МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Транспортные и технологические машины»

ИНФОРМАТИКА

Методические рекомендации к курсовому проектированию для студентов специальности 1-36 11 01 «Подъемно-транспортные, строительные, дорожные машины и оборудование» очной и заочной форм обучения

Могилев 2020

УДК 004.4
ББК 32.81
И 74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ТТМ «06» марта 2020 г., протокол № 8

Составитель ст. преподаватель В. И. Семчен

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации разработаны на основании рабочей программы по дисциплине «Информатика» для обучающихся специальности 1-36 11 01 «Подъемно-транспортные, строительные, дорожные машины и оборудование» и предназначены для использования при выполнении курсовой работы.

Учебно-методическое издание

ИНФОРМАТИКА

Ответственный за выпуск И. В. Лесковец

Корректор Т. А. Рыжикова

Компьютерная верстка Е. В. Ковалевская

Подписано в печать 01.06.2020. Формат 60×84/16. Бумага офсетная. Гарнитура Таймс. Печать трафаретная. Усл. печ. л. 2,55. Уч.-изд. л. 2,69. Тираж 56 экз. Заказ № 238.

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатной продукций
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, Могилев.

Содержание

Введение

Информатика является дисциплиной, обеспечивающей подготовку по использованию информационных технологий в процессе обучения в вузе и последующей профессиональной деятельности. При изучении дисциплины обучающиеся получают представление о программном обеспечении персональных компьютеров, осваивают методы постановки и решения прикладных задач, закрепляют знания и умения в применении математического моделирования и выборе численных методов, приобретают навыки алгоритмирования, создания и отладки программ на языке высокого уровня, анализируют полученные результаты. Полученные знания в дальнейшем используются при изучении общетехнических, специальных дисциплин и дисциплин специализации.

Цель методических рекомендаций – оказание помощи обучающимся при выполнении курсовой работы по дисциплине «Информатика». В методических рекомендациях изложено содержание курсовой работы и даны рекомендации по выполнению практической части и расчетно-пояснительной записки.

1 Цель и задачи курсовой работы

Одним из важных видов учебной деятельности обучающихся, предусмотренных учебным планом дисциплины «Информатика», является курсовая работа, выполняемая во втором семестре.

Цели курсовой работы:

– систематизация, закрепление и углубление теоретических знаний, полученных в процессе обучения;

– оценка степени усвоения основных положений дисциплины;

– выработка и развитие навыков решения задач на ЭВМ с использованием языка высокого уровня в среде Delphi.

Главным результатом выполнения курсовой работы являются отлаженные и протестированные программные модули, обеспечивающие решение задач, сформулированных в задании, и документация, поясняющая основные этапы создания программы.

2 Организация курсового проектирования

Основанием для выполнения курсовой работы является индивидуальное задание, утвержденное заведующим кафедрой. В задании указывается тема, исходные данные, содержание расчетно-пояснительной записки, календарный график выполнения курсовой работы на весь семестр с указанием сроков выполнения отдельных этапов.

Непосредственное руководство выполнением курсовой работы осуществляет преподаватель, который обязан:

– давать рекомендации по содержанию разделов курсовой работы;

– оценивать качество выполнения курсовой работы в соответствии с предъявляемыми к ней требованиями;

– осуществлять систематический контроль выполнения курсовой работы в соответствии с графиком;

– информировать заведующего кафедрой о случаях несоблюдения студентом графика выполнения курсовой работы.

Руководитель курсовой работы имеет право:

– выбрать удобную для него и обучаемого форму организации взаимодействия, согласовать график подготовки курсовой работы и установить периодичность личных встреч;

– по результатам каждой встречи требовать, чтобы обучаемый подготовил план и согласовал дальнейшие шаги по подготовке курсовой работы;

– требовать исполнения полученных рекомендаций;

– при выставлении оценки принять во внимание соблюдение студентом контрольных сроков графика подготовки курсовой работы.

Подведение итогов курсового проектирования включает следующие этапы:

– сдача курсовой работы на проверку руководителю;

– доработка курсовой работы с учетом замечаний руководителя;

– защита курсовой работы.

Законченная курсовая работа, подписанная обучаемым, представляется руководителю на проверку, и если она удовлетворяет предъявляемым требованиям, руководитель делает пометку на титульном листе о допуске к защите. В случае отрицательного заключения работу необходимо доработать или переработать. Срок доработки проекта устанавливается руководителем с учетом сущности замечаний и объема необходимой доработки.

Защите подлежит полностью выполненная и оформленная курсовая работа. Работа защищается перед комиссией, включая руководителя курсовой работы. Обучающийся представляет материалы по своей работе и делает доклад продолжительностью 5–7 минут, после чего отвечает на вопросы. По результатам защиты выставляется оценка по десятибалльной системе. Для оценки курсовой работы используются следующие критерии:

– соответствие разработанной программы заданию, включая правильность построенного алгоритма, полноту анализа особых случаев, полноту тестирований;

– надежность программы, т. е. защищенность от ошибок данных и ошибок самой программы;

– универсальность программы и ее модулей;

– качество оформления документации;

– полнота ответов на вопросы членов комиссии.

Примерный перечень этапов выполнения курсовой работы и количество баллов за каждый из них представлен в таблице 2.1.

Таблица 1.1 – Начисление баллов при выполнении курсовой работы

Этап выполнения	Минимум	Максимум
1 Теоретические исследования проблемы, анализ исходных данных, постановка задачи	6	10
2 Выбор и разработка математической модели	9	15
3 Разработка алгоритма	9	15
4 Разработка программного кода и его отладка в среде Delphi	9	15
5 Анализ результатов и оформление пояснительной записки	3	5
Итого за выполнение курсовой работы	36	60
Защита курсовой работы	15	40

Максимальные баллы начисляются за выполненные точно в срок разделы. Минимальные баллы начисляются при несвоевременном выполнении разделов (без соблюдения календарных сроков выполнения курсовой работы). Итоговая оценка вычисляется как сумма баллов и переводится в десятибальную шкалу по таблице 2.2.

Таблица 1.2 – Итоговая оценка по курсовой работе

Оценка	10	9	8	7	6	5	4	3	2	1	0
Баллы	100...94	93…87	86…80	79...73	72...66	65...59	58...51	50...34	33...17	16...1	0

Студент, не представивший в установленный срок готовую курсовую работу или не защитивший ее, считается имеющим академическую задолженность.

3 Примерная тематика курсовых работ

При выполнении курсовой работы студент решает задачи следующей тематики:

– решение уравнений и их систем с применением численных методов;

– операции над матрицами и векторами;

– вычисление определенных интегралов с применением численных методов;

– решение задач аналитической и исследовательской геометрии на плоскости.

4 Содержание курсовой работы

Выполнение курсовой работы включает проведение анализа исходных данных и постановку задачи, на основании которых выбирают и обосновывают метод решения и приводят его математическое описание. Далее составляют алгоритм и программу для ЭВМ. Производят тестирование программы и полученные в результате выполнения данные анализируют и при необходимости вносят изменения в математическую модель, алгоритм и текст программы.

Рекомендуемый объем пояснительной записки – 20–25 страниц. Структура пояснительной записки приведена в таблице 4.1.

Таблица 4.1 – Структура пояснительной записки

Наименование структурных элементов пояснительной записки	Объем, стр.
Титульный лист	1
Задание на курсовую работу	1
Содержание	1
Введение	1
1 Наименование раздела 1	
1.1 Постановка задачи	1
1.2 Математическая модель	1–2
1.3 Алгоритм программы	1–2
1.4 Интерфейс программы	1–2
1.5 Текст программы	1–2
1.6 Описание использованных в программе операторов и подпрограмм	1–2
1.7 Тестирование и анализ результатов решения задачи	1
2 Наименование раздела 2	
2.1 Анализ исходных данных и постановка задачи	1
2.2 Выбор метода анализа и разработка математической модели	1–2
2.3 Алгоритм программы	1–2
2.4 Интерфейс программы	1–2
2.5 Текст программы	1–2
2.6 Описание использованных в программе операторов, подпрограмм, методов и свойств	1–2
2.7 Тестирование и анализ результатов решения задачи	1
Заключение	1
Список литературы	1

5 Требования к содержанию разделов пояснительной записки

Раздел **«Введение»** должен быть кратким и четким. Из введения должно быть понятно, чему посвящена работа, какие задачи и с помощью каких методов в ней решают, какие результаты должны быть получены в ходе выполнения курсовой работы.

Во введении отражают следующие основные моменты:

– общая формулировка решаемой проблемы;

– цель и задачи курсовой работы;

– описание структуры курсовой работы (названия разделов и их краткая характеристика).

Цель и задачи курсовой работы определяют направления, по которым автор раскрывает тему курсовой работы. Цель курсовой работы – это конечный результат, достигаемый в ходе выполнения курсовой работы. Обычно цель

курсовой работы созвучна названию ей темы. После формулировки цели формируют задачи курсовой работы. Задачи курсовой работы определяют основные этапы для достижения поставленной цели.

В разделе **«Постановка задачи»** приводят исходные данные и формулируют цель решения задачи, поставленной в задании на курсовое проектирование. Анализируют характер и сущность всех величин, используемых в задаче. При необходимости составляют расчетную схему, указывают принятые допущения, исходные данные, формулируют конечную цель решения задачи.

При формулировке цели решения задачи определяют форму представления результатов, перечисляют параметры и зависимости, которые нужно получить в результате решения задачи.

При наличии схемы механической системы приводят сведения об особенностях ее конструкции, действующих внешних факторах и геометрических параметрах.

В разделе «Математическая модель» производят анализ методов решения поставленной задачи и обосновывают выбор одного из них.

Математическая модель – это формальное описание решение задачи в виде математических зависимостей, связывающих исходные данные задачи с искомыми величинами.

Модель должна правильно описывать основные законы физического процесса. Построение или выбор математической модели требует понимания проблемы и знания соответствующих разделов математики.

Разработка модели решаемой задачи заключается в получении уравнений, описывающих выбранный численный метод для расчетных задач или определяющих положения механической системы в зависимости от изменения начальных условий при решении задач аналитической и исследовательской геометрии на плоскости.

В записке излагают последовательность вывода уравнений модели в общем виде с расшифровкой входящих в них величин, не упоминаемых ранее.

Если на последующих этапах будет обнаружена непригодность выбранного метода, необходимо произвести корректировку выбранного метода решения задачи или принять другой численный метод.

В разделе **«Алгоритм программы»** производят описание алгоритмов решаемой задачи и осуществляют пошаговое описание логики их работы.

На практике наиболее распространены следующие формы представления алгоритмов:

– словесная (записи на естественном языке);

– графическая (изображения из графических символов).

Словесный способ не имеет широкого распространения по следующим причинам:

– такие описания строго не формализуемы;

– страдают многословностью записей;

– допускают неоднозначность толкования отдельных предписаний.

Перед началом разработки алгоритма необходимо четко уяснить задачу, что требуется получить в качестве результата, какие исходные данные необходимы и какие имеются в наличии, какие существуют ограничения на эти данные. Далее следует записать, какие действия нужно предпринять для получения из исходных данных требуемого результата и описать их в виде блок-схемы с помощью условных графических обозначений.

При разработке алгоритмов можно использовать следующие методы:

– метод частных целей, который заключается в сведении решения сложной задачи к рассмотрению последовательности более простых задач;

– метод подъема, суть метода в том, что вначале создается самый простой вариант построения алгоритма, который затем последовательно улучшается, пока не достигнет заданного качества.

Эта два метода следует использовать совместно: вначале сложную задачу разбивают на последовательность более простых подзадач, а затем для составления алгоритмов каждой из подзадач используют метод подъема или опять метод частных целей.

Особое внимание следует обратить на проверку выполнения условий и принятие решения по ее результатам, повторяемость (цикличность) вычислений.

Алгоритмы должны однозначно отображать логическую структуру процесса решения задачи.

Графическое изображение алгоритма в виде блок-схемы представляет собой последовательность блоков, соединенных линиями. Внутри символов словами или с помощью формул указывают выполняемую операцию (ввод исходных данных; вычисление расчетных параметров; условие, изменяющее направление выполнения алгоритма).

Схемы алгоритмов следует выполнять в соответствии с требованиями [1].

Все блоки схемы должны иметь уникальные идентификаторы, необходимые для организации ссылок при описании работы алгоритма.

На практике в основном используют типы блоков, представленные далее.

Терминатор. Используют для обозначения выхода во внешнюю среду и входа из внешней среды (начало или конец схемы). Если символ расположен в начале алгоритма, то внутри него записывают «Начало», а если в конце, то «Конец» (рисунок 5.1).

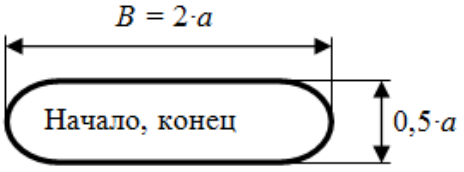

Рисунок 5.1 – Блок «Терминатор»

Данные. Наиболее общий символ ввода-вывода данных, носитель которых не определен. Внутри символа записывают имена вводимых или выводимых данных (рисунок 5.2).

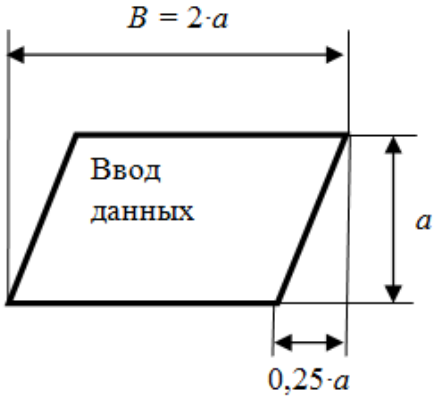

Рисунок 5.2 – Блок «Данные»

Процесс. Указывает на обработку данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации). Внутри символа записывают действия для выполнения операции или группы операций (рисунок 5.3).

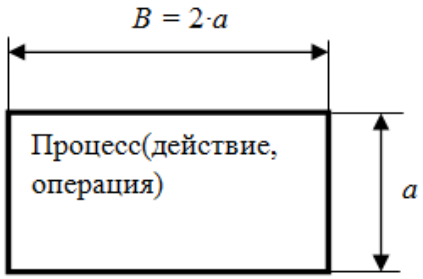

Рисунок 5.3 – Блок «Процесс»

Решение (выбор). Используют для выбора направления выполнения алгоритма в зависимости от некоторого условия, записываемого внутри символа; имеет один вход и ряд альтернативных выходов, один, и только один из которых может быть активизирован после вычисления условий, определенных внутри этого блока (рисунок 5.4).

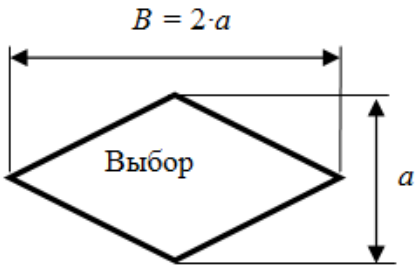

Рисунок 5.4 – Блок «Выбор»

Подготовка. Указывает на использование цикла с параметром. Внутри записывается имя параметра цикла с указанием его начального и конечного значений, а также шаг изменения параметра цикла, если он не равен единице (рисунок 5.5).

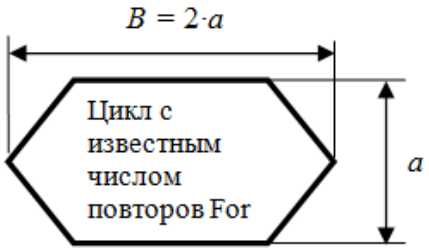

Рисунок 5.5 – Блок «Подготовка»

Предопределенный процесс. Отображает одну или несколько операций, которые определены в подпрограмме, модуле. Внутри символа записывают имя подпрограммы, модуля (рисунок 5.6).

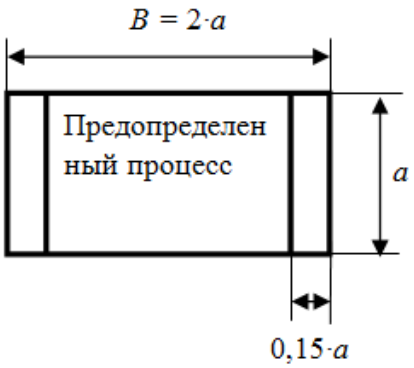

Рисунок 5.6 – Блок «Преопределенный процесс»

Линии потока указывают связи между символами. Для изображения линий потока существуют следующие правила:

– линии потока должны быть строго вертикальными или горизонтальными;

– направление линии потока сверху вниз и слева направо принимается за основное и стрелками не обозначается, в остальных случаях направление линии потока обозначается стрелками;

– изменение направления линии потока производится под углом 90°.

Соединитель. Отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии потока и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение (рисунок 5.7).

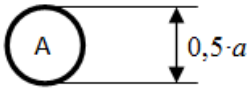

Рисунок 5.7 – Блок «Соединитель»

Комментарий. Применяют для добавления описательных комментариев или пояснительных записей в целях объяснения операций, выполняемых отдельными символами или группой символов. Пунктирные линии в символе

комментария связаны с соответствующим символом или могут обводить группу символов. Текст комментариев должен быть помещен около ограничивающей фигуры (рисунок 5.8).

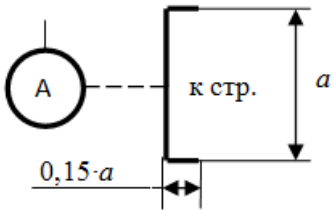

Рисунок 5.8 – Блок «Комментарий»

Граница цикла. Используется для обозначения цикла с условием. Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещены внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие (рисунок 5.9).

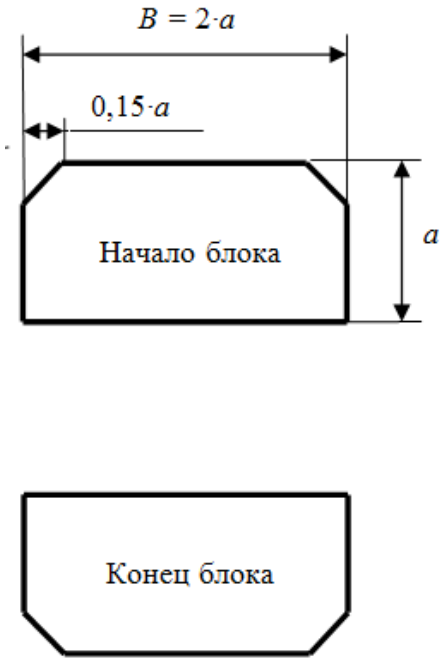

Рисунок 5.9 – Блок «Граница цикла»

Форма символов должна соответствовать [1], рекомендуется соблюдать пропорции, обозначенные на приведенных ранее изображениях, в качестве значения «а» задается целое число 2, 3 и т. д.

Каждому символу может присваиваться идентификатор (цифра, символ или их комбинация), который должен располагаться слева над символом. Идентификатор символа используют при описании работы алгоритма.

При наличии в блок-схеме ветвления над линиями связи пишут слово «да», если линия управления изображает поток при значении логического выражения

«True», и слово «нет», если линия управления изображает поток при значении логического выражения «False».

Пример использования графических элементов для построения блок-схем показан в приложении Б.

В разделе **«Текст программы»** необходимо привести текст программы, включая программные модули, процедуры и функции, созданные в ходе разработки программы.

Раздел **«Описание использованных в программе операторов, подпрограмм, свойств и методов классов»** содержит перечень операторов, их синтаксис и правила работы, описание стандартных и пользовательских подпрограмм (процедур и функций), применяемых в тексте программы [2–4]. Если в тексте программы использованы классы среды Delphi или пользователя, то необходимо представить описание методов и свойств этих объектов.

При описании подпрограмм и методов приводят их назначение, заголовок, включая имя, список входных и выходных параметров с указанием типа и назначения параметров. Для свойств описывают имя свойства и тип принимаемых значений.

В разделе **«Тестирование и анализ результатов решения задачи»** приводят сведения о проверке правильности работы программы. Применяют наборы входных данных, для которых результат работы программы заранее известен. Входные данные подбирают так, чтобы охватить все возможные варианты работы алгоритма программы.

Оценка тестирования заключается в сравнении результатов, полученных при выполнении теста, с заранее известными контрольными значениями. В программах решения задач аналитической и исследовательской геометрии оценивают влияние наборов входных данных на характер движения и возможность существования кинематических связей между элементами схемы.

Раздел **«Заключение»** содержит итоги выполненной работы по созданию программы, ее назначение, степень соответствия требованиям задания, перечень выполненных работ, рекомендации по возможному улучшению эффективности работы созданной программы.

В разделе **«Список литературы»** приводят перечень использованных источников (не менее 5–6 ссылок), в том числе ГОСТов, справочников по языку программирования и т. п. В тексте пояснительной записки на все источники должны быть приведены ссылки.

Список использованных источников составляется по мере появления ссылок в тексте пояснительной записки и оформляется в соответствии с требованиями [3]. Пример оформления списка использованных источников приведен в приложении Б.

6 Оформление расчетно-пояснительной записки

Курсовую работу оформляют в виде пояснительной записки. В пояснительной записке, используя общепринятую научно-техническую терминологию, сжато, логично и аргументированно излагают порядок выполнения всех этапов курсовой работы.

Пояснительную записку следует оформлять в соответствии с требованиями [5].

Пояснительная записка курсовой работы выполняется на листах бумаги формата А4 на одной стороне. Основную надпись выполняют по ГОСТ 2.104–68.

Титульный лист является первой страницей пояснительной записки, перенос слов на нем не допускается, точки в конце заголовков не ставятся (приложение А).

Страницы записки нужно нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему ее тексту. Номер страницы проставляют в правой нижней части листа без точки.

Нумерацию листов производят начиная с титульного, при этом на нем самом номер не ставится.

На листе содержания используют рамку высотой 40 мм, на всех остальных листах – 15 мм (рисунки 6.1 и 6.2).

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="4" rowspan="2"><Шифр></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Изм.</td><td>Лист</td><td>№ докум.</td><td>Подпись</td><td>Дата</td><td colspan="4"></td></tr>
<tr><td>Разраб.</td><td>Фамилия И. О.</td><td></td><td></td><td colspan="2" rowspan="5"><Название работы>
Пояснительная записка</td><td>Лит.</td><td>Лист</td><td>Листов</td></tr>
<tr><td>Провер.</td><td>Фамилия И. О.</td><td></td><td></td><td></td><td></td><td>3</td><td>25</td></tr>
<tr><td>Реценз.</td><td></td><td></td><td></td><td colspan="3" rowspan="3">Бел.-Рос. ун-т
ПДМ-191</td></tr>
<tr><td>Н. Контр.</td><td></td><td></td><td></td></tr>
<tr><td>Утверд.</td><td></td><td></td><td></td></tr>
</table>

Рисунок 6.1– Основная надпись высотой 40 мм

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td rowspan="2"><Шифр></td><td>Лист</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>4</td></tr>
<tr><td>Изм.</td><td>Лист</td><td>№ докум.</td><td>Подпись</td><td>Дата</td><td colspan="2"></td></tr>
</table>

Рисунок 6.2 – Основная надпись высотой 15 мм

<Шифр> имеет следующую структуру:

ПДМ-191.00. <№ зачетки> ПЗ

где ПДМ-191 – код специальности и номер группы;

<№ зачетки> – последние три цифры номера зачетной книжки.

<Название работы> – тема, указанная в задании на курсовую работу.

Текст должен быть набран в редакторе Word и отпечатан на принтере.

Текстовую часть документа выполняют шрифтом Times New Roman высотой 14 pt. Размеры полей: правое – 10 мм, верхнее – 15 мм, левое – 25, нижнее – 30 мм, межстрочный интервал – одинарный. Абзацный отступ в тексте записки устанавливают равным 10 мм, производят выравнивание текста по ширине.

Текст документа при необходимости разбивают на разделы и подразделы.

Разделы должны иметь порядковые номера в пределах всего документа, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и номера подраздела. В конце номера подраздела точку не ставят.

Разделы, как и подразделы, должны состоять из одного или нескольких пунктов. Каждый новый раздел необходимо начинать с нового листа, подразделы следуют друг за другом через двойной интервал. Текст после заголовка раздела и подраздела печатается через один интервал. Если между разделом и подразделом нет текста, то расстояние между ними три интервала. Заголовки разделов и подразделов выделяют полужирным шрифтом и выравнивают по ширине. Названия разделов «Содержание», «Ведение», «Заключение» и «Список литературы» располагают по центру.

Внутри текста записки могут быть приведены перечисления. Перед каждым перечислением следует ставить дефис или при необходимости ссылки в тексте документа на одно из перечислений строчную букву, после которой ставят скобку.

Иллюстрации (рисунки, графики, диаграммы, блок-схемы) и таблицы располагают в записке непосредственно после текста, в котором они упоминаются впервые, или на следующей странице, если в указанном месте они не помещаются.

Иллюстрации нумеруют арабскими цифрами. Применяют сквозную нумерацию в пределах всего документа, состоящую из одной цифры – «Рисунок 1». Допускается нумеровать иллюстрации в пределах раздела или подраздела. В этом случае номер иллюстрации должен состоять из номера раздела – «Рисунок 1.1». Наименование иллюстрации выполняют шрифтом 12 pt и размещают под иллюстрацией после пояснительных данных, выполняемых шрифтом 10 pt. Точка в конце подписи к иллюстрации не ставится.

Таблицы нумеруют арабскими цифрами. Применяют сквозную нумерацию в пределах всего документа, состоящую из одной цифры «Таблица 1». Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы «Таблица 1.1». Таблицы при необходимости могут иметь название, которое располагают через дефис за номером таблицы «Таблица 1.1 – Исходные данные». Наименование таблицы выполняют шрифтом 12 pt и помещают перед таблицей. Точка после наименования таблицы не ставится.

На все иллюстрации и таблицы в тексте должны быть даны ссылки, при этом слова «Рисунок» и «Таблица» пишут полностью: «из рисунка 2.1 следует...», «в таблице 1.2 приведены...».

Формулы следует записывать в общем виде с новой строки, выравнивание формулы производят симметрично тексту. Формулы необходимо набирать в редакторе формул. Если формула одна и требуется пояснение символов, входящих в формулу, то за формулой ставят запятую, в противном случае ставят точку. Формулы, следующие одна за другой и не разделенные текстом, разделяют точкой с запятой. При этом за последней формулой ставят либо запятую, если необходима расшифровка символов формул, либо точку, если символы формул не надо расшифровывать.

Все формулы должны нумероваться сквозной нумерацией арабскими цифрами, которые записывают на уровне формулы справа в круглых скобках. Применяют либо сквозную нумерацию в пределах всего документа, состоящую из одной цифры (1), либо в пределах раздела или подраздела, тогда номер должен состоять из номера раздела или подраздела и порядкового номера формулы, разделенных точкой (3.1).

Пояснения символов, входящих в формулы, если они не пояснены ранее в тексте, должны быть приведены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться словом «где», после которого ставят пробел, затем приводят обозначение символа и через дефис дают описание физического смысла символа с указанием его размерности.

При использовании формул, научно-технических положений, стандартов и др. данных необходимо делать ссылку на использованный источник, указывая его номер из списка литературы в квадратных скобках. Источники располагают в порядке появления ссылок на них в тексте. Запись источника в списке литературы производят в следующей последовательности: фамилия и инициалы автора (авторов), название, место издания (город), издательство, год издания и объем в страницах [6].

Приложения представляют собой материал, дополняющий текст пояснительной записки. Приложениями могут быть, например, графический материал, таблицы большого формата, расчеты, описания схем алгоритмов, тесты программ и т. д.

Приложения оформляют как продолжение курсовой работы. Каждое приложение должно начинаться с новой страницы с указанием вверху посередине страницы слова «приложение» и его обозначения, а под ними в круглых скобках строчными буквами указывают вид приложения (обязательное, рекомендуемое или справочное). Далее с новой строки симметрично относительно текста записывают с прописной буквы заголовок приложения. Приложения обозначают буквами русского алфавита, начиная с буквы А. При ссылках на приложения в тексте следует писать «в соответствии с приложением А ...».

Пример оформления пояснительной записки представлен в приложении Б.

7 Общие требования к программному модулю

Программный модуль, разрабатываемый в курсовой работе, должен удовлетворять следующим требованиям:

– среда программирования Delphi;

– структура программы – многооконная, состоящая из главной формы и вспомогательных форм для решения первой и второй задач соответственно;

– программа должна иметь заставку, появляющуюся при запуске, на которой указана тема проекта, группа и фамилия исполнителя;

– вывод результатов осуществляют на экран.

8 Защита курсовой работы

Курсовая работа допускается к защите после ее проверки и соответствующего разрешения руководителя. Защищают работу перед комиссией с участием руководителя в установленные сроки. Для защиты студент готовит краткий доклад, в котором освещаются основные вопросы разработки, затем студент отвечает на вопросы, поставленные членами комиссии.

Оценка выставляется на основе анализа пояснительной записки, доклада, демонстрации программных средств и ответов на вопросы.

Список литературы

1 **ГОСТ 19.701–90.** Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Москва: Стандартинформ, 2010. – 23 с.: ил.

2 **Эйдлина, Г. М.** Delphi: программирование в примерах и задачах. Практикум: учебное пособие / Г. М. Эйдлина, К. А. Милорадов. – Москва: РИОР: ИНФРА-М, 2017. – 116 с.

3 **Белов, В. В.** Программирование в Delphi: процедурное, объектно-ориентированное, визуальное: учебное пособие для вузов / В. В. Белов, В. И. Чистякова. – 2-е изд., стер. – Москва: Гор. линия-Телеком, 2014, 2015. – 240 с.: ил.

4 **Фленов, М. Е.** Библия Delphi. Практическое руководство / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2011. – 668 с.

5 **ГОСТ 2.105–95.** Единая система конструкторской документации. Общие требования к текстовым документам. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 1995. – 28 с.: ил.

6 **ГОСТ 7.1–2003.** Библиографическая запись. Библиографическое описание. Общие требования и правила оформления. – Москва: ИПК Изд-во стандартов, 2004. – 141 с.: ил.

Приложение А
(справочное)

Образец оформления титульного листа

Министерство образования Республики Беларусь

Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

Кафедра «Транспортные и технологические машины»

КУРСОВАЯ РАБОТА

по дисциплине «Информатика»
Разработка программ для решения задач в среде DELPHI

Пояснительная записка

ПДМ-191.00.124 ПЗ

Выполнил: студент группы ПДМ-191
_____Казаков А.Н.

Руководитель:
_____Сёмчен В. И.

Могилев 2020 г.

Приложение Б
(справочное)

Пример выполнения задания

Б.1 Задача 1. Вычисление корней нелинейного уравнения

Б.1.1 Постановка задачи.

В соответствии с заданием на курсовую работу необходимо найти с точностью 0,001 все корни уравнения и расположить их в порядке возрастания. Нахождение корней необходимо выполнить на интервале значений аргумента от –2 до 2.

Представим уравнение в виде функции аргумента *x*:

$$f(x) = x^3 - 2{,}92 \cdot x^2 + 1{,}4355 \cdot x + 0{,}791136. \qquad \text{(Б.1)}$$

Исходными данными для работы программы являются начальное и конечное значения аргумента и точность расчета.

Результат работы программы – представим в числовом виде отдельно найденные корни и их расположение в порядке возрастания.

Б.1.2 Математическая модель.

Корнем уравнения называют такое значение аргумента, при котором функция обращается в ноль.

Решить нелинейное уравнение – значит найти все его корни или доказать, что корней нет.

Проведем исследование существования корней уравнения (Б.1), определим их количество и расположение в искомом интервале для этого вычислим значения функции таблица Б.1 и построим график функции (рисунок Б.1).

Таблица Б.1 – Значения аргумента и функции

x	*y*	*x*	*y*
–2	–21,7599	0,5	0,903886
–1,5	–11,3071	1	0,306636
–1	–4,56436	1,5	–0,25061
–0,5	–0,78161	2	–0,01786
0	0,791136		

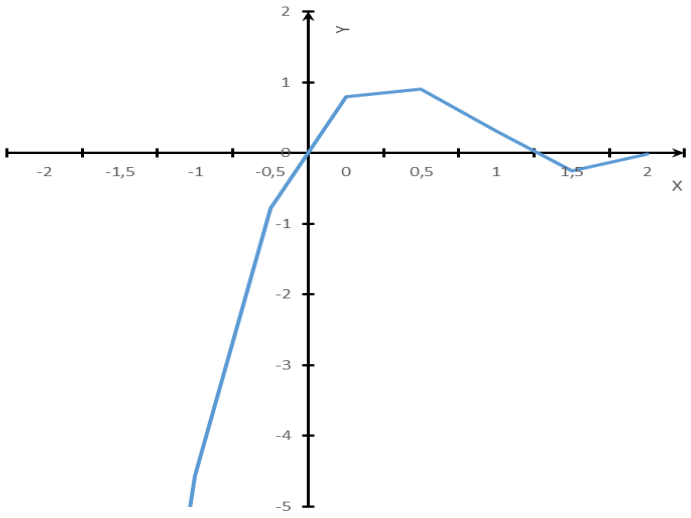

Рисунок Б.1 – График функции

Анализируя результаты, приходим к выводу, что на исследуемом интервале имеется два корня, т. к. значение функции меняет знак 2 раза и график функции дважды пересекает ось абсцис.

Определим интервалы изоляции корня (отрезки локализации) так, чтобы на интервале этот корень был единственным. Таким образом, получаем два интервала локализации [-0.5;0] и [1;1,5].

Для решения нелинейных уравнений используем итерационный метод. Итерационный процесс решения уравнения $F(x) = 0$ состоит в последовательном уточнении некоторого начального приближения x_0. В результате итераций находят последовательность приближенных значений корня $x_1, x_2, \ldots, x_n$ к точному решению.

Основными итерационными методами являются метод дихотомии (половинного деления), метод Ньютона (касательных) и последовательного приближения к значению корня на отрезках длиной равной точности вычисления.

Выбираем последний метод; как более простой и застрахованный от неудачи, он всегда приводит к решению, если на отрезке есть хотя бы один корень, позволяет найти несколько корней на одном отрезке без их предварительной локализации.

Для реализации метода необходимо последовательно рассматривать отрезки длиной равной точности, при этом, если знаки функции на концах отрезка различны корень есть, иначе корень отсутствует.

Началом первого отрезка принимаем начало отрезка локализации корня a.

Определим конец первого рассматриваемого отрезка:

$$x = a + e, \tag{Б.2}$$

где e – заданная точность вычисления, $e = 0{,}001$.

Выполним расчет значений функции на концах отрезка $f(a)$ и $f(x)$.

$$f(a) = a^3 - 2{,}92 \cdot a^2 + 1{,}4355 \cdot a + 0{,}791136;$$
$$f(x) = x^3 - 2{,}92 \cdot x^2 + 1{,}4355 \cdot x + 0{,}791136. \quad (Б.3)$$

Проверим знаки функции на концах отрезка, вычислив значение логического выражения

$$f(a) \cdot f(x) < 0. \quad (Б.4)$$

Если условие в выражении (Б.4) выполняется, то знаки значений функции на концах отрезка различны и один из корней будет лежать в рассматриваемой области значений аргумента.

Рассмотрим следующий отрезок его начало a поместим в точку x – конец предыдущего отрезка – и далее выполним вычисления по формулам Б.2 и Б.3.

Данный процесс позволяет проверить все отрезки длиной, равной точности на интервале локализации корня; итерации должны продолжаться пока не будет достигнут конец интервала локализации корня и выполняется условие $a < b$.

Б.1.3 Алгоритм решения задачи.

Составим алгоритм решения задачи в графической форме (рисунки Б.2–Б.5).

Рисунок Б.2 – Блок-схема алгоритма процедуры очистки формы с обнулением счетчика корней

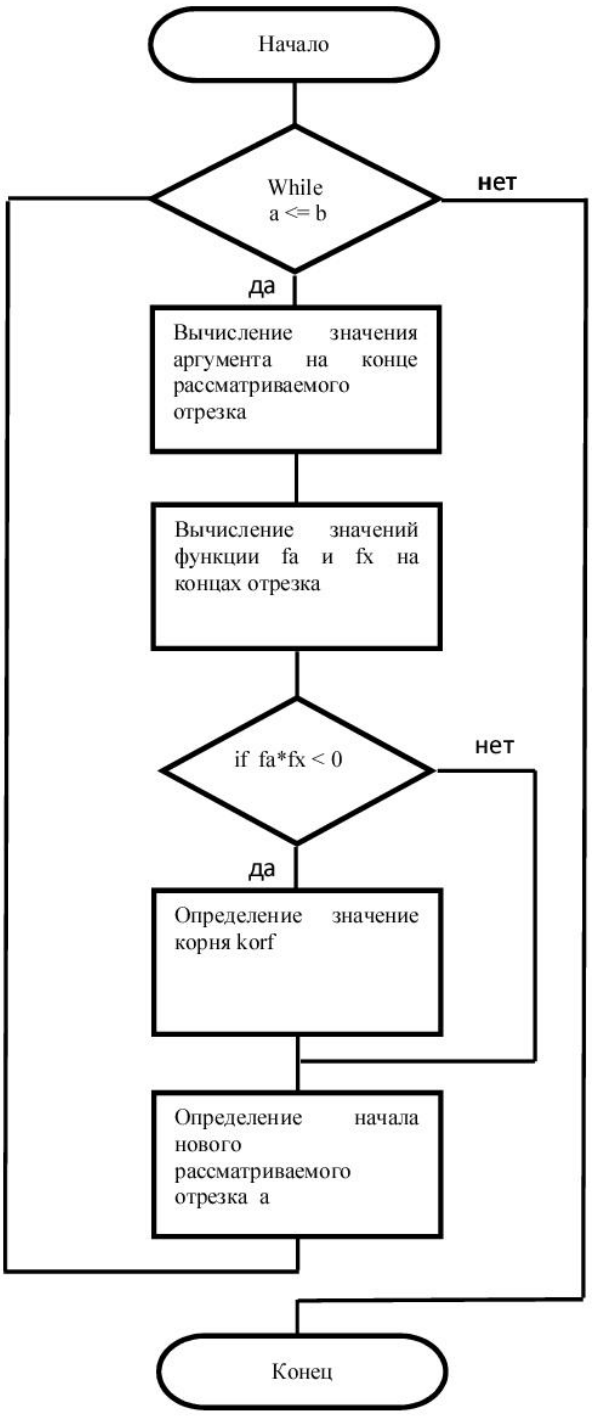

Рисунок Б.3 – Блок-схемы алгоритмов процедуры LocKor

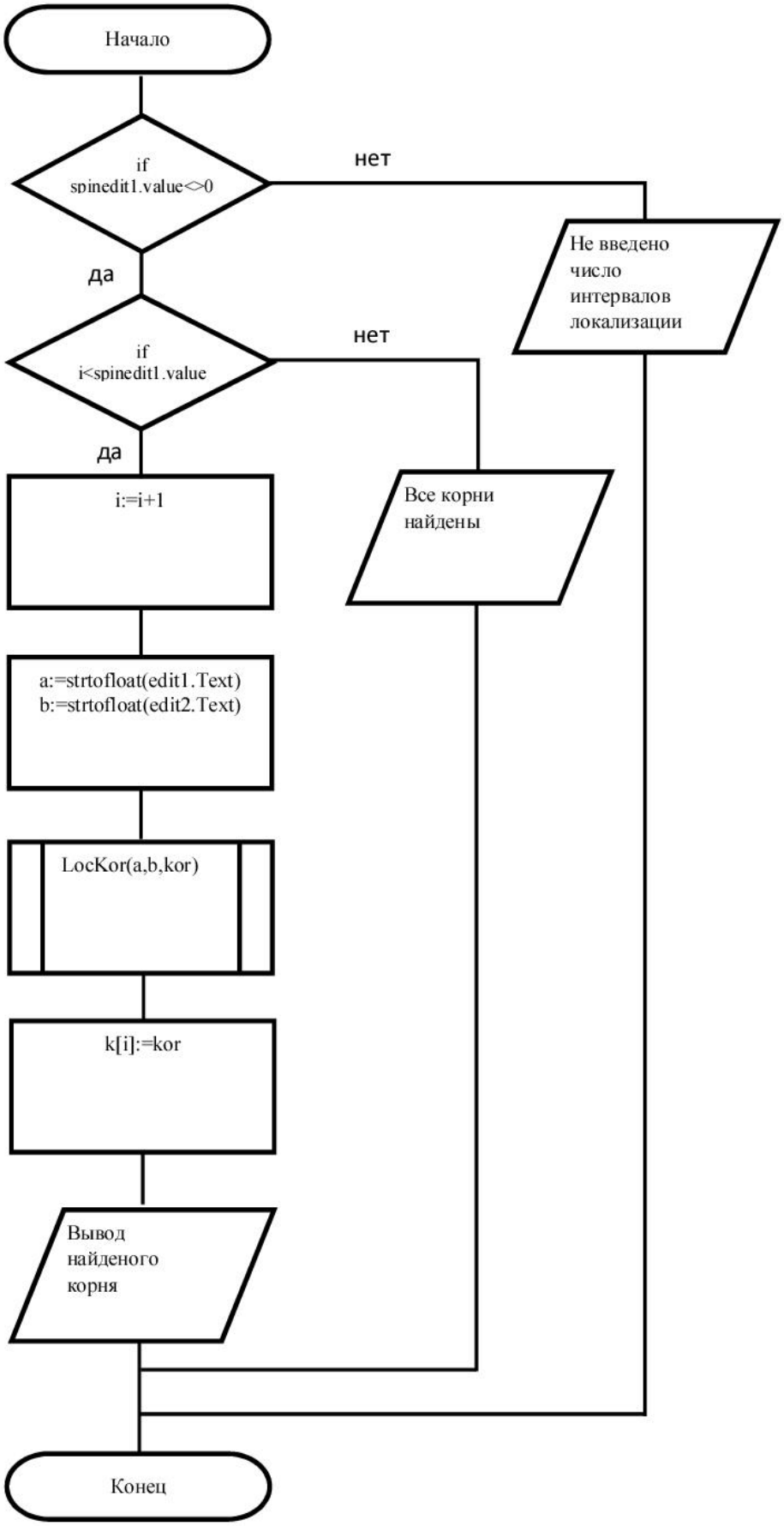

Рисунок Б.4 – Блок-схема алгоритма нахождения корней

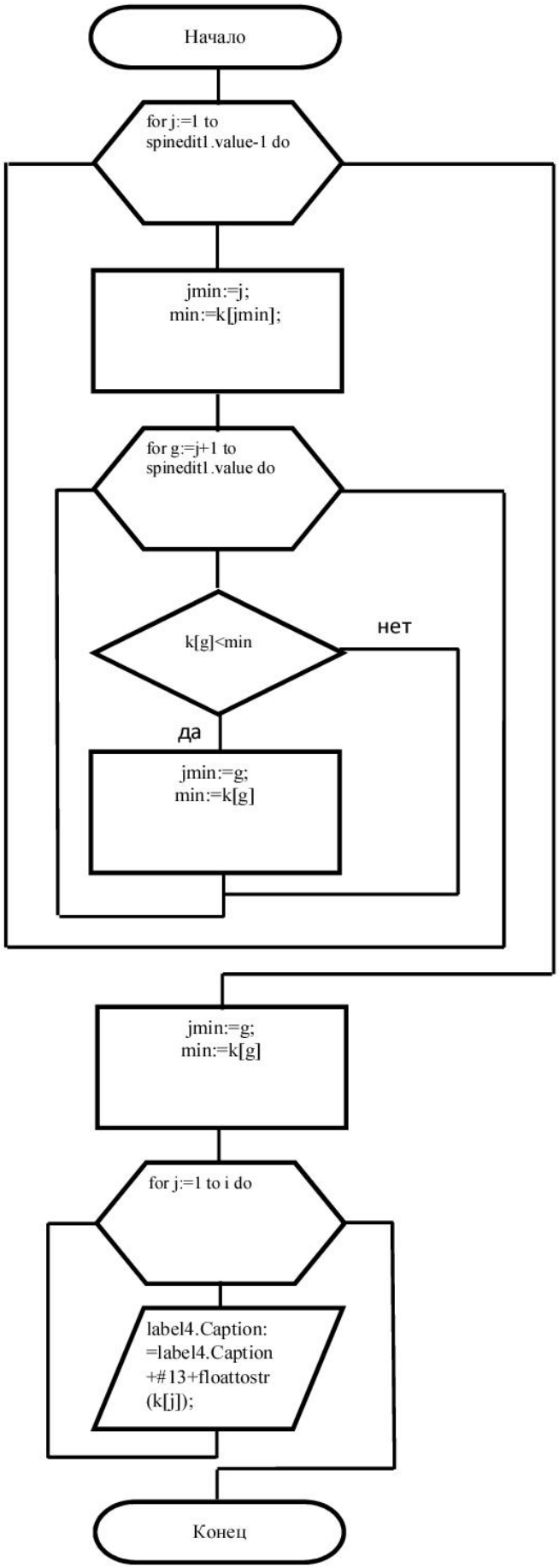

Рисунок Б.5 – Блок-схема алгоритма расположения корней в порядке возрастания

Б.1.4 Интерфейс программы.

Для организации диалогового режима работы с пользователем программы разработаем графический интерфейс, используя визуальные компоненты среды Delphi.

Структура интерфейса базируется на использовании трех окон формы Form (рисунок Б.6), предназначенных для организации диалогового режима с пользователем и управления работой программ.

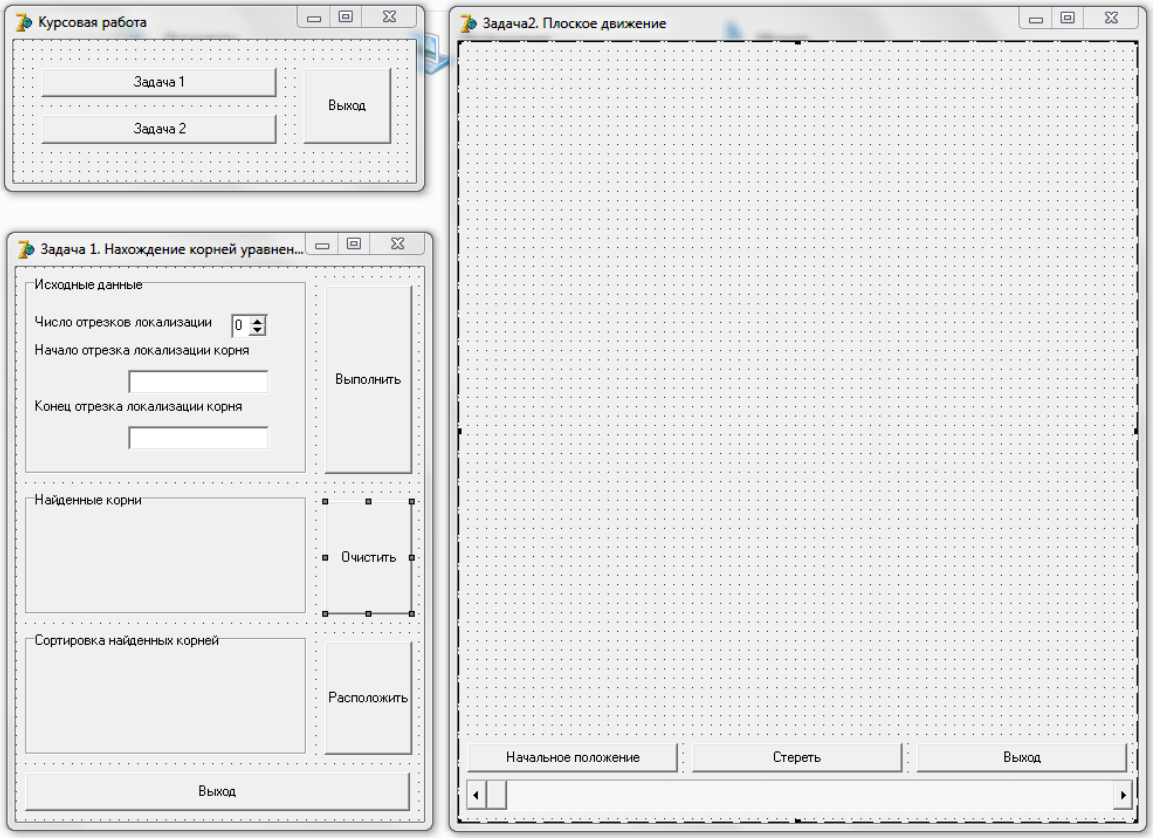

Рисунок Б.6 – Структура интерфейса

Главное окно программы (рисунок Б.7) предназначено для вывода информации о программе и вызова дополнительных окон, в которых производится решение первой и второй задач соответственно.

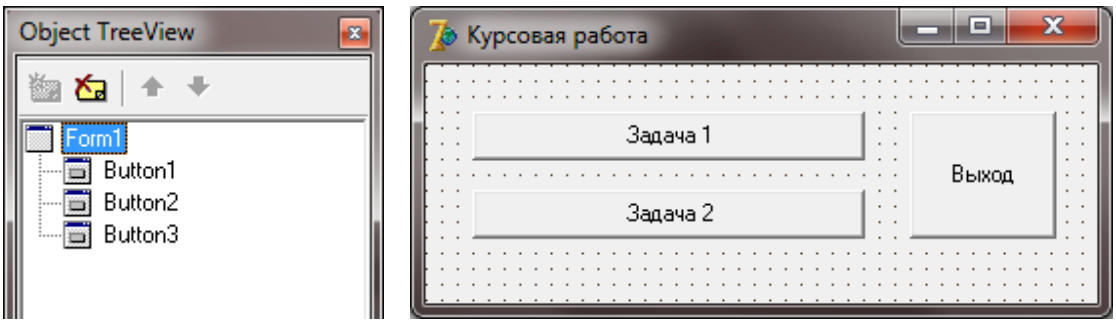

Рисунок Б.7 – Главная форма приложения

На форме размещены следующие компоненты Button – кнопки, компоненты типа TButton. Для добавления надписей на компонентах Button изменено свойство Caption. Связь нажатия на кнопку с текстом программы реализована с помощью обработчика события OnClick этих компонентов.

При нажатии кнопки Button 1 («Задача 1») выполняется вызов окна; при нажатии Button 2 («Задача 2») происходит вычисление. При нажатии Button 3 («Выход») закрывается приложение.

Окно программы для решения задачи 1 (рисунок Б.8) содержит компоненты Edit типа TEdit, однострочное редактируемое поле ввода для отображения или редактирования одной текстовой строки. Применяется в программе для ввода начала и конца отрезков локализации корней. Для организации ввода используется свойство Text.

Spinedit1 – компонент типа TSpinedit, управляемое окно ввода. Применяется в программе для указания числа отрезков локализации определенных матмоделью. Для организации ввода используется свойство Value.

Button – кнопки, компоненты типа TButton. Для добавления надписей на компонентах Button изменено свойство Caption. Связь нажатия на кнопку с текстом программы реализована с помощью обработчик события OnClick этих компонентов.

При нажатии кнопки Button 1 («Выполнить») отыскивается корень на указанном отрезке локализации. Button 2 («Очистить») – отвечает за удаление результатов. Button 3 («Выход») обеспечивает закрытие текущего окна программы, Button 4 («Расположить») выполняет сортировку найденных корней.

Label – метка, компонент типа TLabel. Этот компонент используется для подписи других компонентов. Надписи на компонентах Label выполнены с помощью свойства Caption.

На форме расположены метки для подписи:

– Label1 («Начало отрезка локализации корня») – поля ввода Edit1;

– Label2 («Конец отрезка локализации корня») – поля ввода Edit2;

– Label5 («Число отрезков локализации») – компонента Spinedit1.

Компоненты Label3 и Label4 предназначены для вывода результатов и до вывода результата имеют пустые значения свойства Caption.

GroupBox – компонент типа TGroupbox. Использован для группировки и разделения элементов окна по назначению, для вывода названия блоков изменено свойство Caption.

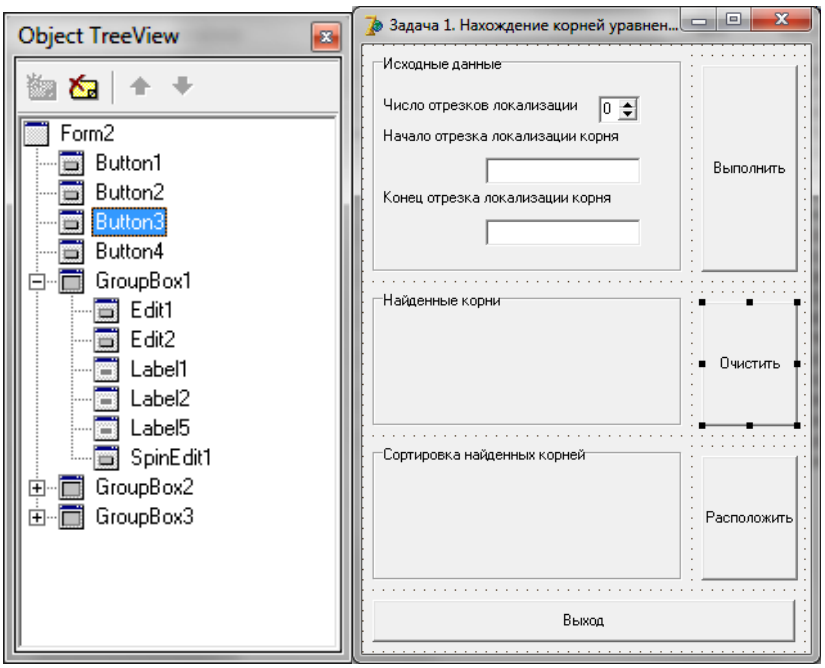

Рисунок Б.8 – Окно для вычисления корней уравнения

Б.1.5 Текст программного модуля.

В данном разделе приведен текст программы на языке Delphi для решения поставленной задачи.

```
unit Unit2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Spin;
type
TForm2 = class(TForm)
GroupBox1: TGroupBox;
GroupBox2: TGroupBox;
Button1: TButton;
Button2: TButton;
Edit1: TEdit;
Edit2: TEdit;
Label1: TLabel;
Label2: TLabel;
GroupBox3: TGroupBox;
Label3: TLabel;
Label4: TLabel;
Button3: TButton;
SpinEdit1: TSpinEdit;
Label5: TLabel;
Button4: TButton;
procedure Button1Click(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
Form2: TForm2;

type mas=array[1..4] of real;

var a,b,fa,fx,x,kor,min:real;
jmin,j,g,i:integer;
k:mas;
const e=0.001;

implementation

{$R *.dfm}
Procedure LocKor(af,bf:real; var korf:real);
begin
  while a<=b do
  begin
    x:=a+e;
    fa:=a*a*a-2.92*a*a+1.4355*a+0.791136;
    fx:=x*x*x-2.92*x*x+1.4355*x+0.791136;
    if fa*fx<0 then korf:=x;
    a:=x;
end;
end;

procedure TForm2.Button1Click(Sender: TObject);
begin
  if spinedit1.value<>0 then
    begin
      if i<spinedit1.value then
        begin
          i:=i+1;
          a:=strtofloat(edit1.Text);
          b:=strtofloat(edit2.Text);
          LocKor(a,b,kor);
          k[i]:=kor;
```

```
        label3.Caption:=label3.Caption+'Корень'+inttostr(i)+'='+floattostr(k[i])+#13;
       end
     else showmessage('Все корни найдены')
   end
  else showmessage('Не введено количество интервалов локализации);
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
  label3.Caption:='';
  label4.Caption:='';
  i:=0;
  edit1.Text:='';
  edit2.Text:=''
end;

procedure TForm2.Button4Click(Sender: TObject);
begin
  Label4.Caption:='';
  for j:=1 to spinedit1.value-1 do
  begin
    jmin:=j;
    min:=k[jmin];
      for g:=j+1 to spinedit1.value do
        if k[g]<min then
        begin
        jmin:=g;
        min:=k[g]
        end;
    k[jmin]:=k[j];
    k[j]:=min;
end;

for j:=1 to i do
  label4.Caption:=label4.Caption+#13+floattostr(k[j]);
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
  close;
end;

end.
```

Б.1.6 Описание основных операторов, процедур, функций и методов, использовавшихся в программе.

В процессе создания программы использовались следующие стандартные функции:

– IntToStr (Number : Integer) : string – преобразует целое число в строку;

– FloatToStr (Value : Extended) : string – преобразует вещественное число в строку;

– StrToFloat (FloatString : string) : Extended – преобразует строку в целое число.

Для решения задачи отыскания корней создана процедура LocKor.

LocKor(af,bf:real; var korf:real) – входными параметрами af, bf вещественного типа являются границы отрезка локализации корней, в выходном параметре korf вещественного типа хранится значение найденного корня для передачи в основную программу.

Составной оператор – простая структура, состоящая из следующих друг за другом операторов, заключенных в операторные скобки begin … end и выполняющихся как один оператор.

Begin
<оператор 1>
<оператор 2>

<оператор N>
end;

Составной оператор применяется в тех случаях, когда синтаксис языка Delphi допускает использование только одного оператора, в то время как алгоритм требует задания некоторой последовательности действий при этом простой заменяют составным.

Оператор повтора for используют, когда известно количество повторений цикла. Он обеспечивает выполнение тела цикла до тех пор, пока оно не будет выполнено со всеми значениями переменной цикла.

for i:=in **to** ik **do** <оператор>;

где i – переменная цикла;

in – начальное значение переменной цикла;

ik – конечное значение переменной цикла.

Переменная цикла должна быть целого типа, переменную цикла в теле цикла изменять нельзя, в данной разновидности оператора for должно выполняться условие выполнения цикла in <= ik.

Работает оператор for в следующем порядке. Принимается начальное значение переменной цикла, вычисляется тело цикла, управление возвращается к оператору for и принимается следующее значение переменной цикла с шагом единица (переменная увеличивается на 1, затем опять выполняется тело цикла.

После того как тело цикла выполнится со всеми значениями переменной цикла, управление передается оператору, следующему за for.

Условный оператор if.

Использованы полная

If <логическое выражение> **then** <оператор 1> **else** <оператор 2>;

и сокращенная форма

If <логическое выражение> **then** <оператор 1>;

Если условие истинно, то выполняется оператор, следующий за словом then, и управление передается следующему оператору за if. В противном случае, когда условие ложно, будет выполняться оператор, следующий за словом else и управление передается следующему оператору за if. В усеченном в случае если условие оказывается ложным, то управление сразу передается следующему оператору за if.

Оператор присваивания.

Имя переменной:= выражение;

Символы «:=» обозначают операцию присваивания, в соответствии с которой сначала вычисляется выражение слева, а затем получившийся результат в виде значения записывается в идентификатор. Тип идентификатора должен соответствовать типу выражения.

В программе использованы методы обработки событий, происходящих на форме, код которых создан во время работы приложения.

Procedure Button1Click (Sender ; Tobject) – обработчик события OnClick на компоненте Button1 для запуска программы.

Procedure Button2Click (Sender ; Tobject) – обработчик события OnClick на компоненте Button2 для закрытия приложения.

Procedure TForm1.Edit1Change(Sender: TObject); – обработчик события OnGhange на компоненте Edit1 для заполнения массива.

Б.1.7 Тестирование и анализ результатов решения задачи.

Проверим правильность работы программы путем запуска ее на выполнение с исходными данными, для которых результат определен при составлении математической модели и ее анализе. Выполним созданную программу для первого интервала локализации (рисунок Б.9).

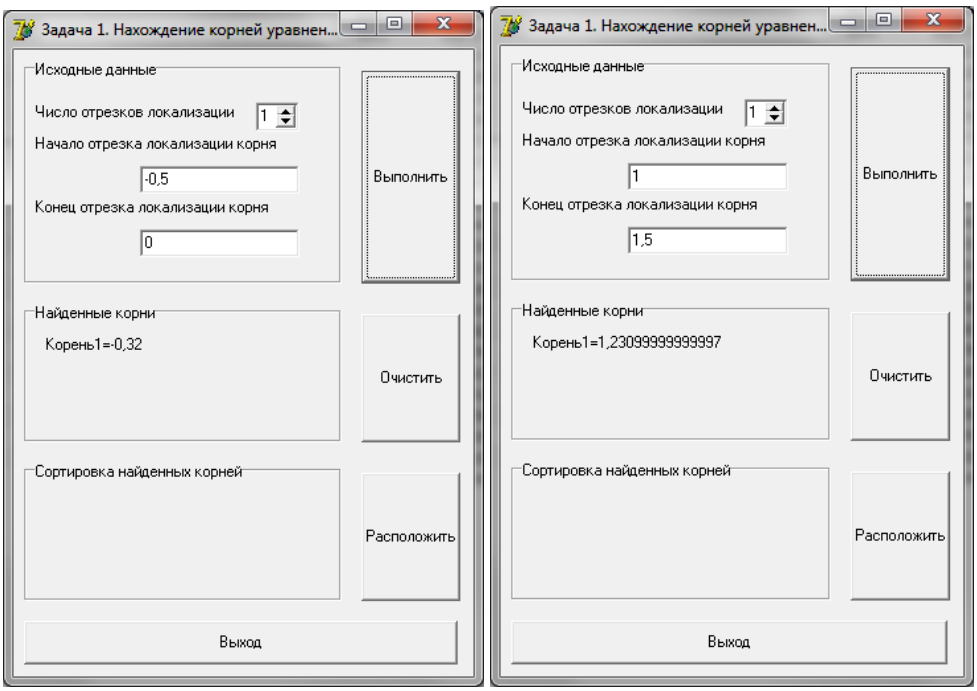

Рисунок Б.9 – Результаты вычисления корня на каждом интервале

Сравнивая полученные тестовые значения, полученные при анализе математической модели, и результаты работы программы, можно сделать вывод о хорошей сходимости полученных значений и точности расчета.

Проведем исследование работы программы для двух участков локализации с последующей сортировкой значений (рисунок Б.10).

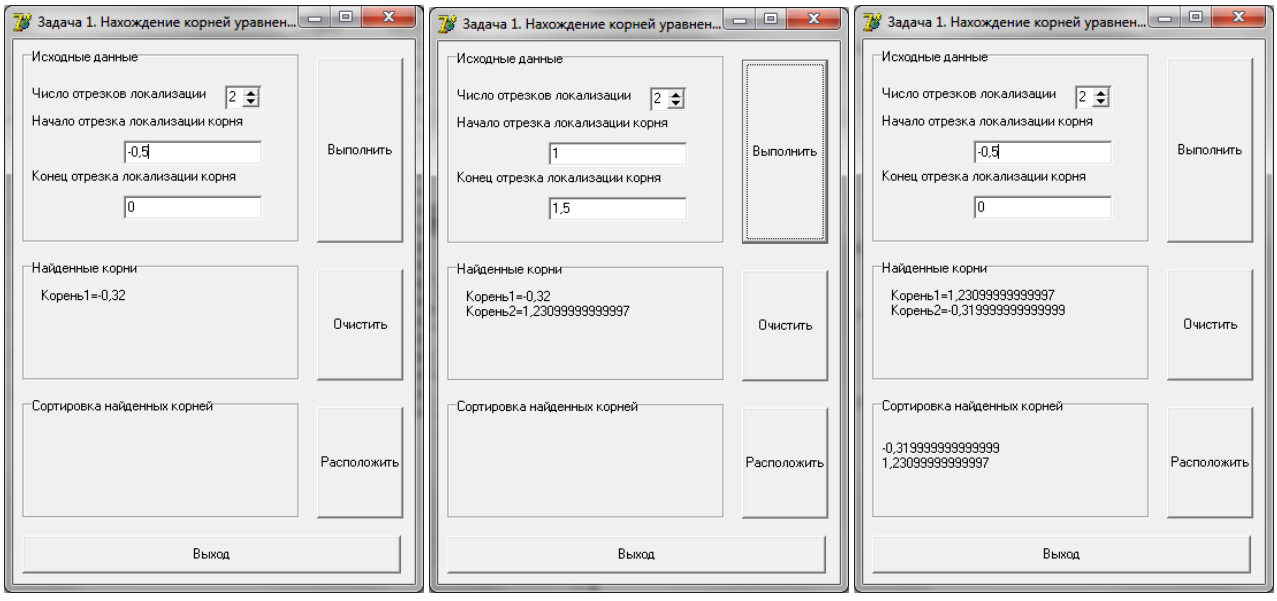

Рисунок Б.10 – Результаты вычисления всех корней на двух интервалах и расположения их в порядке возрастания

Результаты работы подтверждают правильность программной реализации выбранного численного метода, соответствие найденных корней установленным при начальном анализе тестовым значениям и правильность реализации сортировки корней по возрастанию.

Б.2 Задача 2. Решение задачи аналитической геометрии и анимация движения механизма на плоскости

Б.2.1 Постановка задачи.

Создать программу для визуального отображения перемещений механизма в соответствии с заданной схемой кинематической схемой.

На схеме в задании представлен трехзвенный механизм. Исходными данными являются вертикальная координата точки *A*, ее изменение в некотором интервале значений определяет пользователь программы.

Согласно схеме, известны и неизменны обе координаты точки *O*, а для точек *A* и *C* известна и неизменна одна из координат, остальные координаты точек, представленных на схеме, подлежат вычислению в программе в зависимости от заданных пользователем начальных параметров.

Программа должна обеспечить расчет координат всех точек механизма при изменении начальных параметров и вывод графического изображения начального положения механизма, очистку текущего изображения на экране и анимацию движения механизма при изменении пользователем исходных данных.

Б.2.2 Математическая модель.

Для составления математической модели построим расчетную схему (рисунок Б.11), проанализируем работу механизма, выявим зависимости, однозначно определяющие положение механизма на плоскости.

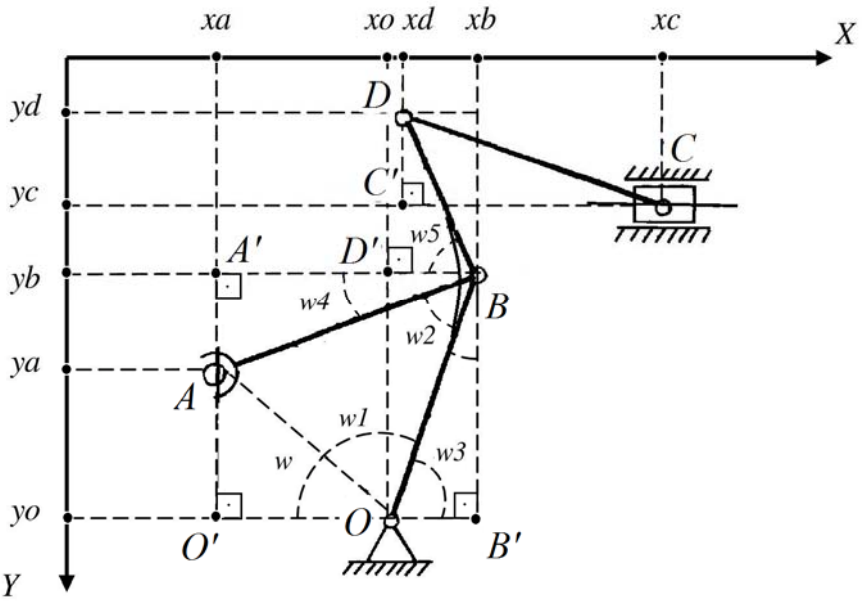

Рисунок Б.11 – Расчетная схема механизма

Для определения координат точек в расчетной схеме принята система координат, соответствующая координатной системе программного свойства canvas (холст) компонента типа TImage на котором будет производится рисование в соответствии с составленной математической моделью.

Длины звеньев *AB*, *OB*, *BD* и *DC* известны.

Координаты *yc*, *xo*, *yo*, *xa* и угол *OBD* = 120º неизменны.

Определим положение опоры в точке *O*, приняв ее в центре компонента, предназначенного для вывода изображения:

$$xo = image1.Width/2; \quad (Б.5)$$

$$yo = image1.Height/2. \quad (Б.6)$$

Определим неизменные координаты опор горизонтальной точки *A* и вертикальной точки *C* на некотором расстоянии от опоры *O*:

$$xa = xo - 100; \quad (Б.7)$$

$$yc = yo - 200. \quad (Б.8)$$

Расстояние задается произвольно и корректируется при необходимости при тестировании программы.

Положение механизма определяет координата ya. При изменении данного параметра каждому его значению будет соответствовать новое положение механизма и соответственно координаты определяющих это положение точек.

Приращение значения *ya* принимается из свойства компонента на форме и отсчитывается относительно опоры в точке *O*:

$$ya = scrollbar1.position + yo. \quad (Б.9)$$

Найдем длину отрезка AO и угол w из треугольника AOO':

$$AO = \sqrt{(xo-xa)^2+(yo-ya)^2}; \quad (Б.10)$$

$$w = \arccos((xo-xa)/ao). \quad (Б.11)$$

Найдем углы *w*1 и *w*2 из треугольника *AOB* по теореме косинусов:

$$w1 = \arccos(((BO*BO)+(AO*AO)-(AB*AB))/(2*BO*AO)); \quad (Б.12)$$

$$w2 = \arccos(((BO*BO)+(AB*AB)-(AO*AO))/(2*BO*AB)). \quad (Б.13)$$

Найдем угол *w*3 в треугольнике *OBB*':

$$w3 = w1 - w. \quad (Б.14)$$

Тогда координату точки *B* определим из треугольника *OBB*':

$$xb=xo-bo\cdot\cos(w3); \quad (Б.15)$$

$$yb=yo-bo\cdot\sin(w3). \quad (Б.16)$$

Найдем угол *w*4 в треугольнике *ABA*':

$$w4=\arccos((xb-xa)/ab). \quad (Б.17)$$

Найдем угол *w*5 в треугольнике *BDD*' и определим координату точки *D*:

$$w5=ugol-w4-w2; \quad (Б.18)$$

$$xd=xb-bd\cdot\cos(w5); \quad (Б.19)$$

$$yd=yb-bd*\sin(w5). \quad (Б.20)$$

Найдем координату *xc* из треугольника *DCC*':

$$xc=xd+\sqrt{dc^2-(yc-yd)^2}. \quad (Б.21)$$

Математические зависимости для определения координат элементов оформления (окружности, прямоугольник, треугольник и текст) являются простейшими арифметическими зависимостями, отдельно в программе не высчитываются и в данной математической модели не представлены.

Б.2.3 Описание алгоритма решения задачи.

На основании полученной математической модели и особенностей структуры создаваемой программы составим алгоритм решения в вербальной форме для:

– определения координат и прорисовки звеньев механизма в начальном положении:

а) произвести инициализацию объекта класса, выделить память;

б) определить неизменные параметры механизма и рассчитать координаты опор;

в) задать приращение управляющего параметра;

к) вычислить геометрические параметры механизма, определяющие положение его звеньев в пространстве;

д) произвести очистку поля вывода и стереть текущее положение механизма;

е) вывести изображение нового положения звеньев механизма;

– определения координат и прорисовки звеньев механизма в движении:

а) принять начальное значение управляющего параметра;

б) произвести инициализацию объекта класса, выделить память;

в) определить неизменные параметры механизма и рассчитать координаты опор;

г) вычислить геометрические параметры механизма, определяющие положение его звеньев в пространстве;

д) произвести очистку поля вывода и стереть текущее положение механизма;

е) вывести изображение начального положения звеньев механизма.

Б.2.4 Интерфейс программы.

Окно программы для решения задачи 2 представлено на рисунке Б.12.

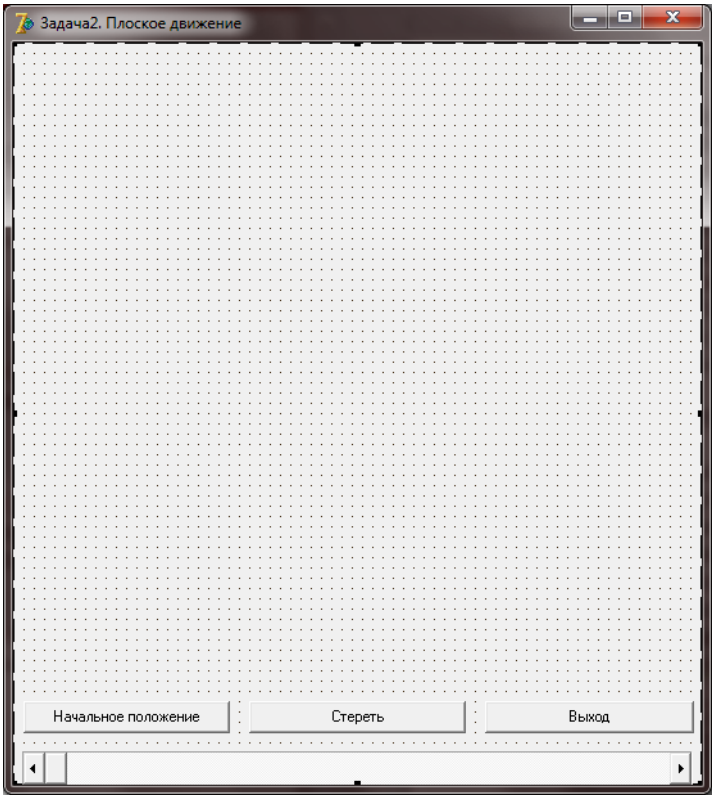

Рисунок Б.12 – Интерфейс программы для решения задачи 2

Форма содержит компоненты Button – кнопки, компоненты типа TButton. Для добавления надписей на компонентах Button изменено свойство Caption. Связь нажатия на кнопку с текстом программы реализована через обработчик события OnClick этих компонентов.

При нажатии кнопки Button 1 («Начальное положение») выводится изображение механизма для начальных условий, при нажатии Button 2 («Стереть») удаляется изображение механизма. При нажатии Button 3 («Выход») закрывается окно программы.

Image1 – компонент типа TImage для вывода изображения, используются свойства Picture при очистке и программное свойство Canvas для прорисовки механизма.

ScrollBar1 – полоса прокрутки, компонент типа TScrollBar. Предназначен для управления параметром, определяющим положения механизма, и,

соответственно, прорисовкой всех этих положений. Значение управляющего параметра принимается из свойства Position, зависящего от положения ползунка данного компонента. Связь перемещения ползунка с текстом программы реализована с помощью обработчика события OnChange этого компонента.

Б.2.5 Текст программного модуля.

В данном разделе приведен текст программы на языке Delphi для решения поставленной задачи.

```
unit Unit3;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, math;

type
TForm3 = class(TForm)
ScrollBar1: TScrollBar;
Image1: TImage;
Button1: TButton;
Button2: TButton;
Button3: TButton;
procedure ScrollBar1Change(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

TShem=class
xa,ya,xc,yc,xb,yb,xd,yd,xo,yo,AB,BO,BD,DC,AO,qw:integer;
ugol,w,w1,w2,w3,w4,w5:real;
procedure Data;
procedure Draw;
end;
var
Form3: TForm3;
Shem1:TShem;

implementation

uses Unit1;
```

```
{$R *.dfm}

procedure TShem.Data;
begin
   AB:=200;
   BO:=140;
   BD:=100;
   DC:=260;
   ugol:=2*pi/3 ;
     xo:=round(Form3.Image1.Width/2);
     yo:=round(Form3.Image1.Height/2);
     xa:=xo-100;
     yc:=yo-200;
       ya:=Form3.scrollbar1.position+yo;
         AO:=round(sqrt(sqr(xo-xa)+sqr(yo-ya)));
         w:=arccos((xo-xa)/ao);
         w1:= arccos(((BO*BO)+(AO*AO)-(AB*AB))/(2*BO*AO));
         w2:= arccos(((BO*BO)+(AB*AB)-(AO*AO))/(2*BO*AB));
         w3:=w1-w;
           xb:=xo-round(bo*cos(w3));
           yb:=yo-round(bo*sin(w3));
             w4:=arccos((xb-xa)/ab);
             w5:=ugol-w4-w2;
               xd:=xb-round(bd*cos(w5));
               yd:=yb-round(bd*sin(w5));
                 xc:=xd+round(sqrt((dc*dc)-sqr(yc-yd)));
end;

procedure TShem.Draw;
begin
form3.Image1.Picture:=nil;
with form3.Image1.canvas do
begin
  pen.Width:=2;
  pen.Color:=clred;
  MoveTo(xo,yo);
  LineTo(xb,yb);
  LineTo(xa,ya);
  MoveTo(xb,yb);
  LineTo(xd,yd);
  LineTo(xc,yc);
    pen.Color:=clblue;
    pen.Width:=2;
    Rectangle(xc-20,yc-10,xc+20,yc+10);
    MoveTo (xo,yo);
```

```
    LineTo (xo+25,yo+40);
    LineTo (xo-25,yo+40);
    LineTo (xo,yo);
      pen.Color:=clgreen;
      ellipse(xa-6,ya-6,xa+6,ya+6);
      ellipse(xb-6,yb-6,xb+6,yb+6);
      ellipse(xc-6,yc-6,xc+6,yc+6);
      ellipse(xd-6,yd-6,xd+6,yd+6);
      ellipse(xo-6,yo-6,xo+6,yo+6);
        textout(xa-10,ya+5,'A');
        textout(xb+10,yb-5,'B');
        textout(xc,yc-30,'Ñ');
        textout(xd-15,yd-20,'D');
        textout(xo+15,yo-10,'O');
end;

end;
procedure TForm3.ScrollBar1Change(Sender: TObject);
begin
  Shem1:=TShem.create;
  Shem1.Data;
  Shem1.Draw;
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
  Form3.scrollbar1.position:=50;
  Shem1:=TShem.create;
  Shem1.Data;
  Shem1.Draw;
end;

procedure TForm3.Button3Click(Sender: TObject);
begin
  form3.Image1.Picture:=nil;
end;

end.
```

Б.2.6 Описание основных операторов, процедур, функций и методов, использовавшихся в программе.

Создание программы решения задачи аналитической геометрии и анимации движения механизма на плоскости потребовало использования операторов языка Delphi, стандартных подпрограмм и методов, подпрограмм пользователя.

При создании программы для вычислений использовались следующие стандартные функции:

– round(X:Extended):Int64 – округляет число с плавающей запятой X до целого значения по банковскому правилу;

– sin(X: Extended):Extended – математическая функция, возвращающая в радианах синус числа X;

– cos(X: Extended):Extended – математическая функция, возвращающая в радианах косинус числа X;

– arccos(X: Extended):Extended – математическая функция, выдающая значение арккосинуса числа X в радианах;

– sqrt (X: Extended):Extended – возвращает квадратный корень числа X;

– sqr(X: Extended):Extended, sqr(X:Integer):Integer – возвращает квадрат числа X, число может быть целого или вещественного типа;

Для прорисовки звеньев механизма использовались методы программного объекта canvas на компоненте Image1:

– MoveTo(X, Y: Integer) – перемещает курсор в позицию, заданную координатами X иУ;

– LineTo(X, Y: Integer) – рисует отрезок в позицию, заданную координатами X и Y;

– Ellipse(X1, Y1, X2, Y2: Integer) – рисует эллипс, вписанный в прямоугольник с координатами левого верхнего угла XI, Y1 и правого нижнего угла X2, Y2;

– Rectangle(Xl,Yl,X2,Y2: Integer) – рисует прямоугольник с координатами левого верхнего угла XI, Y1 и правого нижнего угла X2, Y2;

– Textout(X,Y: Integer; Text:string) – выводит строковую константу Text в поле с координатами верхнего левого угла X,Y, шрифт определяется значением свойства Font объекта canvas.

Цвет и толщина линий изображаемого механизма определялись комплексным свойством pen программного объекта canvas:

– pen.Width – свойство Width комплексного свойства pen отвечает за толщину линии;

– pen.Color – свойство Color комплексного свойства pen отвечает за цвет линии.

Оператор присоединения **with** служит для удобной и быстрой работы с составными частями объектов и сокращения длины идентификаторов методов и свойств используемых объектов.

Оператор имеет следующий синтаксис

with <выражение> **do** < оператор>;

Работает оператор следующим образом: выражение, представляющее имя составного объекта, присоединяется к оператору, содержащему составную часть объекта, расположенному после слова **do**.

Оператор присваивания описан в п п. Б.1.6.

Задание на проектирование предусматривает создание классов пользователя. В результате создан класс TShem, наследующий методы класса TObject и инкапсулирующий поля, определяющие координаты точек и методы, отвечающие за расчет координат и прорисовку механизма.

В качестве методов класса использованы две пользовательские процедуры без параметров:

– procedure Data – отвечает за расчет координат точек;

– procedure Draw – отвечает за очистку изображения и прорисовку нового положения механизма.

В тексте программы использованы методы обработки событий:

– procedure ScrollBar1Change(Sender: TObject) – обработчик события OnChange на компоненте ScrollBar1 для изменения управляющего параметра ya, выделения памяти под объект Shem1 и вызова методов Data и Draw для получения изображения;

– procedure Button3Click(Sender: TObject) – обработчик события OnClick на компоненте Button3 выполняет те же функции, что и ScrollBar1Change, но при начальном значении управляющего параметра;

– procedure Button3Click(Sender: TObject) – обработчик события OnClick на компоненте Button1 для закрытия формы приложения.

Б.2.7 Тестирование и анализ результатов решения задачи

Проверим правильность работы программы путем запуска ее на выполнение и изменения входного параметра в диапазоне значений, определяемом свойствами min = 0 и max = 100 компонента Scrollbar1.

Проведем исследование работы программы в разных режима вывода изображения (рисунки Б.13, Б.14).

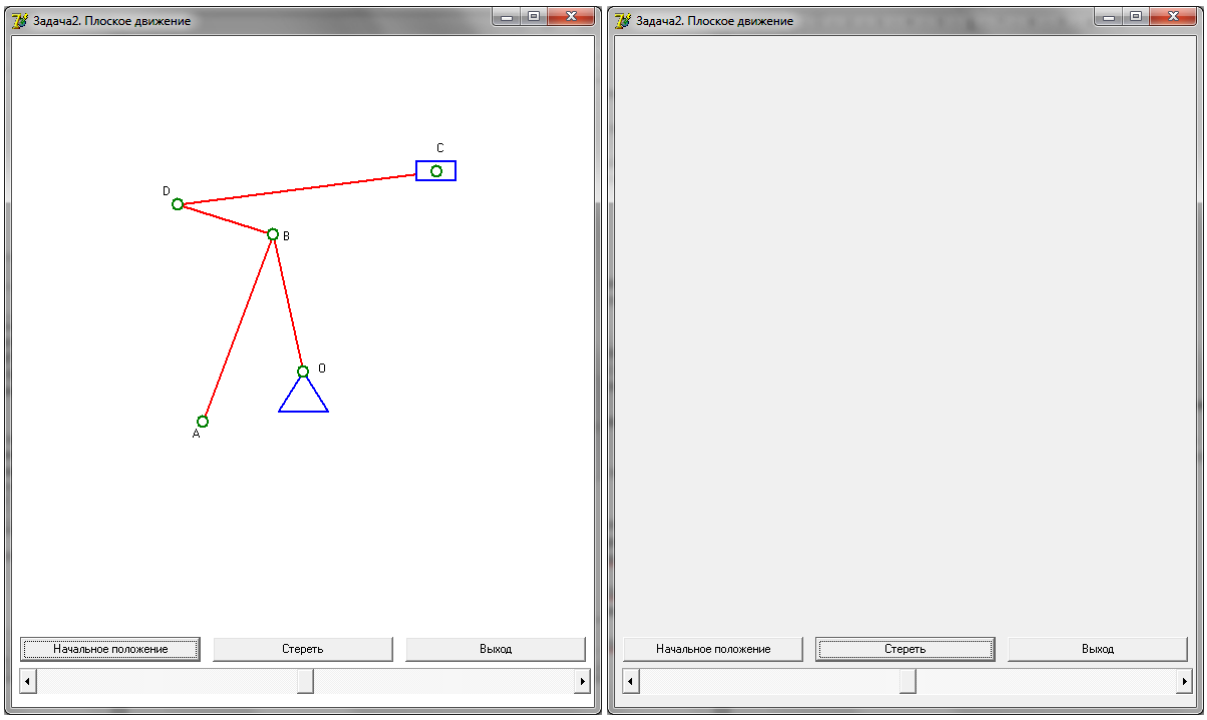

Рисунок Б.13 – Результаты прорисовки механизма начального положения и очистки

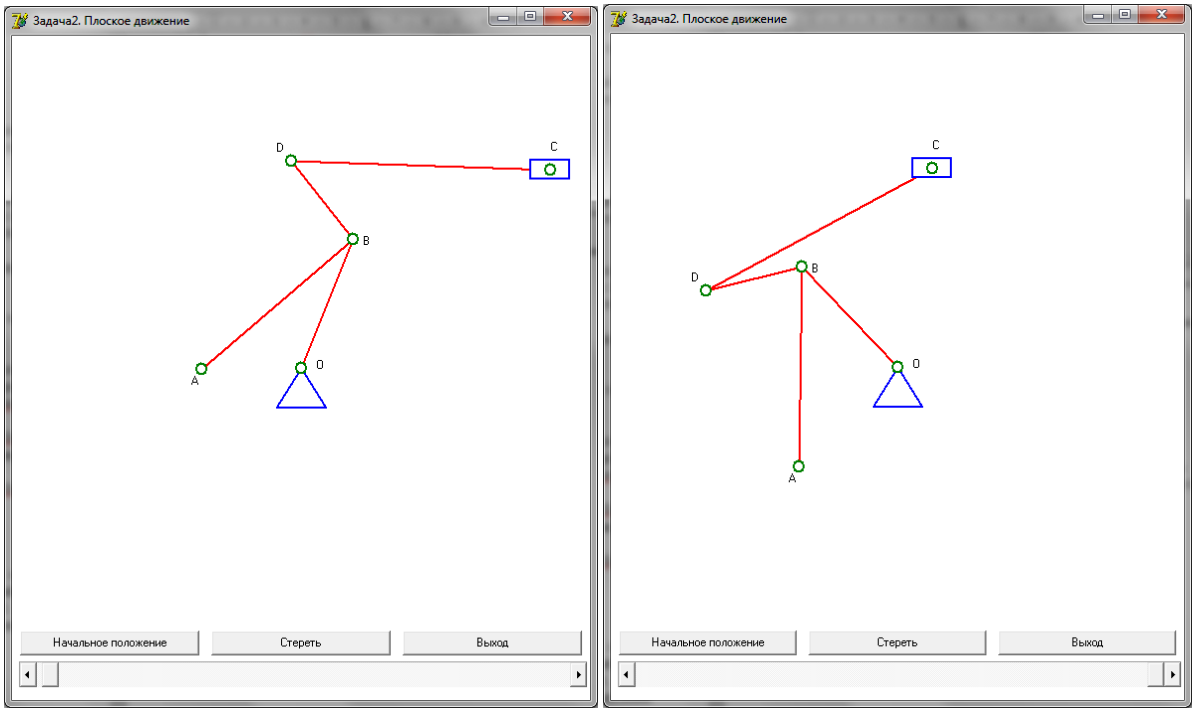

Рисунок Б.14 – Результаты прорисовки механизма в движении

Результаты работы подтверждают достоверность составленной математической модели для выбранного диапазона значений и правильность ее программной реализации.