

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

Томский государственный университет систем управления и
радиоэлектроники (ТУСУР)

Кафедра радиоэлектроники и защиты информации (РЗИ)

УТВЕРЖДАЮ

Зав. кафедрой РЗИ

_____ А.С. Задорин

ИНФОРМАТИКА

Методические указания по выполнению курсовой работы
для студентов специальностей 11.03.01 «Радиотехника» и
11.03.02 «Инфокоммуникационные технологии и системы связи»

Разработчик

доцент кафедры РЗИ

_____ Д.В. Дубинин

"__" _____ 2016 г.

Дубинин Д.В. Информатика. Методические указания по выполнению курсовой работы: учебно-методическое пособие – Томск: Томский государственный университет систем управления и радиоэлектроники, 2016. – 38 с.

Приводятся методические указания по выполнению курсовой работы по дисциплине «Информатика и информационные технологии» для студентов специальностей 11.03.01 «Радиотехника» и 11.03.02 «Инфокоммуникационные технологии и системы связи».

© Дубинин Д.В., 2016.

© Томский государственный университет систем управления и радиоэлектроники, 2016.

ОГЛАВЛЕНИЕ

1. Введение.....	4
2. Задания на курсовое проектирование.....	5
3. Рекомендации по написанию программы.....	16
4. Защита курсовой работы.....	20
5. Требования к элементам пояснительной записки.....	24
6. Требования к оформлению пояснительной записки.....	28
7. Список рекомендуемой литературы.....	32
Приложение А (справочное) Форма бланка задания.....	35
Приложение Б Пример оформления титульного листа.....	36
Приложение В Пример оформления реферата.....	37
Приложение Г Пример оформления оглавления.....	38

1. Введение

Целью курсовой работы по дисциплине «Информатика и информационные технологии» является создание программного обеспечения, с помощью которого решается одна из типовых инженерных задач.

При выполнении курсовой работы решаются следующие задачи:

- развитие у студентов аналитического и алгоритмического образов мышления, понимание их особенностей и взаимосвязи;
- изучение численных методов решения задач, наиболее часто встречающихся в инженерной практике;
- создание программного продукта с использованием одного из языков программирования высоко уровня;
- приобретение навыков оформления технической документации;
- приобретение опыта публичного выступления, которое включает умение кратко и логично изложить существо вопроса, обосновать собственные принимаемые решения и аргументировано их защищать.

Курсовая работа по дисциплине «Информатика» является первой из всех курсовых работ и проектов, которую предстоит выполнить студентам. Поэтому важно с начала обучения научиться грамотно и экономно расходовать свое время, применять вузовские рекомендации и стандарты при оформлении документации.

Во второй главе приведены данные, являющиеся составной частью задания, которые записываются в соответствующие разделы типового бланка.

В третьей главе даны рекомендации по выполнению курсовой работы. Их целью является предупреждение типовых ошибок, которые допускают студенты при написании программы.

В четвертой главе излагается процедура защиты курсовой работы, приводятся критерии, по которым работа оценивается и выставляется итоговая оценка.

В пятой и шестой главах даны правила оформления пояснительной записки с учетом специфики курсовой работы по данной дисциплины. Полный перечень правил дан в [1] или в ГОСТах конструкторской и программной документаций [2-33].

В приложениях даны примеры некоторых листов пояснительной записки.

2. Задания на курсовое проектирование

Курсовая работа выполняется по индивидуальным заданиям. Задание составляется преподавателем, утверждается заведующим кафедрой и выдается студенту на первом занятии. Бланк задания приведен в приложении А.

Тематика курсовых работ связана с изучением студентами разделов высшей математики и программирования на одном из языков высокого уровня. Примерные темы курсовых работ, данные, необходимые для проектирования, и перечень вопросов, подлежащих разработке, приведены в таблице 2.1

Таблица 2.1 Примерные темы курсовых работ

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
1	Программа вычисления интеграла методом левых прямоугольников	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c;$ $y = a \cdot \sin(x + b) + c; \quad y = \frac{a}{x + b} + c.$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются из файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи.</p> <p>Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, абсолютную и относительную погрешности вычисления. Определить требования к структуре входного файла.</p>
2	Программа вычисления интеграла методом правых прямоугольников	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c;$ $y = a \cdot \cos(x + b) + c;$ $y = a \cdot \ln x + b + c.$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи.</p> <p>Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, абсолютную и относительную погрешности вычислений. Определить требования к структуре входного файла.</p>

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
3	Программа вычисления интеграла методом центральных прямоугольников	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ;$ $y = a \cdot \ln x + b + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>
4	Программа вычисления интеграла методом трапеций	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ;$ $y = a \cdot \ln x + b + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
5	Программа вычисления интеграла методом парабол	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ; y = \frac{a}{x + b} + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>
6	Программа вычисления интеграла методом Симпсона 3/8	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ; y = \frac{a}{x + b} + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
7	Программа вычисления интеграла методом Буля	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ; y = \frac{a}{x + b} + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число интервалов разбиения не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>
8	Программа вычисления интеграла методом Гаусса-Лежандра	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ; y = \frac{a}{x + b} + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Число узловых точек от 2 до 5 – задается пользователем. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
9	Программа вычисления интеграла методом Монте-Карло	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ; y = \frac{a}{x + b} + c .$ <p>Коэффициенты a, b, c – вещественные числа. Пределы интегрирования – вещественные числа, лежащие в пределах от – 1000 до 1000. Количество случайных точек не более 500. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления интеграла. Выдать на экран точное и приближенное значения интеграла, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>
10	Программа решения уравнения методом половинного деления	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ;$ $y = a \cdot \ln x + b + c .$ <p>Коэффициенты a, b, c, погрешность вычисления корня, границы интервала поиска корня – вещественные числа. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления корня. Выдать на экран точное и приближенное значения корня, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>
11	Программа решения уравнения методом хорд	<p>Число уравнений 3.</p> $y = a \cdot x^3 + b \cdot x + c ;$ $y = a \cdot \sin(x + b) + c ; y = \frac{a}{x + b} + c .$ <p>Коэффициенты a, b, c, погрешность вычисления корня, границы интервала поиска корня – вещественные числа. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления корня. Выдать на экран точное и приближенное значения корня, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.</p>

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
12	Программа решения уравнения методом Ньютона	Число уравнений 3. $y = a \cdot x^3 + b \cdot x + c$; $y = a \cdot \sin(x + b) + c$; $y = a \cdot \ln x + b + c$. Коэффициенты a, b, c , погрешность вычисления корня, начальная точка поиска корня – вещественные числа. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления корня. Выдать на экран точное и приближенное значения корня, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.
13	Программа решения уравнения модифицированным методом Ньютона	Число уравнений 3. $y = a \cdot x^3 + b \cdot x + c$; $y = a \cdot \cos(x + b) + c$; $y = \frac{a}{x + b} + c$. Коэффициенты a, b, c , погрешность вычисления корня, начальная точка поиска корня – вещественные числа. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления корня. Выдать на экран точное и приближенное значения корня, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.
14	Программа решения уравнения методом итераций	Число уравнений 3. $y = a \cdot x^3 + b \cdot x + c$; $y = a \cdot \cos(x + b) + c$; $y = a \cdot \ln x + b + c$. Коэффициенты a, b, c , погрешность вычисления корня, начальная точка поиска корня – вещественные числа. Все входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования процесс вычисления корня. Выдать на экран точное и приближенное значения корня, относительную и абсолютную погрешности вычислений. Определить требования к структуре входного файла.

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
15	Программа интерполяции результатов измерений с помощью полинома Лагранжа	Результаты измерений являются вещественными числами, которые хранятся в файле или вводятся с клавиатуры. Количество результатов измерений не более 10. Все параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования результаты измерений и интерполяционную функцию. Отобразить на экране уравнение интерполяционной функции. Определить требования к структуре входного файла.
16	Программа интерполяции результатов измерений с помощью полинома Ньютона	Результаты измерений являются вещественными числами, которые хранятся в файле или вводятся с клавиатуры. Количество результатов измерений не более 10. Все параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования результаты измерений и интерполяционную функцию. Отобразить на экране уравнение интерполяционной функции. Определить требования к структуре входного файла.
17	Программа интерполяции результатов измерений кубическим сплайном	Результаты измерений являются вещественными числами, которые хранятся в файле или вводятся с клавиатуры. Количество результатов измерений не более 10. Все параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования результаты измерений и интерполяционную функцию. Отобразить на экране уравнения кубического сплайна. Определить требования к структуре входного файла.

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
18	Программа интерполяции результатов измерений параболическим сплайном	Результаты измерений являются вещественными числами, которые хранятся в файле или вводятся с клавиатуры. Количество результатов измерений не более 10. Все параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования результаты измерений и интерполяционную функцию. Отобразить на экране уравнения параболического сплайна. Определить требования к структуре входного файла.
19	Программа интерполяции результатов измерений по методу наименьших квадратов	Результаты измерений являются вещественными числами, которые хранятся в файле или вводятся с клавиатуры. Имя файла задается пользователем. Количество результатов измерений не более 100. Аппроксимирующая функция (прямая или парабола) выбирается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования результаты измерений и интерполяционную функцию. Отобразить на экране уравнение аппроксимирующей функции, абсолютное и относительное значение максимальной ошибки интерполяции. Определить требования к структуре входного файла.
20	Программа интерполяции результатов измерений тригонометрическим полиномом	Результаты измерений являются вещественными числами, которые хранятся в файле или вводятся с клавиатуры. Имя файла задается пользователем. Количество результатов измерений не более 100. Аппроксимирующая функция тригонометрический полином. Степень полинома не более 10. Степень полинома выбирается пользователем.	Составить алгоритм решения задачи. Отобразить на экране в графическом режиме с учетом масштабирования результаты измерений и интерполяционную функцию. Отобразить на экране уравнение аппроксимирующей функции, абсолютное и относительное значение максимальной ошибки интерполяции. Определить требования к структуре входного файла.

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
21	Демонстрационная программа сортировки методом «пузырек»	Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99.	Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных методом «пузырек». Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.
22	Демонстрационная программа сортировки методом «выбора»	Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении массива с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99.	Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных методом «выбора». Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.
23	Демонстрационная программа сортировки методом Шелла	Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении массива с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99. Последовательность шагов сортировки $2^k - 1$ или $2^k + 1$ определяется пользователем.	Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных методом Шелла. Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
24	Демонстрационная программа сортировки «быстрым» методом	Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении массива с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99.	Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных «быстрым» методом. Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.
25	Демонстрационная программа сортировки методом простых вставок	Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении массива с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99.	Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных методом простых вставок. Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.
26	Демонстрационная программа сортировки методом бинарных вставок	Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении массива с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99.	Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных методом бинарных вставок. Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.

№ вар-та	Тема работы	Исходные данные для проектирования	Перечень вопросов, подлежащих разработке
27	Демонстрационная программа сортировки методом подсчета	<p>Размер массива не превышает 40 и задается с клавиатуры. Заполнение массива с помощью датчика случайных чисел, из файла или с клавиатуры по выбору пользователя. Имя файла задается пользователем. Элементы массива целые неотрицательные числа. При заполнении массива с помощью датчика случайных чисел максимальное значение элементов массива задается с клавиатуры и не превышает 99.</p>	<p>Составить алгоритм решения задачи. Отобразить на экране в пошаговом или автоматическом режиме сортировку данных методом бинарных вставок. Режим отображения задается пользователем. Вывести на экран значения числа сравнений и числа перестановок.</p>
28	Научный калькулятор	<p>Количество десятичных знаков, поддерживаемых калькулятором – 60. Число выполняемых функций – 8: сложение, вычитание, деление, умножение, вычисление синуса, косинуса, тангенса, извлечение квадратного корня. Входные параметры вводятся с клавиатуры или считываются с файла по выбору пользователя. Имя файла задается пользователем.</p>	<p>Отобразить на экране входные данные и результат вычислений.</p>

3. Рекомендации по написанию программы

При написании программы необходимо выполнить следующие шаги.

- Определить задачу, которую должен выполнить компьютер.
- Определить интерфейс программы (что увидит пользователь, запустив программу).
- Разбить программу на логические шаги, направленные на получение конечного результата.
- Набрать исходный текст программы.
- Откомпилировать программу, исправив синтаксические ошибки.
- Протестировать работу программы.

При создании программы предварительно необходимо изучить теоретические основы предлагаемого метода решения поставленной задачи, выяснить его особенности, недостатки и преимущества по сравнению с аналогичными методами. При изучении технической литературы нужно помнить, что большинство авторов, написавших книги по численным методам, не являются профессиональными программистами. Следовательно, особенности, тонкости программной реализации данного численного метода, скорее всего, не будут разобраны.

Как правило, всю программу можно условно разделить на три больших части:

- ввод исходных данных,
- проведение вычислений,
- вывод результатов вычислений на экран.

При проектировании решения каждой части нужно представить общий алгоритм в виде множества простейших шагов. Для каждого шага целесообразно написать подпрограмму. Например, если по заданию необходимо последовательно ввести значения трех параметров уравнения, то лучше написать одну подпрограмму, которая обеспечивает корректный ввод числа, и обратиться к ней три раза с разными параметрами, чем несколько раз дублировать строки программы. Хорошим правилом программирования является написание подпрограммы, которая выполняет только одну задачу.

Основная проблема при написании первой части программы связана с защитой от неправильного ввода. Например, требуется ввести значение переменной целого типа N , а пользователь может ввести текстовую информацию. Если оператором ввода считать с клавиатуры значение вышеуказанной переменной, возникнет ошибка, которая может привести к аварийному завершению программы.

Существует два способа реализации защиты от подобной ошибки. Продемонстрируем суть обоих способов на примере реализации языка Pascal.

Суть первого способа состоит в том, что оператором ввода *readln(MyString)* считывается строка текста, которая присваивается строковой переменной *MyString*. Затем с помощью стандартной подпрограммы языка Pascal *Val* выполняется попытка преобразования данной строки в число. Один из параметров подпрограммы *Val* принимает значение, которое зависит от исхода преобразования. Если такое преобразование прошло успешно, переменной *N* присваивается результат преобразования. Если же возникла ошибка, программа должна сообщить пользователю об ошибке, вывести соответствующее предупреждение, в котором попросит повторно ввести целое число.

Второй способ реализации защиты более сложный. Информация считывается с клавиатуры посимвольно стандартной функцией *readkey*. Данная функция определяет ASCII код нажатой клавиши. Если считанный код совпадает с кодами «разрешенных» клавиш (например, при вводе натурального числа – это цифры от 0 до 9, ASCII коды которых имеют значения от 48 до 57), то он обрабатывается, и на экране появляется соответствующий символ. Нажатия на остальные, «запрещенные» клавиши игнорируются, и на экране изменений не происходит. Считывание данных заканчивается после нажатия клавиши *Enter*.

При проведении вычислений, реализующих заданный численный метод, следует не допускать ситуаций приводящих к аварийному завершению программ. Например, при вычислении интеграла методом трапеций требуется определить значения функции в узловых точках. Если подынтегральная функция имеет разрыв, то при определенных значениях параметров функции, пределов интегрирования и числе разбиений может сложиться ситуация, когда узловая точка совпадет с точкой разрыва. В этом случае необходимо вывести сообщение, говорящее, что при введенных значениях входных данных программа не сможет решить поставленную задачу указанным методом. Такое завершение аварийным не считается.

При написании третьей части программы обычно трудности возникают при реализации масштабирования. В графическом режиме для рисования различных объектов (точек, прямых, прямоугольников, окружностей) используются целочисленные координаты, связанные с системой координат экрана монитора. Например, для рисования на экране линии с помощью подпрограммы *line* необходимо задать координаты начальной и конечной точек. Следует помнить, что началом координат является точка верхнего левого угла экрана. Ее координаты (0;0). Ось абсцисс направлена слева направо. Ось

ординат направлена сверху вниз. Максимальное разрешение экрана в пикселях можно определить, используя стандартные функции *getmaxx* и *getmaxy*.

Результаты вычислений (интеграла, значения функции) принимают вещественные (дробные) значения. Для графического отображения результатов вычислений на экране монитора требуется преобразовать вещественное значение в целое. А для того, чтобы выполнить требование «...с учетом масштабирования...», нужно выполнить это преобразование пропорционально имеющимся вещественным значениям. Например, при вычислении определенного интеграла пользователь может ввести пределы интегрирования в первом случае от -100 до 200, а во втором от -0,01 до 0,02. Необходимо, чтобы в обоих случаях график функции отображался в соответствующих пределах.

После написания исходного текста программы выполняется ее компиляция или получение исполняемого кода. Если в тексте программы есть синтаксические ошибки, компилятор языка программирования укажет на них. После исправления всех синтаксических ошибок и получения исполняемого кода программы необходимо протестировать правильность работы программы.

Тестированием называется процесс поиска ошибок в программе. Тестирование программы является важнейшим этапом ее создания. Из всего времени выполнения курсовой работы на долю тестирования должно приходиться порядка 40 процентов.

Распространенным заблуждением является утверждение, что цель тестирования – возможность убедиться в правильности работы программы. Это приводит к сокращению числа тестов. Тесты не содержат подвохов и описывают обычно те ситуации, которые рассматривались при написании программы. Такое тестирование характерно для начинающих программистов, которые не допускают мысли о непредусмотренных ситуациях.

В действительности тестирование – это поиск ошибок. Качественное выполнение этой работы требует изменение психологического подхода. Цель тестирующего – «сломать» программу, доказать ее неработоспособность. В этом случае дело не ограничивается простейшими тестами. В ход идут самые каверзные входные данные, самые неожиданные их сочетания. Удачным считается тест, который обнаружил ошибку. Тестирование программы такой же творческий процесс, как ее написание.

Все методы тестирования можно разделить на две группы: тестирование по принципу черного ящика и по принципу белого ящика.

Черным ящиком обычно называют объект, который способен реагировать на внешние воздействия, но его внутренняя структура не известна. Тестирование по методу черного ящика основана на анализе задачи, которую программа должна решать. Изучая

условия задачи, определяют возможные ситуации, формируют для каждой из них пример входных данных и обязательно находят правильный ответ. Полученный набор тестов применим к любой программе, которая решает данную задачу. Сравнивая выходные данные тестируемой программы с эталоном, определяют правильность ее работы. Тестирование по принципу черного ящика применяется на олимпиадах по программированию.

Тестирование по методу белого ящика основано на тексте конкретной программы, в которой выделяются отдельные блоки. Тесты составляются так, чтобы были задействованы все блоки программы, все ветви программы.

Самое трудное в тестирование – это составить набор тестов. Несколько советов по составлению тестов:

- тесты составляются на основе условий задачи, они не должны учитывать особенности конкретной программы;
- если задача разбивается на разные случаи, необходимы тесты, проверяющие их все;
- если на входные значения накладываются ограничения, требуется протестировать работу программы при предельных и близких к ним значениях;
- любой удачный тест должен быть запомнен и включен в тестовый пакет данной программы для повторного тестирования;
- в тесты должны включаться не только допустимые в задаче значения, но и запрещенные.

4. Защита курсовой работы

После написания программы оформляется пояснительная записка, которая сдается преподавателю на проверку. Пояснительная записка должна быть скреплена или переплетена.

Преподаватель проверяет работу и на обороте титульного листа записывает свои замечания. Если в работе отсутствуют серьезные недостатки, на титульном листе пишется заключение: «После доработки к защите». Это означает, что после устранения замечаний данную работу можно защищать. В противном случае, при наличии грубых ошибок, пояснительная записка сдается на проверку вторично.

Процедура защита является важным образовательным элементом выполнения курсовой работы. Начиная с первого курса, студент должен учиться аргументировано защищать принимаемые решения. Можно получить низкую отметку, имея идеально работающую программу. И наоборот, получить высокую отметку, написав программу, имеющую определенные недостатки.

Отметка за выполнение курсовой работы выставляется на основании семи показателей:

- сложность задания на проектирование;
- качество оформления пояснительной записки;
- доклад;
- качество интерфейса программы;
- степень соответствия разработанной программы выданному заданию;
- стиль программирования;
- ответы на вопросы.

Сложность типового задания, варианты которого приведены в таблице 2.1, оценивается в четыре балла. Если студент получает нетривиальное задание, выполнение которого требует нестандартного решения, применения нетрадиционных алгоритмов, поиска и изучения дополнительной литературы, его сложность оценивается в пять баллов.

Оценка за второй показатель выставляется за степень соответствия пояснительной записки соответствующим ГОСТам, образовательному стандарту вуза. В главах 5 и 6 данного пособия приведены те правила, которые наиболее часто используются при оформлении пояснительной записки. Полностью образовательный стандарт университета приведен в работе [1].

Важным элементом защиты является доклад. На него дается пять минут. Не стоит тратить много времени, рассказывая об общих вопросах теории. Уверенный, логически

связанный доклад задает тон защите, показывает профессиональный уровень разработчика. Делая доклад, не забывайте подкреплять свои слова демонстрацией работы программы. Следует продумать значения входных параметров программы, которые позволят наиболее эффектно преподнести написанный программный продукт во время доклада.

Создавая программный продукт, разработчик должен помнить: работать с программой будет другой человек – пользователь. Как правило, пользователи являются менее квалифицированными специалистами, чем разработчики. Инструкции или руководства они не читают. Возможности находиться рядом с пользователем, при каждом запуске программы, у разработчика нет. Следовательно, программа должна обладать дружественным, интуитивно понятным интерфейсом. Чем больше возможностей предоставляется пользователю программы, тем выше оценка разработчика по данному критерию. Например, при вводе неправильных данных программа не завершает работу, а предлагает пользователю ввести данные повторно. Или выбор функции осуществляется с помощью клавиш управления курсором, а не вводом числового значения. Реализация в программе помощи повышает оценку по данному показателю.

На бланке задания в разделе 5 указаны требования, предъявляемые к программе. При выполнении всех пунктов раздела 5 по данному критерию ставиться оценка пять. Если программа соответствует не всем пунктам требований, оценка снижается.

Из всех указанных пунктов труднее всего реализовать пункт 5.2. А именно, обеспечить безаварийное завершение программы. Как правило, причины этого кроются в недостаточном тестировании программы. К типовым ошибкам, вызывающим аварийное завершение программы, можно отнести:

- не реализована защита от некорректного ввода (например, требуется ввести число, а пользователь ввел текстовую информацию);
- попытка считать данные из файла, которого нет в данном каталоге;
- во время работы программы проводится деление на ноль или извлечение квадратного корня из отрицательного значения выражения (например, программа пытается получить значение функции в точке разрыва);
- ошибка инициализации графического режима (не найден соответствующий драйвер).

Трудно дать однозначное определение, что такое хороший стиль программирования. Стиль программирования – это набор правил, который используется при написании кода программы. Существует несколько таких стандартов: стандарт кодирования K&R (по имени авторов описания языка Си Kernighan и Ritchie), стандарт

Microsoft (стиль MFC), стандарт GNU и др. При написании программы следует остановиться на каком-то одном стандарте и последовательно его придерживаться.

Приведу несколько примеров что должно, и чего не должно быть в программе, написанной в хорошем стиле.

- Все идентификаторы (константы, типы, переменные, функции, процедуры, модули) должны иметь осмысленные имена. Например, для хранения наибольшего значения элементов массива предпочтительней использовать переменную `max`, чем `a2s`.
- Имена функций должны отражать то, что эти функции делают и что они возвращают.
- Необходимо минимизировать использование глобальных переменных, а лучше вообще отказаться от них.
- Имена указателей должны начинаться с символа “p” (от англ. pointer – указатель).
- Имена ссылок должны начинаться с символа “r” (от англ. reference – ссылка).
- В тексте программы не должно быть чисел. Все численные значения либо описываются как константы, либо считываются из файла или с клавиатуры. Исключения составляют такие числа, которые в данном месте программы принципиально не могут иметь другого значения. Например, ноль (при вычислении суммы ряда необходимо предварительно обнулить соответствующую переменную), единица (цикл начинается от 1 до некоторого конечного значения).
- На одной строке не должно находиться более одного оператора. Максимальная длина строки не должна превышать 70 символов. Если вызов функции, инициализация переменных занимает более одной строки, перенос на следующую строку должен следовать после запятой.
- В программе не должно быть оператора `goto`. Использование оператора `goto` в ранних языках программирования связано с тем, что тогда не было таких гибких конструкций, которые есть в современных языках. В 1969 году Эдсгер Дейкстра доказал, что для записи любого алгоритма достаточно основных конструкций: последовательность операторов, альтернатива (выбор), повторение (цикл). Использование оператора `goto` приводит к тому, что исполняемый код становится медленней и объемней. При написании больших программ в ряде случаев применение оператора `goto` может быть оправданно. Однако программа данной курсовой работы к числу таких программ не относится. Использование оператора безусловного перехода будет рассматриваться преподавателем, как нежелание искать более совершенный алгоритм, и приведет к снижению соответствующей оценки.
- При работе с внешними файлами категорически запрещается использовать абсолютные имена путей. Например, “`c:\lang\bc31\bgi\egavga.bgi`”. Это может сделать

невозможным выполнение пункта 5.6 задания. За «точку отсчета» лучше выбрать текущий каталог. Следует помнить, что некоторые старые системы программирования работают с файлами в спецификации 8.3 и не поддерживают длинные имена.

- Текст программы должен быть снабжен комментариями. Комментарии дают информацию о том, как работает программа, и не влияют на выполнение программы. Большинство специалистов считают, что комментарии должны объяснять намерения программиста. Бессмысленно ставить комментарии в каждой строке программы. Более того, комментарии в каждой строке программы косвенно указывают на то, что ее автором является кто-то другой.

После окончания доклада студенту задаются вопросы. В отличие от теоретического экзамена вопросы задаются по теме курсовой работы, методу и алгоритму решения задачи и тексту программы. Студент должен понимать и уметь объяснить необходимость и работу каждого оператора своей программы. Правильность и полнота ответов определяют оценку по наиболее важному критерию.

Итоговая отметка, которая проставляется в зачетную книжку и ведомость, определяется путем усреднения семи оценок, но не может быть выше, чем оценка, выставляемая за ответы на вопросы.

5. Требования к элементам пояснительной записки

Согласно [1] курсовая работа по информатике состоит из текстового документа и графического материала. Текстовый документ именуется «Пояснительная записка к курсовой работе по информатике» и должен содержать следующие разделы:

- титульный лист;
- реферат (аннотация);
- задание;
- оглавление;
- введение;
- основную часть;
- заключение;
- список использованных источников;
- приложения.

Текст пояснительной записки должен быть выполнен на русском языке.

К графическому материалу относят демонстрационные материалы (иллюстрации), чертежи и схемы, которые позволяют более наглядно отобразить суть курсовой работы при ее публичной защите.

5.1 Титульный лист

Пример оформления титульного листа приведен в приложении Б данного пособия.

5.2 Реферат

Реферат размещается на втором листе пояснительно записки. Заголовком этого листа служит слово «Реферат», записанное симметрично тексту. Реферат должен содержать:

- сведения о количестве листов, количестве иллюстраций, таблиц, использованных источников, приложений;
- перечень ключевых слов;
- текст реферата.

Перечень ключевых слов состоит из 5 – 15 слов или словосочетаний, которые в наибольшей мере характеризуют содержание текста пояснительной записки. Все слова или словосочетания пишутся прописными буквами в именительном падеже через запятые.

Текст реферата должен отражать

- объект исследования или разработки;
- цель работы;
- метод исследования и название программного обеспечения, которое использовалось для разработки программы;
- полученные результаты;
- основные характеристики программы (названия файлов, их размер, требования программы к аппаратному и программному обеспечению);
- степень внедрения;
- рекомендации по внедрению;
- область применения;
- эффективность и значимость работы;
- предположения о развитии объекта исследования;
- дополнительные сведения (особенности выполнения и оформления работы).

Если текст пояснительной записки не содержит сведений по какой-либо из перечисленных структурных частей реферата, то в тексте реферата она опускается, при этом последовательность изложения сохраняется.

Пример составления реферата приведен в приложении В данного пособия.

5.3 Задание

Задание на курсовую работу выдается на типовом бланке. Задание подписывается студентом, преподавателем и утверждается заведующим кафедрой. После утверждения задания заведующим кафедрой вносить в него исправления и дополнения запрещается. Защищать курсовую работу, в которой нет листа задания, запрещается.

Пример типового бланка задания на курсовую работу дан в приложении А данного пособия.

5.4 Оглавление

Оглавление должно отражать все материалы, представленные в пояснительной записке. Слово "Оглавление" записывают в виде заголовка, симметрично тексту, с прописной буквы. В оглавлении перечисляются заголовки разделов, подразделов, список литературы, каждое приложение и указывают номера листов, на которых они начинаются.

Материалы, представляемые на технических носителях данных ЭВМ, должны быть перечислены в содержании с указанием вида носителя (дискета или компакт диск), обозначения и наименования документов, имен и форматов соответствующих файлов, а также места расположения носителя в пояснительной записке.

Если при публичной защите используется графический материал, в конце содержания его перечисляют с указанием: «На отдельных листах».

Пример оформления оглавления приведен в приложении Г данного пособия.

5.5 Введение

В разделе «Введение» указывают цель работы, область применения разрабатываемой программы, и техническое значение. Заголовок «Введение» записывают с абзаца с прописной буквы.

5.6 Основная часть

Содержание основной части работы должно отвечать техническому заданию на проектирование и требованиям, изложенным в методических указаниях соответствующей кафедры.

5.7 Заключение

Заголовок «Заключение» записывается с абзаца с прописной буквы. Заключение должно содержать краткие выводы по результатам работы, степень соответствия написанной программы полученному заданию, характеристики программы (например, количество и названия файлов, их размер, минимальную и рекомендуемую конфигурацию аппаратных и программных средств, необходимых для работы программы), рекомендации по использованию программы.

5.8 Список использованных источников

Заголовок «Список использованных источников» записывается симметрично тексту. В список включают только те источники, на которые имеются ссылки в тексте пояснительной записки. Источники нумеруются в порядке их упоминания в тексте пояснительной записки арабскими цифрами.

5.9 Приложения

В приложения рекомендуется включать материалы иллюстративного и вспомогательного характера. Как правило, в приложения включаются:

- блок-схемы алгоритмов;
- листинги программ;
- протоколы испытаний программы;
- акты внедрения.

На все приложения в тексте пояснительной записки должны быть даны ссылки. Приложения располагают и обозначают в порядке ссылок на них в тексте пояснительной записки. Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, И, О, Ч, Ъ, Ы, Ь. Например: «Приложение Б».

Каждое приложение пояснительной записки следует начинать с нового листа с указанием наверху посередине страницы слова «Приложение» и его обозначения, а под ним в скобках – «обязательное» (если его выполнения предусмотрено заданием) или «справочное».

Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

6. Требования к оформлению пояснительной записки

6.1 Общие требования

Текст пояснительной записки должен быть выполнен на белой бумаге формата А4 по ГОСТ 2.301 (210x297 мм) с одной стороны листа.

Рекомендуемые значения:

- Размеры полей: левое – 30 мм, правое – 10 мм, верхнее – 15 мм, нижнее – 20 мм.
- Абзацы в тексте начинают отступом 15 мм.
- Шрифт – Times New Roman 12 размера. Цвет – черный.
- Межстрочный интервал – полуторный.

Опечатки, описки, графические неточности, обнаруженные в процессе выполнения пояснительной записки, допускается исправлять аккуратным заклеиванием или закрасиванием белой краской и нанесением на том же месте и тем же способом исправленного текста. Повреждение листов пояснительной записки, помарки и следы не полностью удалённого текста не допускаются.

Иллюстрации, таблицы и распечатки с ЭВМ допускается выполнять на листах формата А3 ГОСТ 2.301, при этом они должны быть сложены на формат А4 "гармоникой" по ГОСТ 2.501.

Пояснительная записка должна быть сшита (переплетена) и иметь обложку. Обложку рекомендуется выполнять из плотной бумаги, совмещающая ее с титульным листом.

6.2 Требования к тексту

В тексте пояснительной записки должны применяться термины, обозначения и определения, установленные стандартами по соответствующему направлению науки, техники и технологии, а при их отсутствии – общепринятые в научно-технической литературе.

В тексте пояснительной записки не допускается:

- применять для одного и того же понятия различные научно-технические термины, близкие по смыслу (синонимы), а также иностранные слова и термины при наличии равнозначных слов и терминов в русском языке;
- применять произвольные словообразования;
- применять индексы стандартов (ГОСТ, ГОСТ Р, ОСТ и т.п.), технических условий (ТУ) и других документов без регистрационного номера.

- использовать в тексте математические знаки и знак \varnothing (диаметр), а также знаки № (номер) и % (процент) без числовых значений.

6.3 Деление текста

Текст пояснительной записки разделяют на разделы, подразделы, пункты. Пункты, при необходимости, могут быть разделены на подпункты.

Каждый раздел рекомендуется начинать с нового листа.

Разделы должны иметь порядковые номера, обозначенные арабскими цифрами и записанные с абзацного отступа. Подразделы и пункты должны иметь нумерацию в пределах каждого раздела или подраздела, подпункты – в пределах пункта. Отдельные разделы могут не иметь подразделов и состоят непосредственно из пунктов.

Если раздел или подраздел состоит из одного пункта, этот пункт также нумеруется.

Точка в конце номеров разделов, подразделов, пунктов, подпунктов не ставится.

6.4 Заголовки

Разделы, подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов, подразделов, пунктов.

Заголовки следует выполнять с абзацного отступа с прописной буквы без точки в конце, не подчеркивая. В начале заголовка помещают номер соответствующего раздела, подраздела, пункта.

Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Расстояние между заголовком и текстом должно быть равно удвоенному межстрочному расстоянию; между заголовками раздела и подраздела – одному межстрочному расстоянию.

6.5 Таблицы

Таблицы применяют для лучшей наглядности и удобства сравнения показателей.

Таблицы слева, справа и снизу, как правило, ограничивают линиями. Заголовок таблицы должен быть отделен линией от остальной части таблицы. Высота строк таблицы должна быть не менее 8 мм.

Все таблицы нумеруют в пределах раздела арабскими цифрами.

Над левым верхним углом таблицы помещают надпись «Таблица» с указанием номера таблицы, например: «Таблица 2.1» (первая таблица второго раздела), «Таблица В.5» (пятая таблица приложения В).

Таблица может иметь название. Название таблицы должно отражать содержание, быть точным, кратким. Если таблица имеет название, то его помещают после номера таблицы через тире, с прописной буквы.

На все таблицы должны быть ссылки в тексте пояснительной записки. Таблицу следует располагать непосредственно после абзаца, где она упоминается впервые, или на следующем листе.

6.6 Иллюстрации

Иллюстрации помещаются для пояснения текста и должны быть выполнены в соответствии с требованиями государственных стандартов.

Иллюстрации следует выполнять на бумаге или пленке того же формата, что и текст, с соблюдением тех же полей, что и для текста. Допускается наклеивание отдельно выполненных изображений на форматный лист. Цвет изображений, как правило, черный на белом фоне.

В тексте пояснительной записки все иллюстрации (фотографии, схемы, чертежи и пр.) именуется рисунками.

Рисунки нумеруются в пределах раздела (приложения) арабскими цифрами, например: «Рисунок 1.2» (второй рисунок первого раздела); «Рисунок В.3» (третий рисунок приложения В).

Рисунок может иметь тематическое наименование и пояснительные данные (подрисуночный текст).

Слово «рисунок», его номер и тематическое наименование (при наличии) помещают ниже изображения и пояснительных данных симметрично иллюстрации.

6.7 Формулы

Формулы следует выделять из текста в отдельную строку.

Значения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Значение каждого символа дают с новой строки в той последовательности, в какой они приведены в формуле. Первая строка расшифровки должна начинаться со слова «где» без двоеточия после него.

6.8 Ссылки

В тексте пояснительной записки приводят ссылки на данную работу или на использованные источники.

При ссылках на данную работу указывают номера структурных частей текста, формул, таблиц, рисунков, обозначения чертежей и схем, а при необходимости – также графы и строки таблиц и позиции составных частей изделия на рисунке, чертеже или схеме.

При ссылках на структурные части текста пояснительной записки указывают номера разделов (со словом «раздел»), приложений (со словом «приложение»), подразделов, пунктов, подпунктов, перечислений, например: «...в соответствии с разделом 2», «... согласно 3.1», «... по 3.1.1», «... в соответствии с 4.2.2, перечисление б»; (приложение Л); «... как указано в приложении М».

Ссылки в тексте на номер формулы дают в скобках, например: «...согласно формуле (В.1)»; «...как следует из выражения (2.5)».

Ссылки в тексте на таблицы и иллюстрации оформляют по типу: (таблица 4.3); «... в таблице 1.1, графа 4»; (рисунок 2.11); «...в соответствии с рисунком 1.2»; «... как показано на рисунке Г.7, поз. 12 и 13».

6.8.2 При ссылке в тексте на использованные источники следует приводить порядковые номера по списку использованных источников, заключенные в квадратные скобки, например: «... как указано в монографии [10]»; «... в работах [11, 12, 15-17]».

При необходимости в дополнение к номеру источника указывают номер его раздела, подраздела, страницы, иллюстрации, таблицы, например: [12, раздел 2]; [18, подраздел 1.3, приложение А]; [19, с.25, таблица 8.3].

6.9 Сокращения

При многократном упоминании устойчивых словосочетаний в тексте ТД следует использовать аббревиатуры или сокращения.

При первом упоминании должно быть приведено полное название с указанием в скобках сокращенного названия или аббревиатуры, например: «операционная система (ОС)», а при последующих упоминаниях следует употреблять сокращенное название или аббревиатуру.

Расшифровку аббревиатур и сокращений, установленных государственными стандартами (ГОСТ 2.316, ГОСТ 7.12) и правилами русской орфографии, допускается не приводить, например: ЭВМ, НИИ, АСУ, с. (страница), т.е. (то есть), вуз (высшее учебное заведение) и др.

7. Список рекомендуемой литературы

1. ОС ТУСУР 01-2013 «Работы студенческие по направлениям подготовки и специальностям технического профиля». Томск, 2013.
2. ГОСТ 2.104-68 ЕСКД. Основные надписи.
3. ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам.
4. ГОСТ 2.106-68 ЕСКД. Текстовые документы.
5. ГОСТ 2.108-68 ЕСКД. Спецификация.
6. ГОСТ 2.109-73 ЕСКД. Основные требования к чертежам.
7. ГОСТ 2.119-73 ЕСКД. Эскизный проект.
8. ГОСТ 2.201-80 ЕСКД. Обозначение изделий и конструкторских документов.
9. ГОСТ 2.301-68 ЕСКД. Форматы.
10. ГОСТ 2.304-81 ЕСКД, Шрифты чертежные.
11. ГОСТ 2.316-68 ЕСКД. Правила нанесения на чертежах надписей, технических требований и таблиц.
12. ГОСТ 2.501-88 ЕСКД. Правила учета и хранения.
13. ГОСТ 2.701-84 ЕСКД. Схемы. Виды и типы. Общие требования к выполнению.
14. ГОСТ 2.702-75 ЕСКД. Правила выполнения электрических схем.
15. ГОСТ 2.710-81 ЕСКД. Обозначения буквенно-цифровые в электрических схемах.
16. ГОСТ 3.1201-85 ЕСТД. Система обозначения технологической документации.
17. ГОСТ 7.1-84 Система стандартов по информации, библиотечному и издательскому делу (ССИБИД) Библиографическое описание документа. Общие требования и правила составления.
18. ГОСТ 7.9-95 ССИБИД Реферат и аннотация.
19. ГОСТ 7.12-93 ССИБИД Сокращения русских слов и словосочетаний в библиографическом описании произведений печати.
20. ГОСТ 7.32-91 ССИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления.
21. ГОСТ 8.417-81 Государственная система обеспечения единства измерений. Единицы физических величин.
22. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
23. ГОСТ 19.103-77 ЕСПД. Обозначения программ и программных документов.
24. ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.

25. ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
26. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.
27. ГОСТ 24.301-80 Система технической документации на АСУ. Общие требования к текстовым документам.
28. ГОСТ 24.302-80 Система технической документации на АСУ. Общие требования к выполнению схем.
29. ГОСТ 24.303-80 Система технической документации на АСУ. Обозначения условные графические технических средств.
30. ГОСТ 34.201-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
31. ГОСТ 25346-89 Основные нормы взаимозаменяемости. ЕСДП. Общие положения, ряды допусков и основных отклонений.
32. ГОСТ 28388-89 Система обработки информации. Документы на магнитных носителях данных. Порядок выполнения и обращения.
33. Р 50-77-88 Рекомендации. ЕСКД. Правила выполнения диаграмм.
34. Информатика. Базовый курс: Учебное пособие для студентов вузов/ Сергей Витальевич Симонович [и др.]; ред. С. В. Симонович. - СПб.: Питер, 2001. - 640 с.: ил.
35. Информатика: Учебник для вузов/ Владислав Алексеевич Острейковский. - М.: Высшая школа, 2001. - 512 с.: ил. - ISBN 5-06-003533-6.
36. Фигурнов В.Э. IBM PC для пользователя. Издание 7-е, переработанное и дополненное. - М.: ИНФРА - М, 2000. - 640 с.
37. Информатика: Под ред. Н. В. Макаровой. - 3-е изд., перераб.. - М.: Финансы и статистика, 2002. - 768 с.: ил. - ISBN 5-279-02202-0
38. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль. - Томск: МП "РАСКО", 1991. - 272 с.
39. Мак-Кракен Д., Дорн У. Численные методы и программирование на ФОРТРАНЕ. - М.: Мир, 1977. - 584 с.
40. Мэтьюз, Джон, Г., Финк, Куртис, Д. Численные методы. Использование MATLAB, 3-е издание.: Пер. с англ. - М.: Издательский дом «Вильямс», 2001. - 720с.: ил.
41. Калиткин Н.Н. Численные методы. - М.: Наука, 1978. - 400 с.
42. Турчак Л.И. Основы численных методов. - М.: Наука, 1987. - 320 с.

43. Кнут, Дональд Эрвин. Искусство программирования для ЭВМ. В трех томах: пер. с англ./ М.: Мир, 1976.
44. Поттс С., Монк Т.С. Borland C++ в примерах. Перевод с английского. Мн., ООО «Попурри», 1996, 752 с.
45. Уинер Р. Язык Турбо Си: перевод с английского. М., Мир, 1991 – 384 с.
46. Касаткин А.И. Профессиональное программирование на языке Си. В трех томах. – Мн.: Выш. Шк., 1993.
45. Рейсдорф К., Хендерсон К. Borland C++ Builder. Освой самостоятельно. М., «Издательство БИНОМ», 1998 г. – 704 с.
46. Зуев Е.А. Программирование на языке Турбо-Паскаль 6.0., 7.0 - М.: Веста, Радио и связь, 1993. - 384 с.
47. Офицеров Д.В., Старых В.А. Программирование в интегрированной среде Турбо-Паскаль. - Минск: Беларусь, 1992. - 240 с.
48. Джонс Ж., Харроу К. Решение задач в системе Турбо-Паскаль. - М.: Финансы и статистика, 1991. – 520 с.
49. Белецкий Я. Турбо Паскаль с графикой для персональных компьютеров. М.: Машиностроение, 1991. - 320 с.
50. Васюкова Н.Д., Тюляева В.В. Практикум по основам программирования. Язык Паскаль. - М.: Высшая школа, 1991. - 215 с.
51. Святозарова Г.И., Сигитов Е.В., Козловский А.В. Практикум по программированию на алгоритмических языках. - М.: Наука, 1980. - 318 с.
52. Дагене В.А., Григас Г.К., Аугутис К.Ф. 100 задач по программированию: Кн. для учащихся: Пер. с литовского. - М., Просвещение, 1993. – 255 с.
53. Степанов Е.О., Чириков С.В. Стиль программирования на C++. Учебное пособие. – СПб.: СПб ГИТМО(ТУ), 2001. – 48 с.

Приложение А
(справочное)
Форма бланка задания

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра радиоэлектроники и защиты информации (РЗИ)

«УТВЕРЖДАЮ»
Заведующий кафедрой РЗИ
_____ А.С. Задорин
«__» _____ 201__ г.

ЗАДАНИЕ

на курсовую работу по курсу «Информатика и информационные технологии»

студенту группы _____

1. Тема работы: _____
 2. Срок сдачи работы: «__» _____ 201__ г.
 3. Исходные данные: _____
-
4. Перечень вопросов подлежащих разработке: _____
-
5. Требования к программе
 - 5.1. Все надписи в программе должны выводиться на одном языке
 - 5.2. Программа должна осуществлять корректную обработку вводимой информации, неправильных действий пользователя. Программа не должна завершаться аварийно. В случае возникновения аварийной ситуации должно выдаваться соответствующее сообщение.
 - 5.3. Использование проекта (программа должна состоять из нескольких файлов)
 - 5.4. Часто используемые константы должны быть вынесены в отдельный файл(ы)
 - 5.5. Программа должна содержать комментарии, поясняющие используемые идентификаторы, подпрограммы и алгоритмы.
 - 5.6. Программа должна сохранять работоспособность при переписывании в любой каталог.
 - 5.7. Программа должна иметь встроенную справку (помощь).
 6. Требования к отчету

Пояснительная записка по работе оформляется в соответствии с нормами ЕСПД, вузовским стандартом и должен содержать:

 - 6.1. Цель работы
 - 6.2. Описание метода решения задачи, программы и используемых алгоритмов
 - 6.3. Описание методики тестирования программы
 - 6.4. Руководство пользователя по работе с программой
 - 6.5. Блок-схему программы
 - 6.6. Распечатку текста программы
 - 6.7. Выводы по работы
- Задание выдал: _____
- _____ «__» _____ 201__ г.
- Принял к выполнению: _____
- _____ «__» _____ 201__ г.

Приложение Б

(справочное)

Пример оформления титульного листа

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего
образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиоэлектроники и защиты информации (РЗИ)

ПРОГРАММА ВЫЧИСЛЕНИЯ ИНТЕГРАЛА МЕТОДОМ МОНТЕ-КАРЛО

Пояснительная записка к курсовой работе

Студент группы 1А1

_____ С.В. Лопарев

Руководитель

Доцент кафедры РЗИ, к.т.н.

_____ Д.В. Дубинин

Приложение В

(справочное)

Пример оформления реферата

РЕФЕРАТ

Курсовая работа 37 с., 12 рис., 4 табл., 5 источников, 2 приложения.

СЛУЧАЙНЫЕ ВЕЛИЧИНЫ, СТАТИСТИЧЕСКИЕ ХАРАКТЕРИСТИКИ, КОРРЕЛЯЦИЯ, НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ, ИНТЕРФЕЙС.

Объектом исследования являются датчики случайных чисел, формирующие последовательность случайных величин, распределенных по нормальному закону распределения.

Цель работы – разработка программы генерации последовательности случайных величин, распределенных по нормальному закону, и оценки ее статистических характеристик: математического ожидания, дисперсии, функции и плотности распределения.

В процессе работы проводились экспериментальные исследования различных датчиков случайных величин, оценивались их статистические характеристики.

Разработанная программа позволяет получить последовательность случайных величин, распределенных по нормальному закону, со статистическими характеристиками задаваемыми пользователем: математическим ожиданием, дисперсией. Случайные величины записываются в файл. Имя файла и длина выборки задается пользователем.

Результаты работы могут использоваться при проведении лабораторных работ по дисциплине «Радиоавтоматика».

Программное обеспечение разработано в среде Borland C++ 3.1. Пояснительная записка выполнена в текстовом редакторе Microsoft Word 2003.

Приложение Г (справочное)

Пример оформления оглавления

Оглавление

1 Введение.....	5
2 Описание метода решения задачи.....	6
3 Описание программы и используемы алгоритмов.....	10
3.1 Описание типов и глобальных переменных.....	10
3.2 Описание структуры программы.....	12
3.3 Описание функций и их параметров	15
3.4 Защита программы от неправильного ввода данных	20
3.5 Вывод данных в графическом режиме	25
4 Описание методики тестирования программы	29
5 Руководство пользователя	33
6 Заключение	39
Список использованных источников	40
Приложение А Блок-схема программы	45
Приложение Б Листинг программы	50