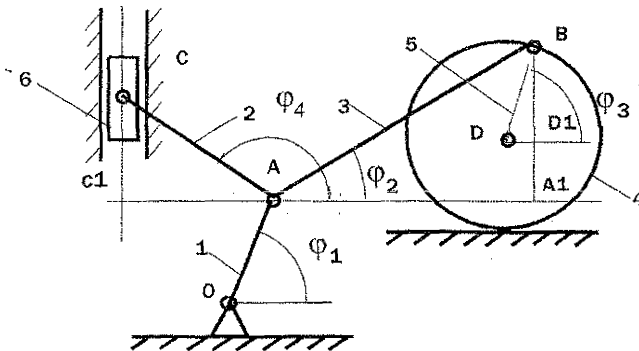


Рис. П1.1. Механизм

Координаты опоры O: X=100, Y=50, угол $\phi_1=45$ градусов, длина звена OA=0,40 м. Длина звена AB=1,5 м. Координата X точки C равна 100, длина звена CA=0,8 м. Высота расположения опоры E для ролика равна 0,60 м. Радиус ролика DB равен 0,4 м.

Исходные данные сведем для удобства использования в таблицу (табл. П1.1). При записи данных в таблицу проведем преобразование линейных размеров звеньев механизма с учетом принятого масштаба.

Таблица П1.1

Исходные данные

	X	Y	F	L
O	100	50	45	40
A	-	-	-	150
B	-	-	-	-

	X	Y	F	L
C	100	-	-	80
D	-	60+40	-	40
E	-	60	-	-

В табл. П1.1 номера строк обозначены номерами точек звеньев механизма О, А, В, С, D, Е, номера столбцов обозначают координаты точек X, Y, угол положения звена механизма - F и длину звена механизма - L. Значения неизвестных параметров обозначены в таблице знаком “?”.

Состояние механизма в статике определяется положением кривошипа (угол φ_1).

Расчету в данном механизме подлежат значения координат точек А, В, координаты Y_c для точки С и X_d для точки D. В качестве промежуточных результатов необходимо иметь значения углов φ_2 и φ_3 .

П1.3. Разработка математической модели

Координаты точки А определяются по длине кривошипа ОА и значению угла φ_1 :

$$X_a = X_0 + L_1 \cos \varphi_1 \quad (\text{П1.1})$$

$$Y_a = X_0 + L_1 \sin \varphi_1. \quad (\text{П1.2})$$

Координата Y точки С определяется из прямоугольного треугольника CAC_1 . Длина СА известна, а $C_1A = X_a - X_c$, тогда

$$Y_c = Y_a + \sqrt{C_A^2 - C_1A^2} \quad (\text{П1.3})$$

В данном механизме ролик имеет две степени свободы. Он может вращаться вокруг своей оси и перемещаться в горизонтальном направлении. В исходных данных высота расположения ролика и радиус ролика заданы, а следовательно, и вертикальная координата центра ролика Y_d известна. Горизонтальное положение ролика X_d - неизвестно. Наибольшие трудности возникают при определении значения угла φ_2 .

Для определения значения угла φ_2 необходимо составить и решить систему из пяти уравнений:

$$x_b = x_a + l_2 \cos \varphi_2 \quad (\text{П1.4})$$

$$x_b = x_d + l_3 \cos \varphi_3 \quad (\text{П1.5})$$

$$y_b = y_a + l_2 \sin \varphi_2 \quad (\text{П1.6})$$

$$y_a = y_d + l_3 \sin \varphi_3 \quad (\text{П1.7})$$

$$(x_b - x_d)^2 + (y_b - y_d)^2 = l_3^2 \quad (\text{П1.8})$$

Здесь l_2 - длина шатуна АВ, l_3 - радиус ролика. Первые четыре уравнения получают из рассмотрения прямоугольников ABB_1 и DBD_1 . Пятое уравнение составляется из следующих соображений: координаты точки В лежат на поверхности окружности, поэтому можно записать уравнение окружности с центром в точке D и радиусом, равным радиусу ролика l_3 .

Из уравнений П1.4 и П1.5 находим значение X_d :

$$x_d = x_a + l_2 \cos \varphi_2 - l_3 \cos \varphi_3 \quad (\text{П1.9})$$

Из уравнений П1.6 и П1.7 находим значение угла φ_3 :

$$\varphi_3 = \arcsin((y_a + l_2 \sin \varphi_2 - y_d) / l_3) \quad (\text{П1.10})$$

Заменив в выражении (П1.9) угол φ_3 на его значение из (5.1.10) и подставив значения X_b (П1.4), X_d и Y_b (П1.9) в (П1.6), получим уравнение:

$$((x_a + l_2 \cos \varphi_2 - (x_a + l_2 \cos \varphi_2 - l_3 \cos(\arcsin((y_a + l_2 \sin \varphi_2 - y_d) / l_3)))^2 + (y_a + l_2 \sin \varphi_2 - y_d)^2) - l_3^2 = 0. \quad (\text{П1.11})$$

В выражении (П1.11) один неизвестный параметр - угол φ_2 . Найти значение угла φ_2 из данного уравнения можно с помощью одного из методов поиска значения корня, например, методом итераций.

П1.4. Схема алгоритма программы

Список задач, выполняемых программой, полностью вытекает из задания. В соответствии с этим в программе необходимо предусмотреть следующие функциональные блоки:

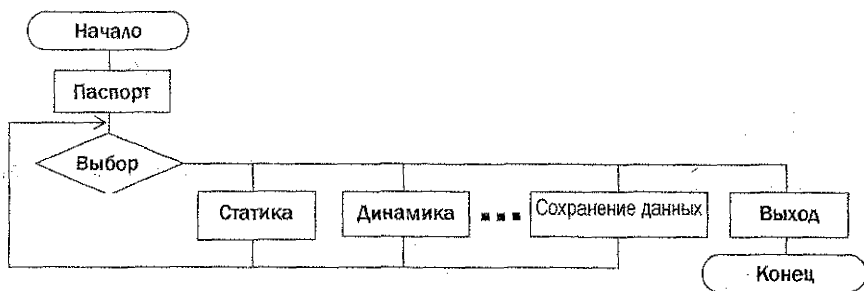


Рис. П1.2. Обобщенный алгоритм программы

- Паспорт программы;
- Меню;
- Блок моделирования механизма в статике;
- Блок моделирования механизма в динамике;
- Блок моделирования движения заданной точки или произвольной точки механизма;
- Блок определения зон движения звеньев механизма и механизма в целом.

Все блоки независимы друг от друга. Вызов блоков программы осуществляется из меню пользователя, и после окончания работы блока управление передается меню программы. Указанная особенность программы полностью согласуется с принципами структурного программирования, и принципом событийно-управляемого программирования, характерного для программирования в среде Visual Basic. Схема алгоритма приведена на рис. П1.2.

П1.5. Разработка проекта программы

Программа должна состоять из нескольких форм в соответствии с определенными задачами. Совместную работу форм удобно организовать с помощью MDI-формы. В эту форму можно поместить меню пользователя и фоновый рисунок.

Так как в задании не оговорены требования к интерфейсу пользователя при моделировании движения механизма, то состав элементов управления на форме разработчик определяет в соответствии со своим представлением об удобстве управления демонстрацией свойств механизма. В простейшем случае можно ограничиться двумя кнопками: старт и выход. В учебных целях при разработке интерфейса целесообразно использовать как можно больше различных элементов управления.

Описание переменных

Опишем глобальные переменные, используемые в программе (табл. П1.2)

Аналогично описываются переменные процедур и функций.

Таблица П1.2

Глобальные переменные

Наимен. програм. <i>Mechanizm</i>	Тема проекта: <i>Моделирование движения механизма</i>		Лист 1
			Листов __
Обозначение переменной	Имя переменной	Тип	Примечание
	Obj	Object	Глобальная переменная типа Object, заменяет имя объекта при вызове процедур, осуществляющих графические построения. В каждой форме, в которой рисуется механизм, введен оператор: Set Obj=<имя формы>

Продолжение Таблица П1.2

X ₀ , X _a , X _b , X _c , X _d	X(5)	Single	Координаты X узлов механизма
Y ₀ , Y _a , Y _b , Y _c , Y _d	Y(5)	Single	Координаты Y узлов механизма
φ ₁ , φ ₂ , φ ₃ , φ ₄	f(5)	Single	Значения углов
l ₁ , l ₂ , l ₃ , l ₄ , l ₅	L(5)	Single	Длины звеньев механизма
	f1	Single	Текущий угол поворота
	pi	Double	число
	pi2	Double	число 2π
	dx	Single	шаг
	Ff	Single	Вспомогательная переменная, хранит вычисленное значение угла φ ₂
	i1	Double	Максимальный угол поворота кривошипа i1=2*π
	f1	Byte	Флаг
	f12	Byte	Флаг
	x0	Integer	Вспомогательная переменная, для хранения целых чисел, используется при построении опор.
	y0	Integer	то же
	x1	Integer	то же
	y1	Integer	то же
	n	Integer	то же

Описание элементов управления

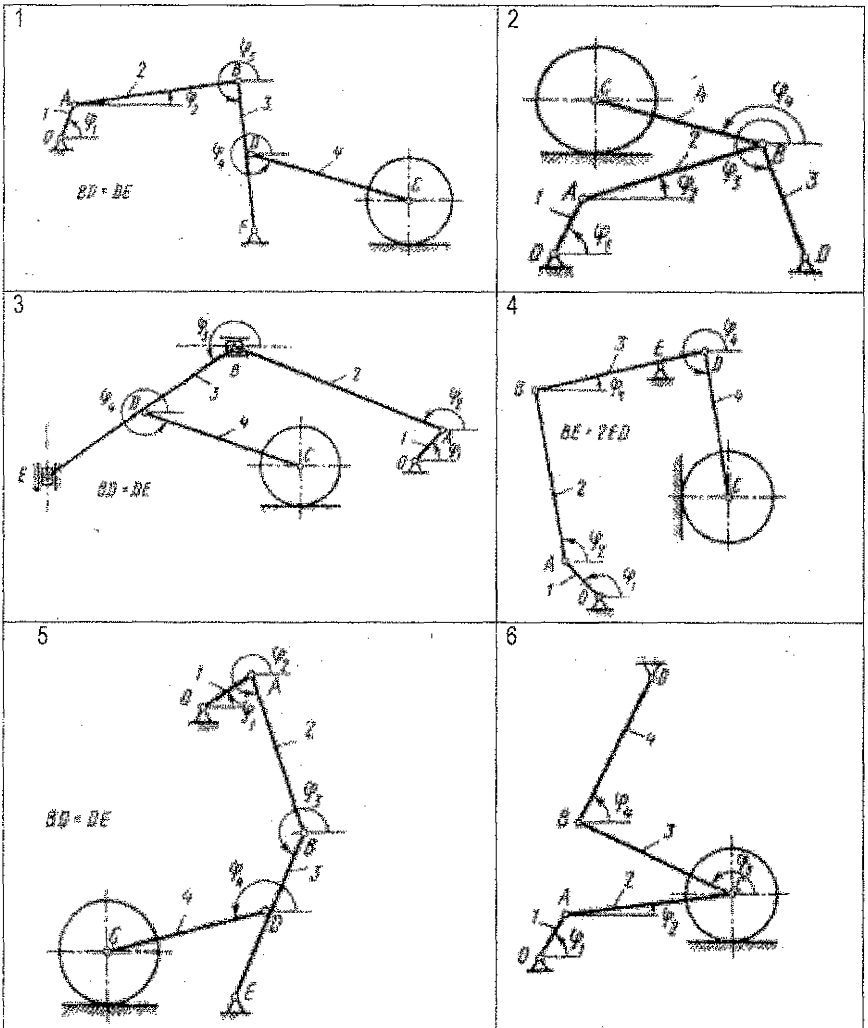
Опишем основные свойства элементов управления формы Статика (табл. П1.3). Аналогично описываются элементы управления и других форм.

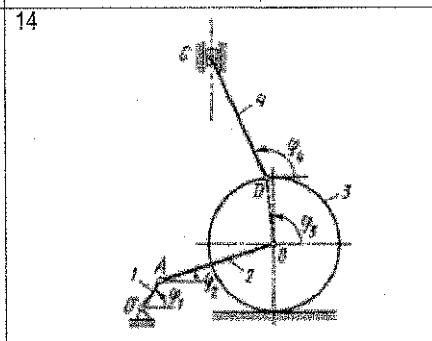
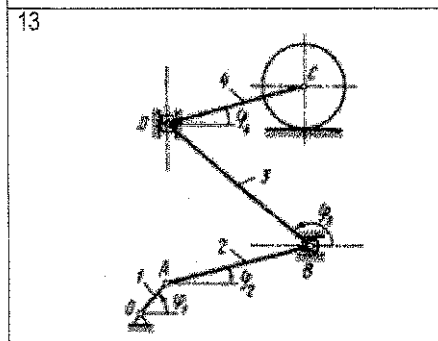
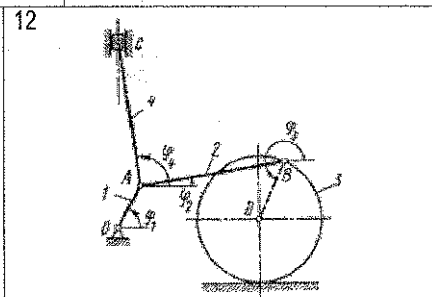
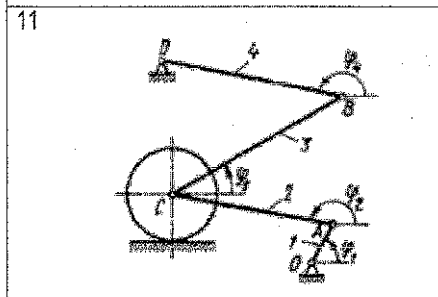
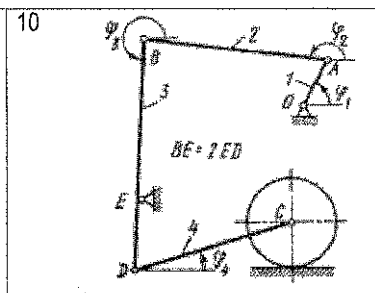
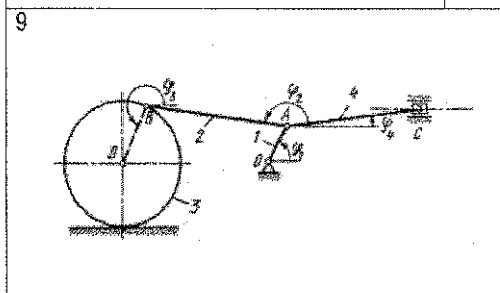
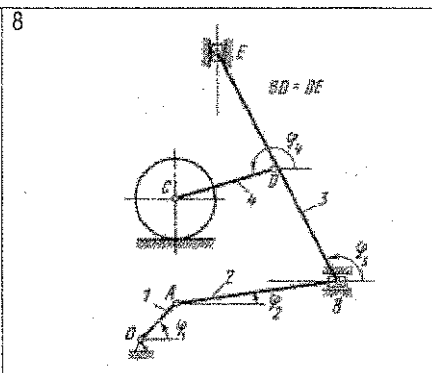
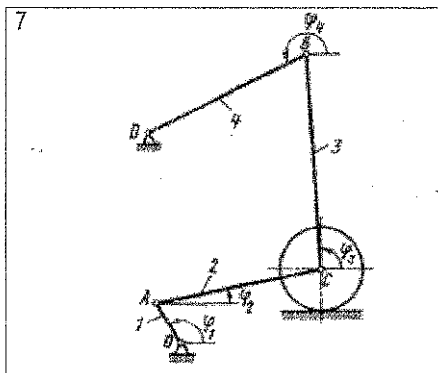
Таблица П1.3.

Свойства элементов управления формы Статика

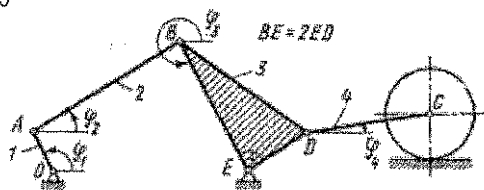
Наимен. програм. <i>Mechanism</i>	Тема проекта: <i>Моделирование движения механизма</i>		Лист <u>2</u>
			Листов <u> </u>
Объект	Свойство	Значение	
Форма	Name	FormStatica	
	Caption	Статика	
	AutoRedraw	True	
	MDIChild	2 * Maximize	
Командная кнопка 1	Name	Command1Statica	
	Caption	Старт	
	Height	375	
	Width	1335	
Командная кнопка 2	Name	Command2Statica	
	Caption	Выход	
	Height	375	
	Width	1335	
Сетка	MSFlexGrid	MSFlexGridStatic	
	Rows	6	
	Cols	3	
Надпись	Name	LabelStatica	
	Caption	"Укажите угол поворота в градусах"	
	Alignment	2 * Center	
Текстовое поле	Name	TextStatica	
	Text	0	

Рисунки для расчета движения механизма

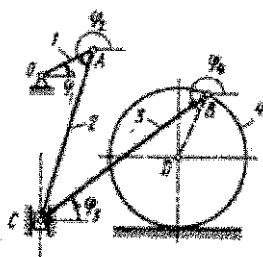




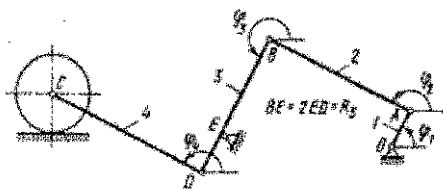
15



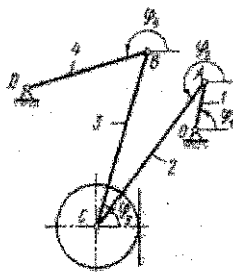
16



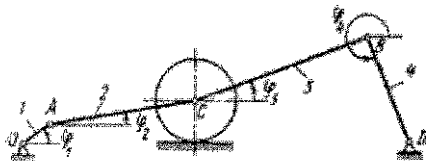
17



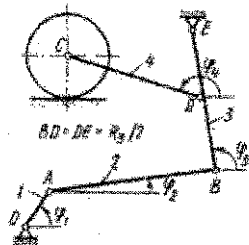
18



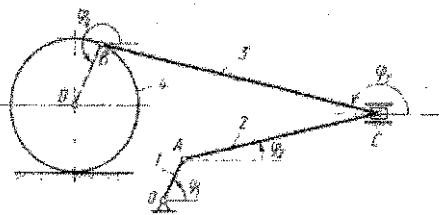
19



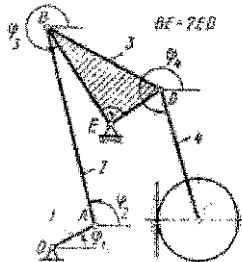
20



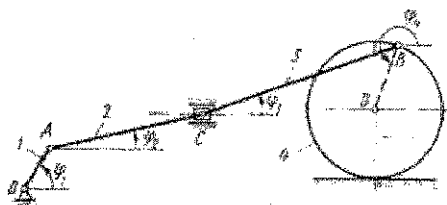
21



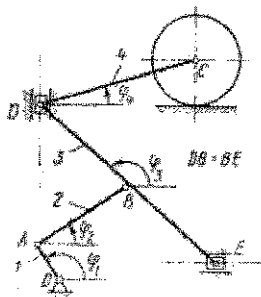
22



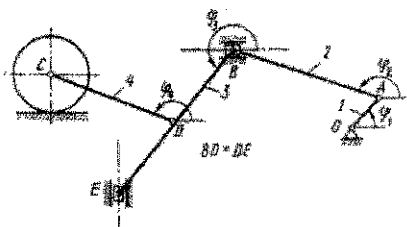
23



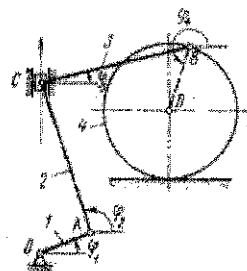
24



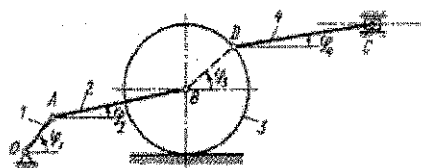
25



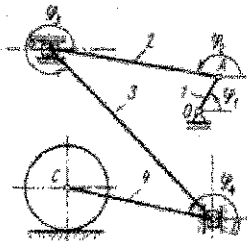
26



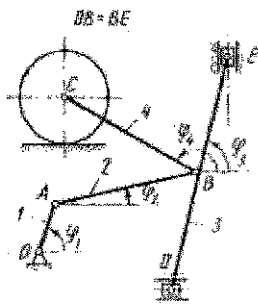
27



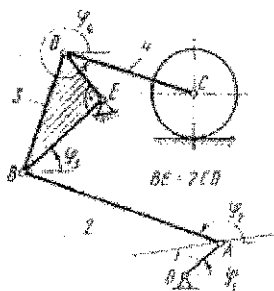
28



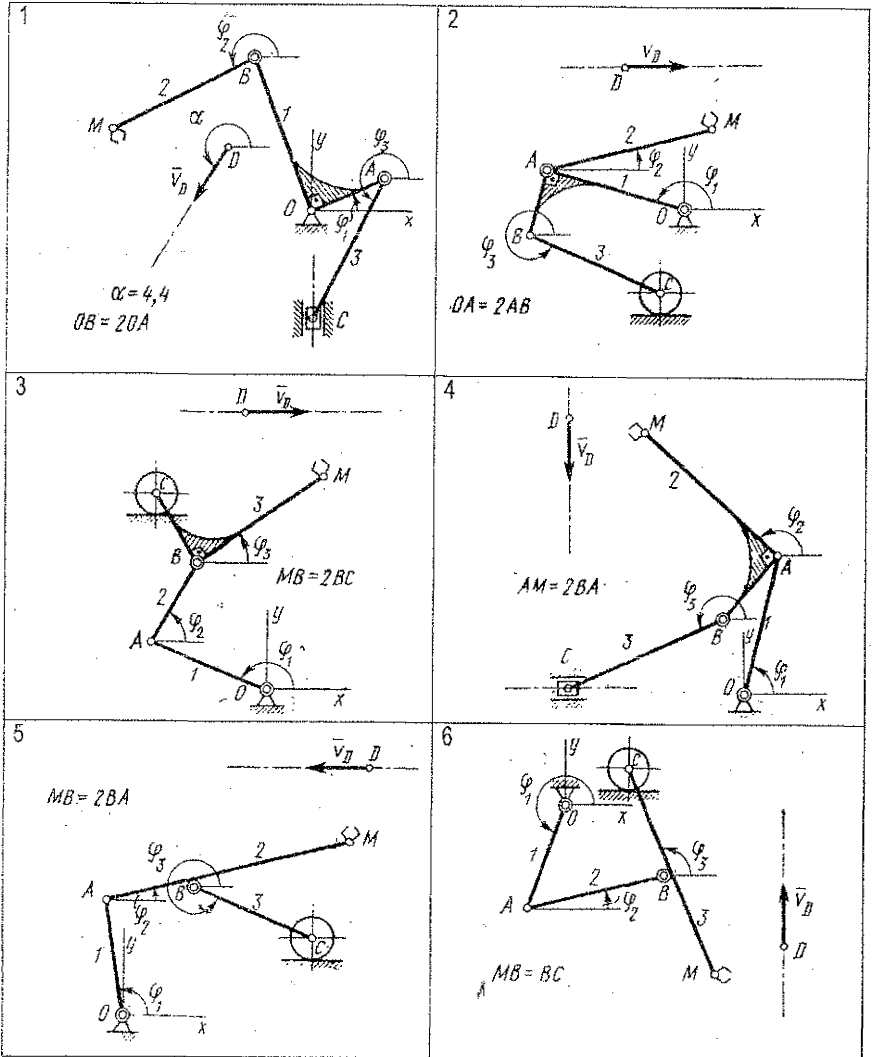
29

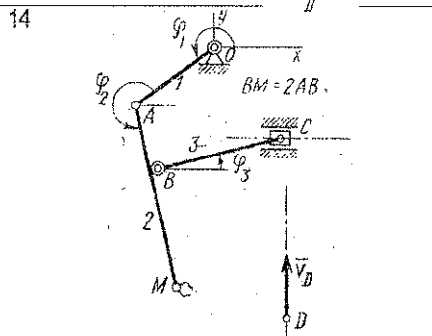
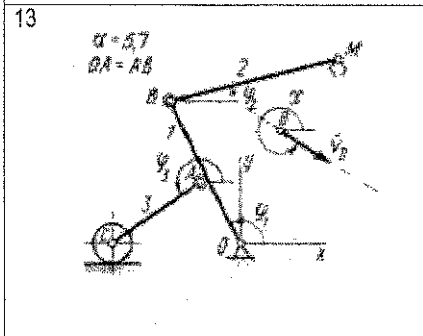
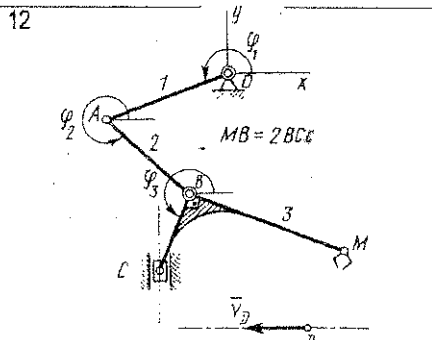
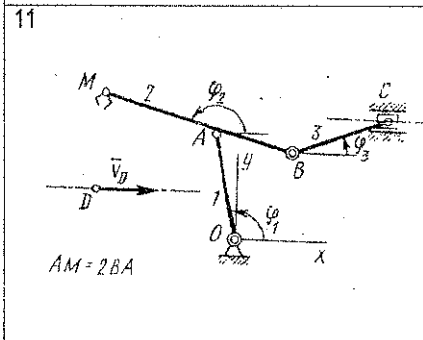
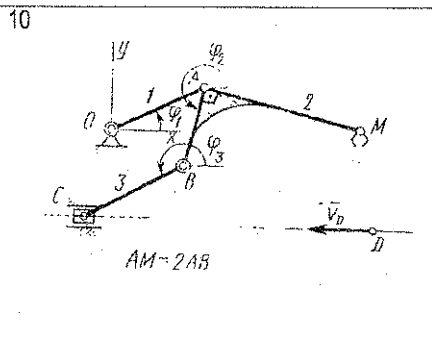
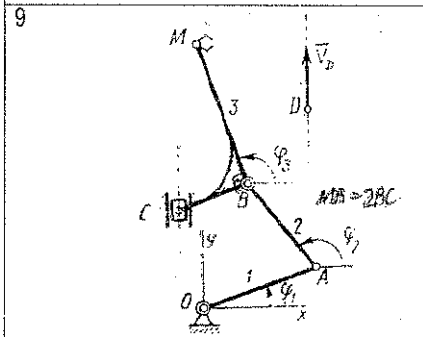
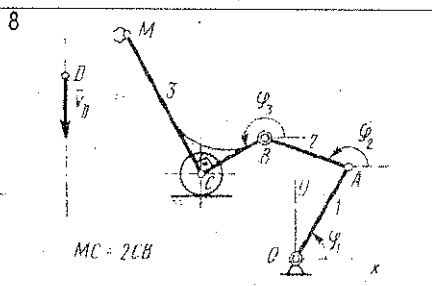
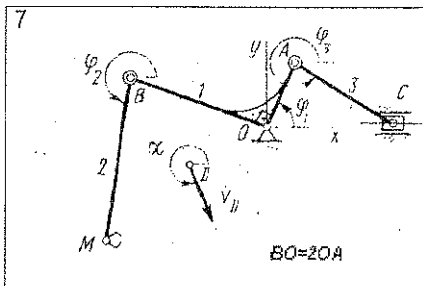


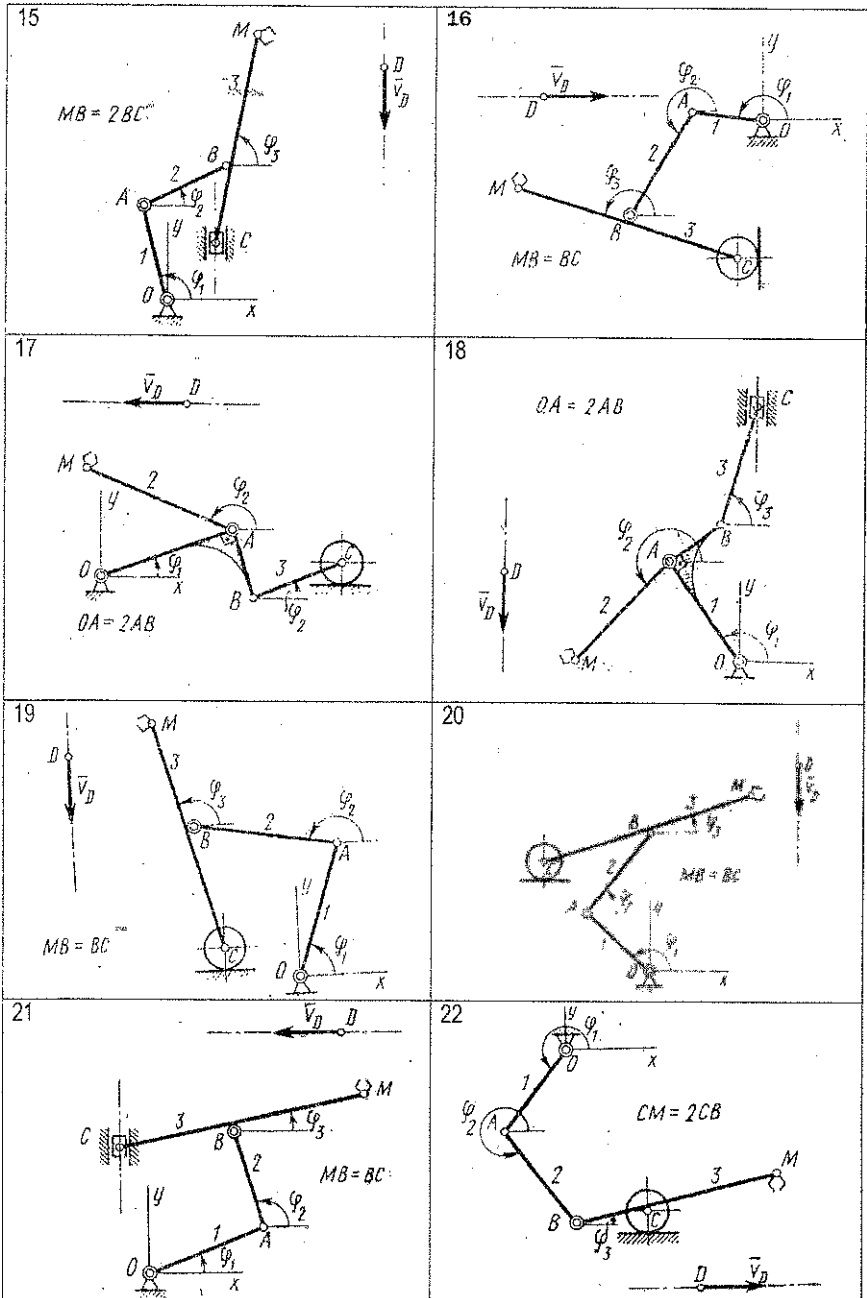
30



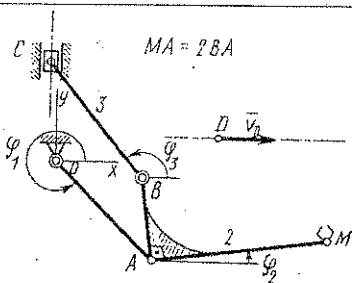
Рисунки для расчета манипулятора



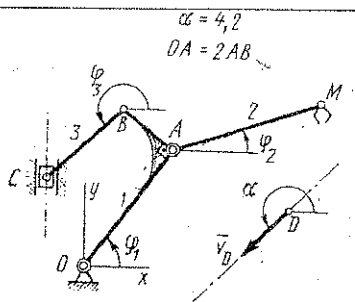




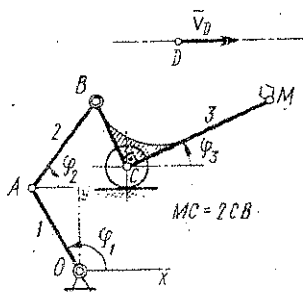
23



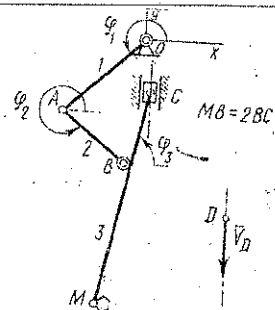
24



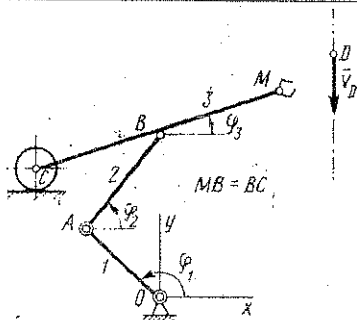
25



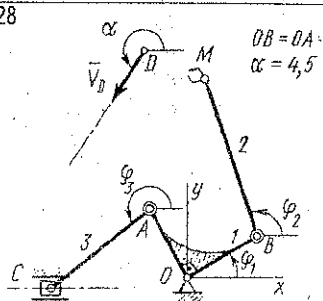
26



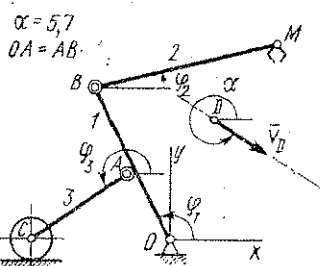
27



28



29



30

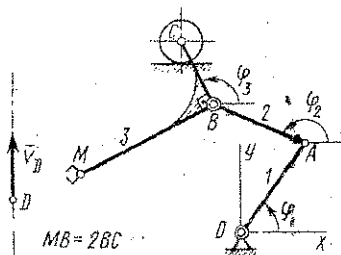
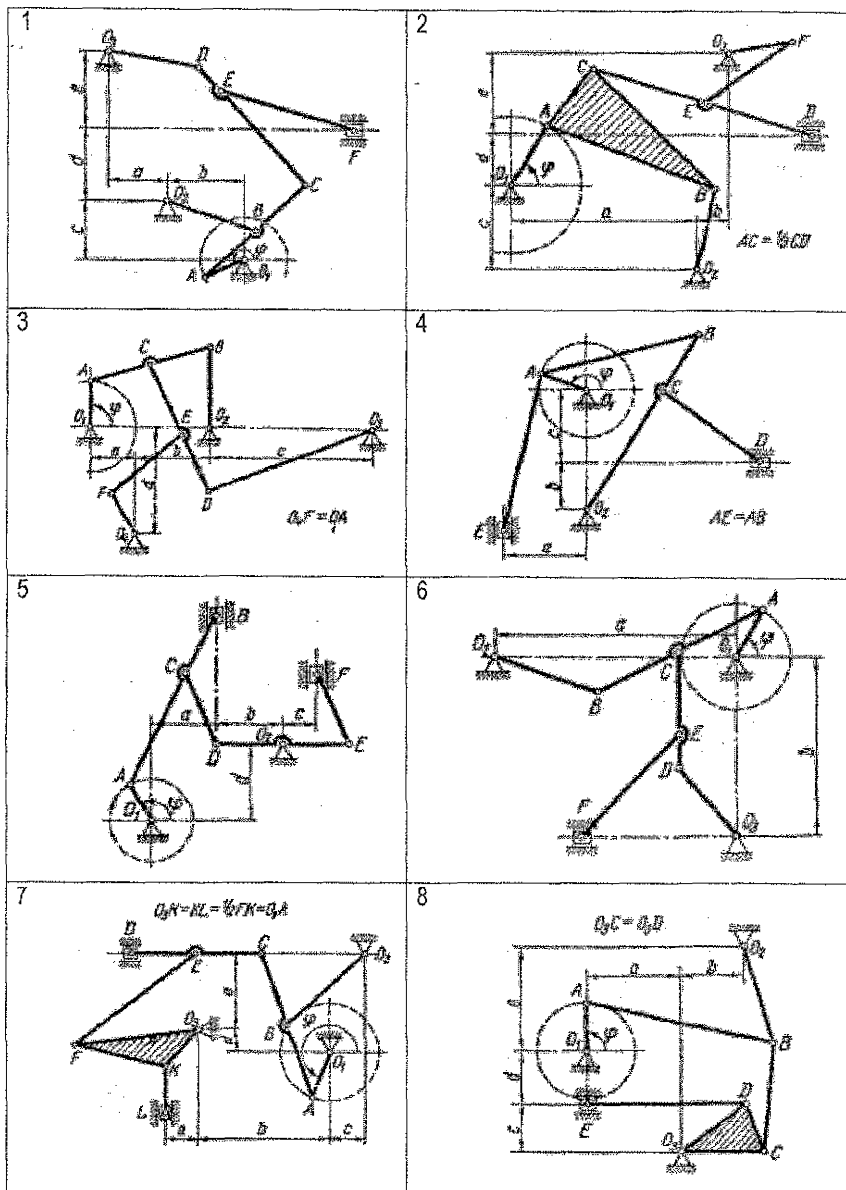
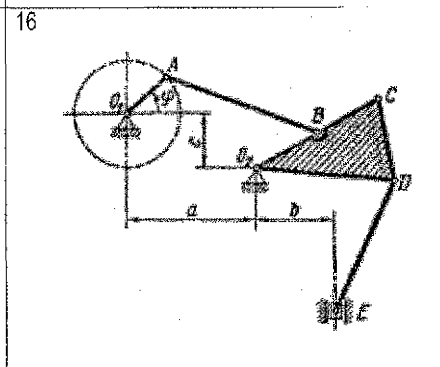
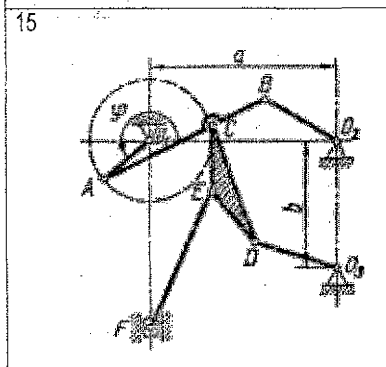
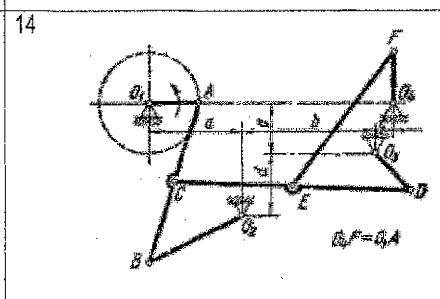
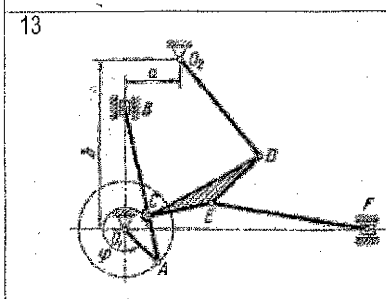
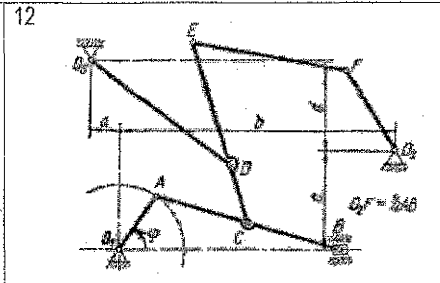
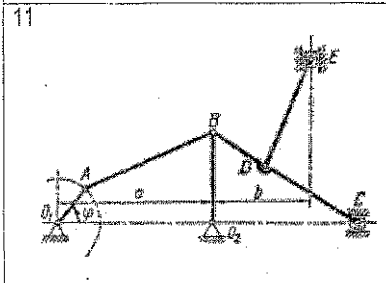
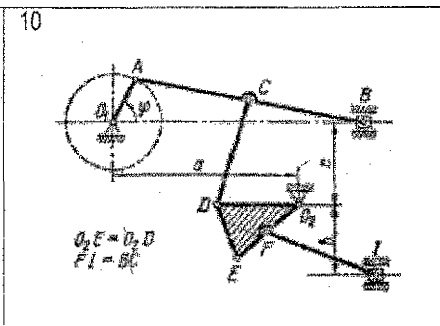
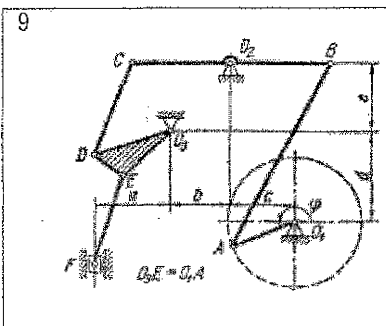


Таблица исходных данных для расчета манипулятора

	$r1\tau$	$r2\tau$	$r3\tau$	$\varphi1\tau$	$\varphi2\tau$	$\varphi3\tau$	$v\Delta\tau$	$x\Delta(0)$	$yD(0)$	δ
1	0.82	0.68	0.46	2.9	1.1	0.5	0.508	-0.15	1.85	0.011
2	0.81	0.47	0.91	1.3	2.2	3.6	0.308	-0.94	1.71	0.012
3	0.43	0.91	0.84	0.3	3.8	4.2	0.512	-0.42	1.25	0.013
4	0.42	0.97	0.88	2.8	0.2	5.7	0.462	-0.21	1.22	0.014
5	0.78	0.45	0.91	1.7	0.1	5.8	0.385	1.35	1.51	0.015
6	0.71	0.89	0.76	4.6	0.1	1.6	0.312	1.33	-1.20	0.016
7	0.46	0.97	0.74	1.3	4.3	5.6	0.421	-0.61	-0.24	0.017
8	0.81	0.72	0.48	1.1	3.0	3.3	0.472	-1.38	1.61	0.018
9	0.76	0.79	0.45	0.3	2.4	1.8	0.465	0.54	1.02	0.019
10	0.72	0.49	0.78	0.5	4.2	3.6	0.375	1.61	-0.55	0.020
11	0.83	0.57	0.49	0.5	1.6	3.0	0.525	-0.92	1.78	0.021
12	0.68	0.46	0.83	3.9	4.9	0.3	0.310	0.46	-2.04	0.022
13	0.78	0.85	0.49	2.1	1.0	0.1	0.460	0.51	1.65	0.023
14	0.48	0.97	0.73	0.3	1.8	3.7	0.402	-0.26	1.30	0.024
15	0.42	0.97	0.78	0.3	2.9	0.4	0.455	0.45	1.12	0.025
16	0.51	0.82	0.79	3.2	4.1	3.0	0.288	-1.57	0.13	0.026
17	0.41	0.83	0.98	2.0	4.3	1.4	0.451	-1.18	0.56	0.027
18	0.82	0.45	0.78	1.6	2.9	0.4	0.312	-0.99	0.52	0.028
19	0.92	0.98	0.81	1.5	2.7	1.7	0.294	-1.43	1.95	0.029
20	0.79	0.68	0.48	4.1	5.8	1.1	0.306	0.41	-1.43	0.030
21	0.76	0.42	0.85	5.2	0.4	2.3	0.380	0.84	0.26	0.031
22	0.75	0.78	0.47	1.1	2.8	2.0	0.515	-1.66	0.42	0.032
23	0.71	0.49	0.82	4.9	0.1	1.9	0.385	0.62	0.12	0.033
24	0.75	0.65	0.78	0.3	1.9	0.1	0.398	1.11	1.32	0.034
25	0.68	0.79	0.82	2.3	0.7	0.5	0.392	1.40	1.67	0.035
26	0.81	0.72	0.49	3.7	5.4	4.2	0.371	0.43	-1.98	0.036
27	0.78	0.65	0.48	1.6	0.1	1.5	0.275	1.31	1.62	0.037
28	0.45	0.97	0.78	0.9	0.5	3.9	0.290	1.22	0.78	0.038
29	0.49	0.98	0.77	2.1	0.4	3.7	0.305	0.21	0.72	0.039
30	0.72	0.75	0.49	3.9	5.4	0.3	0.340	1.14	-1.25	0.040

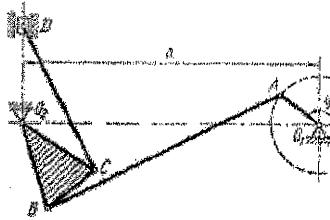
Рисунки для моделирования движения механизма



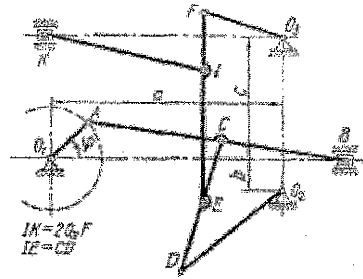


17

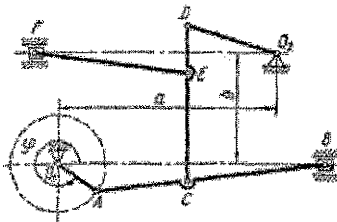
$$O_2C = O_2B$$



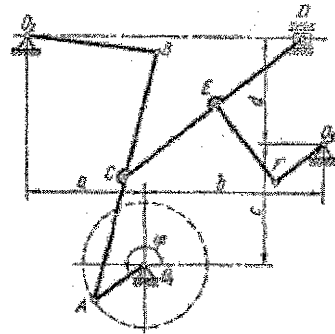
18



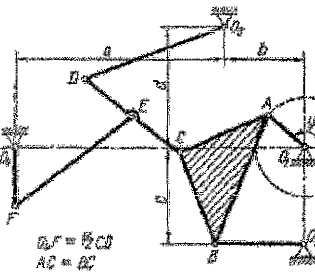
19



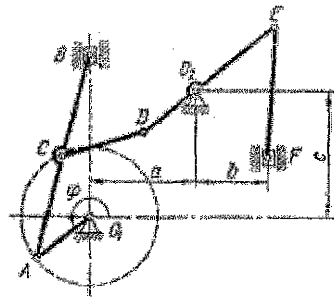
20



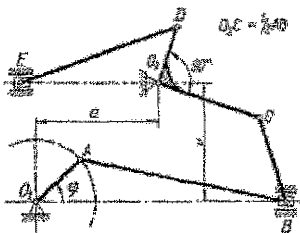
21



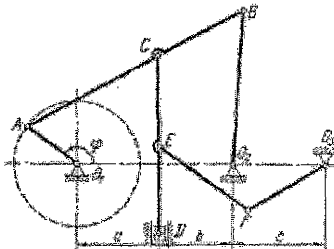
22



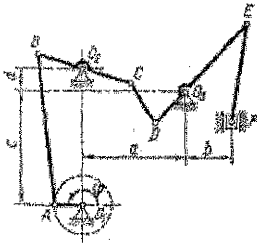
23



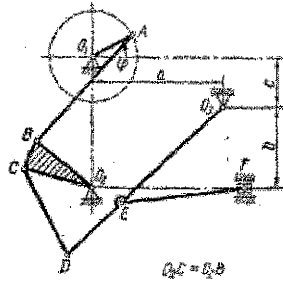
24



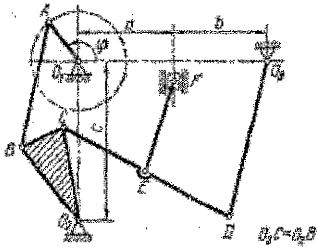
25



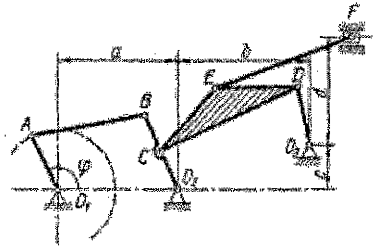
26



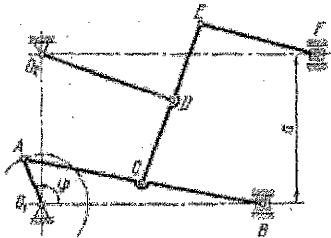
27



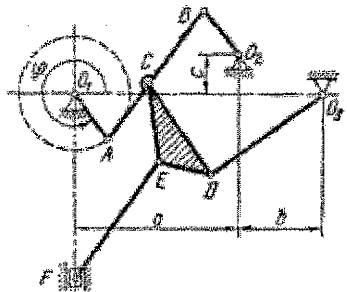
28



29



30



Список рекомендуемой литературы

1. Стандарт университета. Оформление материалов курсовых и дипломных проектов (работ), отчетов по практике. Общие требования и правила оформления. СТ БГТУ01 – 2008. – Брест: БрГТУ, 2008. – 47 с.
2. Быков В. Л. Основы информатики: конспект лекций. – Брест: БГТУ, 2003. – 253 с.
3. Быков В. Л., Ашаев Ю. П. Основы информатики: пособие – Брест, БрГТУ, 2006. – 450.
4. Сборник индивидуальных заданий по высшей математике, ч.1. – Мн.: Высшая школа, 1990. – 270 с.
5. Воробьева Г. Н., Данилова А. Н. Практикум по вычислительной математике. – М.: Высшая школа, 1990. – 208 с.: ил.
6. Турчак Л. И. Основы численных методов. – М.: Наука, 1987. – 320 с.
7. Дьяконов В. П. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ. – М.: Наука, 1989. – 240 с.
8. Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. – М.: Наука, 1972. – 368 с.: ил.

Учебное издание

Составитель: Быков Вячеслав Леонидович

Методические рекомендации

по выполнению курсовых работ по дисциплине «Информатика»
для специальностей 36 01 01 «Технология машиностроения», 36 01 03
«Технологическое оборудование машиностроительного производства»

Издание четвертое переработанное, дополненное

Ответственный за выпуск: Быков В.Л.

Редактор: Строкач Т.В.

Компьютерная верстка: Кармаш Е.П.

Корректор: Никитчик Е.В.

Подписано к печати 09.03.2009 г. Формат 60x84 1/16. Усл. п. л. 2,79. Уч. изд. 3,0.
Бумага «Снегурочка». Тираж 100 экз. Заказ №299. Отпечатано на ризографе уч-
реждения образования «Брестский государственный технический университет».

224017, Брест, ул. Московская, 267