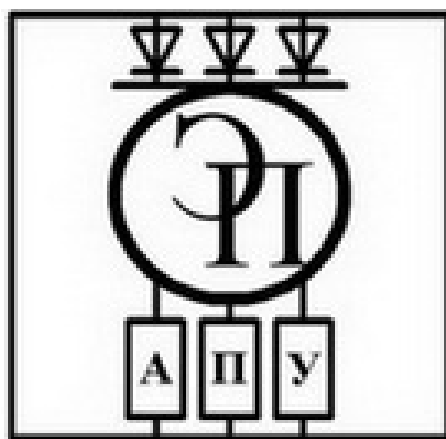


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и автоматизация промышленных установок»

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ

*Методические рекомендации к курсовому проектированию
для студентов направления подготовки
13.03.02 «Электроэнергетика и электротехника»
дневной формы обучения*



Могилев 2024

УДК 004.42
ББК 32.973-018
О75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Электропривод и АПУ» «31» августа 2023 г.,
протокол № 1

Составитель канд. физ.-мат. наук, доц. О. В. Обидина

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации к курсовому проектированию разработаны в соответствии с рабочей программой по дисциплине «Основы алгоритмизации и программирование» для студентов направления подготовки 13.03.02 «Электроэнергетика и электротехника» дневной формы обучения.

Учебное издание

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЕ

Ответственный за выпуск	А. С. Коваль
Корректор	И. В. Голубцова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

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

© Белорусско-Российский
университет, 2024

Введение

Современное состояние науки и техники требует от инженерно-технических и научных работников знания средств вычислительной техники и умения обращения с современными программно-техническими комплексами. Эффективное использование компьютеров для решения инженерных и научных задач невозможно без знаний основных методов составления схем алгоритмов, написания эффективного программного обеспечения на языке программирования, использования пакетов программ инженерной графики и математических систем.

Цель данной курсовой работы – закрепление на практике знаний, полученных при изучении курса «Основы алгоритмизации и программирование», и приобретение опыта самостоятельной разработки прикладного программного обеспечения для решения расчетных задач для персонального компьютера.

1 Содержание курсовой работы

Цель выполнения курсовой работы – закрепить практические навыки, приобретенные в процессе изучения дисциплины. Курсовая работа представляет собой разработку приложения для Microsoft Windows и программной документации с помощью программного обеспечения ПК. Каждому студенту выдается индивидуальное задание на курсовую работу.

Курсовая работа включает следующие этапы работы, каждый из которых оформляется как отдельный раздел пояснительной записки.

- 1 Анализ требований к разрабатываемому приложению.
- 2 Описание алгоритмов приложения.
- 3 Проектирование интерфейса приложения.
- 4 Программирование работы приложения.
- 5 Тестирование приложения.

Курсовая работа включает пояснительную записку объемом 25–35 страниц формата А4 и графическую часть объемом 1 лист формата А1. Следует пояснительную записку и графическую часть работы выполнять автоматизированным способом, используя изученные программные продукты.

2 Требования к содержанию пояснительной записки

2.1 Анализ требований к разрабатываемому приложению

В разделе четко определяются следующие требования к разрабатываемой программе:

- 1) тип операционной системы;
- 2) требования к составу аппаратной части;
- 3) состав и вид исходных данных, а также допустимый диапазон их значений;
- 4) условия проверки корректности введенных данных;
- 5) способ задания выражения задачи (если оно необходимо), при котором возможны следующие варианты: в виде подпрограммы, с помощью специального шаблона-интерпретатора, ввод в виде строки текста с последующей обработкой с помощью внешнего компилятора;
- 6) точность получения результата;
- 7) отображение хода процесса выполнения расчета;
- 8) возможность прерывания пользователем выполнения расчета и (или) автоматическое определение режима «зависания» или некорректных результатов;
- 9) вид результатов расчета.

2.2 Описание алгоритмов приложения

В рассматриваемом разделе производится проектирование алгоритмов решения задачи. Раздел включает следующие подразделы:

- разработка алгоритма головной программы;
- проектирование алгоритма ввода исходных данных;
- разработка алгоритма вывода результатов;
- проектирование алгоритма обработки матриц.

Прежде чем приступить к построению каждой схемы алгоритма, предварительно следует определить следующие данные:

- промежуточные результаты, необходимые для реализации вычислительного процесса. Результаты расчета непосредственно в программе не должны отображаться, но могут использоваться при ее отладке и верификации;
- выходные результаты решения, которые должны отображаться на экране или выводиться в текстовый файл.

По согласованию с руководителем работы возможно добавление других блок-схем, описывающих промежуточные этапы обработки данных.

При разработке алгоритмов следует использовать следующие методы:

- 1) метод частных целей, который заключается в сведении решения сложной задачи к рассмотрению последовательности более простых задач;
- 2) метод подъема, когда задается начальное самое простое предположение варианта построения алгоритма этапа решения задачи, которое затем последовательно улучшается, пока не достигнет заданного качества.

Эти два метода используются комплексно: сначала сложная задача разбивается на последовательность более простых подзадач, а затем для составления алгоритмов каждой из задач используется метод подъема или опять метод частных целей.

2.2.1 Разработка алгоритма головной программы.

Схема алгоритма головной программы разрабатывается с учетом интерактивного характера проектируемой программы-приложения. Поэтому следует учитывать событийный характер функционирования программы. При этом каждому возможному событию соответствует вызов определенного обработчика. Программа-приложение должна иметь обработчики следующих событий:

- 1) ввод исходных данных задачи;
- 2) расчет задачи;
- 3) вывод результатов расчетов;
- 4) получение справочной информации.

При составлении общей схемы алгоритма разрабатываемого приложения следует произвести описание обработки событий на главной форме проекта в виде цикла с предусловием. Событиями являются выбор пунктов меню или нажатие соответствующих управляющих компонентов, например кнопок. Общая схема приложения упрощенно может быть представлена на рисунке 2.1.

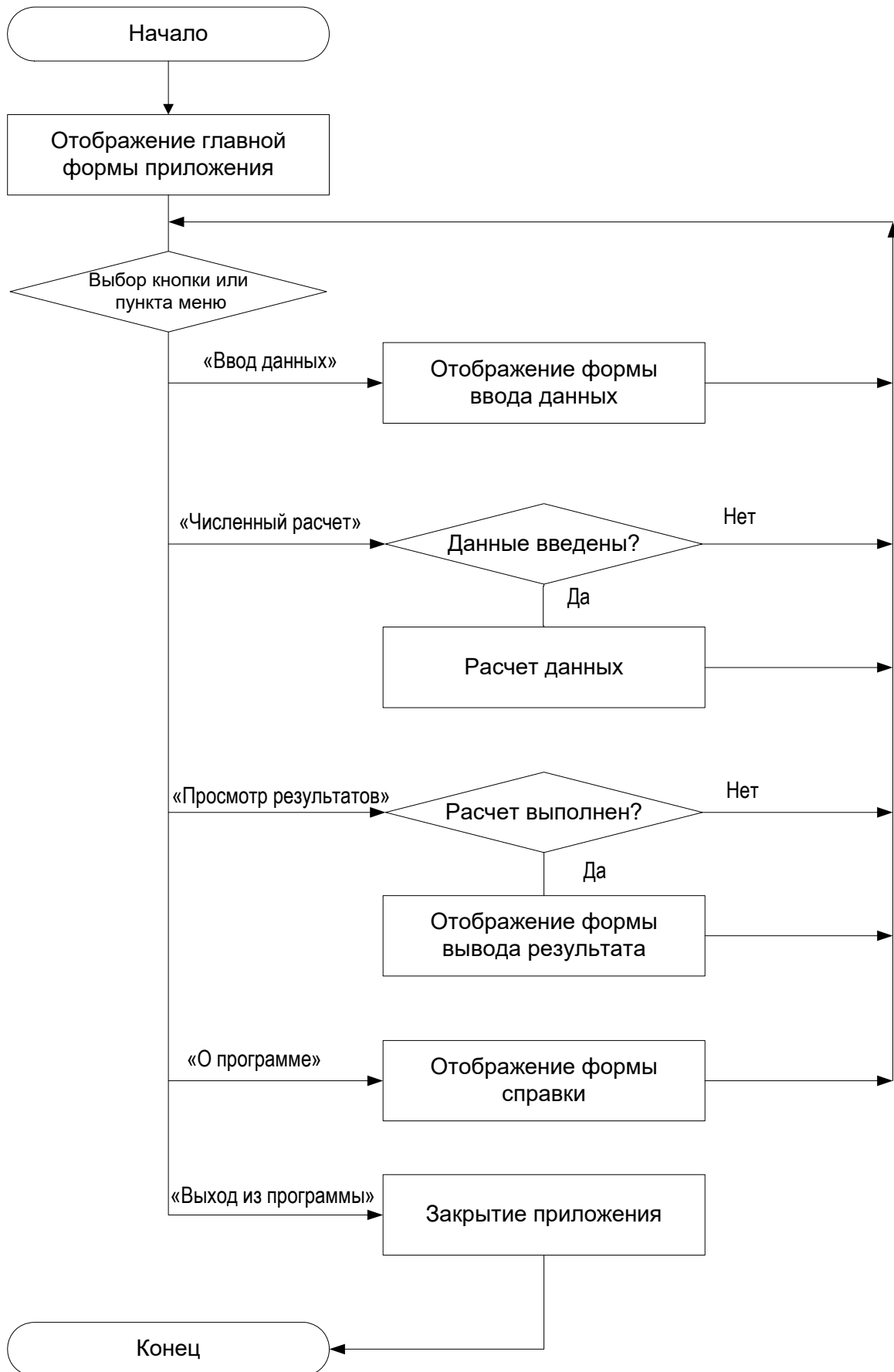


Рисунок 2.1 – Общая схема приложения

2.2.2 Проектирование алгоритма ввода исходных данных.

При проектировании алгоритма ввода исходных данных следует предусматривать следующие операции:

- ввод численных значений исходных данных с помощью клавиатуры или выбор варианта из списка с помощью переключателей мыши;
- проверка корректности введенных данных;
- сохранение введенных данных в отдельный файл;
- возможность загрузки исходных данных из ранее сохраненного файла.

2.2.3 Разработка алгоритма вывода результатов.

Алгоритм вывода результатов расчета должен обеспечивать следующее:

- вывод результатов расчета на экран в виде таблицы или численных значений;
- построение графика искомой зависимости (для решения дифференциальных уравнений обязательно);
- сохранение результатов таблиц в файл отчета формата rtf или txt;
- сохранение графика (если он использовался) в файл формата wmf;
- вывод результатов расчета для печати на системном принтере компьютера.

При выводе результатов в файл вначале следует показать значения исходных данных, при которых были получены результаты.

2.2.4 Проектирование алгоритма обработки матриц.

При разработке подраздела вначале оговариваются дополнительные переменные для промежуточных результатов расчета. Далее реализуется вычислительный алгоритм обработки матриц.

Правила построения схем алгоритмов.

Схема программы должна четко и однозначно отображать всю последовательность дискретных операций на рассматриваемом этапе. Согласно ГОСТ 19.701–90 в схемах алгоритмов используются:

- символы процесса, обозначающие операции обработки данных;
- линейные символы, указывающие направление потока управления;
- специальные символы, предназначенные для упрощения описания и чтения схемы: комментарии, разрывы и т. п.

Для каждой подпрограммы составляется схема алгоритма (схема программы). При этом внутри подпрограмм рекомендуется использовать вложенные подпрограммы, которые описывают функционально законченные части вычислительного алгоритма. Схемы алгоритма всех подпрограмм, за исключением стандартных подпрограмм языка программирования, изображаются в курсовой работе.

При построении схемы алгоритма программы должны соблюдаться следующие правила:

- символы в схеме алгоритма должны быть расположены равномерно с минимальным числом длинных линий;

– размеры сторон символов выбираются произвольно таким образом, чтобы не затруднялась идентификация символа;

– все символы схемы должны быть, по возможности, одного размера или, по крайней мере, иметь одинаковую ширину;

– символы схемы могут быть вычерчены в любой ориентации, но предпочтительной является горизонтальная; допускается зеркальное изображение формы символа;

– внутри символа отображается минимальное количество текста, необходимого для понимания функции данного символа, при этом текст для чтения должен записываться слева направо и сверху вниз независимо от направления потока. Если объем текста, помещаемого внутри символа, превышает его размеры, следует использовать символ комментария;

– в схеме алгоритма рекомендуется использовать цифровой идентификатор символов, который должен располагаться слева над символом. Рекомендуется в идентификаторе указывать через точку составной номер, первая цифра которого указывает номер основной подпрограммы, а вторая – номер символа в подпрограмме;

– в схеме алгоритма может использоваться описание символов – любая другая информация, например, для отображения специального применения символа с перекрестной ссылкой или для улучшения понимания функции как части схемы. Описание символа должно быть расположено справа над символом;

– в схеме алгоритма может использоваться подробное представление, которое обозначается горизонтальной полосой в его верхней части, указывающее, что в другом месте имеется более подробное представление. Между этой линией и верхней линией символа помещен идентификатор, указывающий на подробное представление данного символа. В качестве первого и последнего символа подробного представления должен быть использован символ терминатора. Первый символ указателя конца должен содержать ссылку, которая имеется также в символе с полосой;

– потоки данных или потоки управления в схеме алгоритма показываются сплошными линиями;

– направления потока управления слева направо и сверху вниз считаются стандартными и их допускается не указывать, во всех остальных случаях направление потока передачи указывается с помощью стрелок на концах или в середине линий;

– две входящие линии потока могут объединяться в одну исходящую линию, если число объединяемых линий превышает две. Место объединения каждой следующей присоединяемой линии должно быть смещено;

– линии потока в схемах алгоритмов должны подходить к символу процесса либо слева, либо сверху, а исходить либо справа, либо снизу. Линии должны быть направлены к центру символа;

– при необходимости линии потоков управления в схемах программ следует разрывать для избежания излишних пересечений или слишком длинных линий, а также, если схема состоит из нескольких страниц;

– при переносе части схемы алгоритма на другой лист используется символ «соединитель», который в начале разрыва называется внешним соединителем, а соединитель в конце разрыва – внутренним соединителем. Внутри них указывается одинаковое текстовое обозначение;

– при межстраничном переносе дополнительно к символу «соединитель» могут быть приведены ссылки к страницам внутри символов комментария.

Схемы алгоритмов должны изображаться на листе графической части работы в объеме не менее 80 % от площади заданного формата.

В пояснительной записке производится описание назначений блоков схемы алгоритма. При этом ссылки на блоки схемы алгоритма производятся путем указания номера блока.

2.3 Проектирование интерфейса приложения

При разработке интерфейса пользователя используются стандартные компоненты среды программирования с использованием библиотек стандартных элементов API. При реализации интерфейса в пояснительной записке приводится описание основных компонентов для каждой формы, которые будут использоваться в программе. Типовой интерфейс приложения в этом случае должен иметь:

– главную форму программы, которая раскрывается при запуске программы и содержит следующий минимальный набор элементов:

а) заголовок приложения с изображением собственной пиктограммы;

б) строки главного меню в верхней части формы;

в) строки состояния программы, где отображаются подсказки назначения текущих элементов меню;

– форму ввода исходных данных с клавиатуры, где собственно значения должны задаваться в полях ввода, реализованных с помощью однострочных редакторов или таблиц строк. Режимы работы программы можно задавать с помощью переключателей;

– форму отображения результата расчета, где расчетные данные отображаются в виде таблицы и графика; сохранение результатов в отдельном файле рекомендуется выполнять с помощью нажатия соответствующих кнопок;

– форму информации по программе.

Отдельно описывается состав разделов главного меню и кнопок с пояснением действий, которые они реализуют.

В записке следует описать структуру интерфейса каждой формы, указав использованные стандартные классы и пояснив настройку их отдельных свойств.

В подразделе следует привести копию экрана с каждой из форм, реализованной в проекте, которые оформляются в виде рисунков согласно требованиям ГОСТ 2.105–95.

2.4 Программирование работы приложения

Ввод и вывод данных в программе реализуется с помощью стандартных компонентов системы управления: однострочных редакторов, таблиц строковых данных, графиков и т. д. В пояснительной записке следует указать, каким образом в программе производится преобразование строковых данных полей в численные и численных данных в строковые значения. Возможно организовывать ввод-вывод значений с помощью стандартных диалоговых окон. Особенно эффективно их использование для указания имени файлов исходных данных или результатов, а также для задания настроек системного принтера.

Текст программы расчета разрабатывается на основании составленной блок-схемы алгоритма с использованием описанных в предыдущих разделах данных. При программировании интерфейса программы следует использовать ресурсы стандартных библиотек системы программирования для отображения текущего состояния выполненного объема расчета (с помощью строки процента завершения или указания номера текущей итерации). Возможно заполнение графиков и таблиц расчета динамически по мере расчета контрольных значений.

Текст операторной части (тела) подпрограммы заданного метода должен соответствовать изображенной схеме алгоритма. В тексте следует использовать комментарии для пояснений соответствия текста программы схеме алгоритма и локальных идентификаторов переменных и констант. Допускается при кодировании операторной части метода детализировать разработанный алгоритм, но запрещается пропускать указанные в схеме алгоритма операции.

Текст программы оформляется в виде распечатки на принтере в приложении пояснительной записки. Допускается на листах распечатки не отображать основную надпись. Тексты программ рекомендуется распечатывать с помощью равномерного шрифта (например, Courier New). Для текста приложений допустимой является высота шрифта не ниже 8 пт, а для основного текста записки – 14 пт.

Необходимо привести следующие исходные коды:

- файл проекта приложения;
- тексты всех разработанных классов.

2.5 Тестирование приложения

Цель выполнения данного раздела – проверка работоспособности разработанной при выполнении предыдущего раздела программы и ее отладка.

2.5.1 Расчет тестового примера.

Тестирование составленного текста программы должно производиться в три этапа:

1) тестирование устойчивости, когда проверяется реакция программы на некорректный ввод данных, значения которых выходят за допустимый диапазон;

2) тестирование функциональности, когда выполняется проверка правильности вычисления по известным исходным данным;

3) тестирование применимости, где выполняется проверка удовлетворения способов использования требованиям задания (время реакции приложения на команды пользователя, понятность интерфейса, устойчивость вычислительного процесса).

В пояснительной записке указывается, какое математическое выражение использовалось для тестирования и исходные данные. Для данного математического выражения и исходных данных выполняется расчет нескольких итераций вручную.

2.5.2 Решение задачи в математической системе Mathcad.

В рассматриваемом подразделе производится решение поставленной в задании задачи с использованием математического пакета Mathcad. Исходные данные для решения должны совпадать с исходными данными подраздела 5.1. В подразделе пояснительной записки описываются использованные стандартные функции математического пакета, а также дополнительные переменные. Результаты тестирования указываются в подразделе, а в приложении приводится копия рабочей области документа Mathcad.

2.5.3 Решение задачи в математической системе MATLAB.

В данном разделе составляется m-файл со стандартными функциями решения поставленной задачи и исходными данными, которые совпадают с использованными в подразделе 5.1. В приложение пояснительной записки помещаются результаты расчета в среде MATLAB в виде копии командного окна MATLAB.

3 Варианты заданий по курсовой работе

Задание на курсовую работу содержит тему работы и исходные данные (тип операционной системы, среду программирования и т. п.).

Тема работы может быть скорректирована по согласованию с руководителем курсовой работы.

Электронный вариант заданий для выполнения курсовой работы хранится в методическом кабинете кафедры (200/2) и в классе ПЭВМ кафедры (207/2).

4 Оформление курсовой работы

4.1 Оформление графической части

Графическая часть проекта выполняется на одном листе формата А1. В ней обязательно отображаются следующие схемы алгоритмов:

- общая схема работы приложения (для Windows-приложения) или схема головной программы (для консольного приложения);
- схема алгоритма ввода исходных данных;
- схема алгоритма обработки матриц;
- схема алгоритма вывода результатов расчета.

Схемы алгоритмов выполняются в соответствии с требованиями ГОСТ 19.701–90 в среде инженерной графики AutoCAD или другой, согласованной с руководителем работы. При этом наименьшая величина высоты отдельного символа процесса и данных – 10 мм, а наименьшее расстояние между блоками – 5 мм. Рекомендуется выполнять изображение всех блоков структурной схемы одинаковой высоты и ширины. При этом допускается такое отклонение в соотношениях геометрической формы изображений, которое не затрудняет определение назначения блока при чтении схемы.

В графе обозначения основной надписи листа записывается обозначение, состоящее не менее чем из двух букв, сокращенно обозначающих наименование заданного метода, и семизначного цифрового кода вида 00.00.000, заканчивающегося обозначением кода документа чертежа Д1, означающего документы прочие.

4.2 Оформление пояснительной записки

Пояснительная записка оформляется на листах бумаги формата А4. Текст пояснительной записки оформляется с соблюдением всех требований ГОСТ 2.105–95 *Общие требования к текстовым документам*. Используемые в пояснительной записке термины должны соответствовать ГОСТ ИСО/МЭК 2382–99, а все страницы должны иметь основную надпись согласно ГОСТ 2.104–68. Каждый раздел, кроме содержания, введения, заключения и списка литературных источников, должен начинаться с новой страницы с основной надписью формы 2 высотой 40 мм. Остальные страницы должны иметь основную надпись формы 2а высотой 15 мм. Обозначение документа, записываемое в основной надписи листов пояснительной записки, должно совпадать с обозначением основной надписи листа графической части и заканчиваться шифром ПЗ.

4.2.1 Оформление содержания.

Содержание пояснительной записки размещается сразу после бланка задания на курсовую работу. Оно включает наименования разделов и подразделов пояснительной записки с указанием номера страницы. Сквозная нумерация ее листов выполняется в правом верхнем углу, начиная с титульного

листа (на нем номер не ставится). Содержание фактически расположено на третьей странице пояснительной записки. При оформлении электронного варианта текста рекомендуется использовать иерархическую структуру заголовков, автоматическую расстановку номеров страниц и вставку автосодержания.

4.2.2 Оформление введения.

Введение оформляется на отдельной странице. Объем его не должен превышать четыре процента от общего состава пояснительной записки. В тексте введения описывается, к какой области относится данная работа и на основе какого документа производится ее выполнение.

4.2.3 Оформление и состав заключения.

В заключении необходимо проанализировать полученные результаты в ходе выполнения курсовой работы. Отражаются сильные и слабые стороны разработанной программы, дается рекомендация по ее дальнейшему применению и развитию. Объем заключения не должен превышать четыре процента от общего состава пояснительной записки.

4.2.4 Оформление списка используемых источников.

Оформление списка литературных источников, используемых при выполнении курсовой работы, производится в соответствии с требованиями ГОСТ 7.1–2003 *Библиографическое описание документа. Общие требования и правила составления*.

Список используемых источников составляется согласно порядку упоминания в тексте пояснительной записки, а ссылки на список источников выполняются в прямоугольных скобках.

Пояснительная записка должна помещаться в жесткую обложку для курсовой работы с титульным листом без основной надписи.

5 Расчет оценки по курсовой работе

Для расчета итоговой оценки по курсовой работе используется методика на основании Положения о проведении рейтинг-контролей. Перечень этапов выполнения курсовой работы и количества баллов за каждый из них представлен в таблице 5.1.

Максимальные баллы начисляются за выполненные досрочно (точно в срок) разделы. За выполнение необязательных подразделов начисляются дополнительные баллы. Минимальные баллы начисляются при несвоевременном выполнении разделов (без соблюдения календарных сроков выполнения курсовой работы).

По итогам защиты курсовой работы начисляются дополнительно от 15 до 40 баллов. Итоговая оценка вычисляется как сумма баллов и переводится в пятибалльную шкалу по таблице 5.2.

Таблица 5.1 – Перечень этапов курсовой работы

Этап выполнения	Минимум	Максимум
Анализ требований к разрабатываемому приложению	5	5
Описание алгоритмов приложения	5	10
Проектирование интерфейса приложения	6	10
Программирование работы приложения	7	10
Тестирование приложения	5	10
Оформление пояснительной записки	5	10
Графическая часть курсовой работы		
Схемы алгоритмов	3	5
Итого за выполнение курсовой работы	36	60
Защита курсовой работы	15	40

Таблица 5.2 – Итоговая оценка по курсовой работе

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Балл	85–100	68–84	51–67	0–50

Список литературы

1 **ГОСТ 19.701–90 (ИСО 5807–85)**. Схемы алгоритмов, программ, данных и систем. – Москва: Изд-во стандартов, 1991. – 24 с.

2 **Гвоздева, В. А.** Информатика, автоматизированные информационные технологии и системы: учебник / В. А. Гвоздева. – Москва: ФОРУМ; ИНФРА-М, 2021. – 542 с.

3 **Гагарина, Л. Г.** Технология разработки программного обеспечения: учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул; под ред. Л. Г. Гагариной. – Москва: ФОРУМ; ИНФРА-М, 2022. – 400 с.

4 **Дадян, Э. Г.** Современные технологии программирования. Язык C#: учебник: в 2 т. Т. 1: Для начинающих пользователей / Э. Г. Дадян. – Москва: ИНФРА-М, 2021. – 312 с.

5 **Хорев, П. Б.** Объектно-ориентированное программирование с примерами на C#: учебное пособие / П. Б. Хорев. – Москва: ФОРУМ; ИНФРА-М, 2020. – 200 с.