

*Федеральное государственное бюджетное образовательное учреждение высшего образования*



**«Московский государственный технический университет  
имени Н.Э. Баумана»  
(национальный исследовательский университет)**

---

Факультет «Информатика и системы управления»

Кафедра «Компьютерные системы и сети»

**Г. С. Иванова, Т. Н. Ничушкина, Е.К. Пугачев, Р.С.Самарев**

## **Методические указания по выполнению курсовой работы**

**по дисциплине «Технология разработки программных систем»**

Москва  
Кафедра «Компьютерные системы и сети»  
2018

УДК 004.021

*Рецензент:*

**Г.С. Иванова, Т.Н. Ничушкина, Е.К. Пугачев, Р.С. Самарев**

Методические указания по выполнению курсовой работы по дисциплине «Технология разработки программных систем» - М.: МГТУ им. Н.Э. Баумана, 2018- 38 с.

Издание содержит: указания и рекомендации по выполнению и оформлению курсовой работы по дисциплине «Технология разработки программных систем», предусмотренной учебным планом кафедры ИУ-6 МГТУ им. Н.Э.Баумана.

Для студентов МГТУ имени Н.Э. Баумана направления «Информатика и вычислительная техника».

**Иванова Галина Сергеевна  
Ничушкина Татьяна Николаевна  
Пугачев Евгений Константинович  
Самарев Роман Станиславович**

Методические указания по выполнению курсовой работы  
по дисциплине «Технология разработки программных систем»

© 2018 МГТУ имени Н.Э. Баумана

## Оглавление

Предисловие.....	4
Введение.....	4
1 Организация проведения курсовой работы.....	5
1.1 Темы курсовых работ.....	5
1.2 Порядок выполнения курсовой работы.....	6
1.3 Сроки выполнения отдельных этапов.....	8
1.4 Порядок защиты курсовой работы.....	10
2 Содержание расчетно-пояснительной записки.....	11
2.1 Реферат.....	13
2.2 Введение.....	14
2.3 Анализ технических требований и уточнение спецификаций.....	14
2.4 Проектирование структуры и компонентов программного продукта.....	17
2.5 Подготовка тестовых данных.....	23
2.6 Заключение.....	24
3 Правила оформления пояснительной записки.....	24
3.1 Оформление текстового и графического материала.....	24
3.2 Оформление рисунков, схем алгоритмов, таблиц и формул.....	26
3.3 Оформление текстов программ.....	29
3.4 Оформление приложений.....	30
3.5 Оформление списка литературы.....	31
4 Критерии оценки курсовой работы.....	32
ПРИЛОЖЕНИЕ А. Техническое задание на курсовую работу.....	33
ПРИЛОЖЕНИЕ Б. Титульный лист расчетно-пояснительной записки.....	37
ПРИЛОЖЕНИЕ В. Пример реферата на расчетно-пояснительную записку.....	38

## Предисловие

Настоящее методическое пособие содержит указания и рекомендации по выполнению и оформлению курсовой работы по дисциплине «Технология разработки программных систем». В пособии описываются порядок выполнения и защиты курсовой работы и требования к представляемым документам. Особое внимание обращено на оформление текстовых документов: технического задания и расчетно-пояснительной записки. В приложении приводятся примеры технического задания и оглавления расчетно-пояснительной записки.

Пособие предназначено для студентов третьего курса специальности «Компьютерные системы и сети».

## Введение

Курсовая работа по дисциплине «Технология разработки программных систем» посвящена разработке программного продукта среднего уровня сложности (размером не менее 1000 операторов универсального языка программирования и с развитым пользовательским интерфейсом).

*Цель* курсовой работы – приобретение практических навыков в следующих областях:

- 1) разработка программных продуктов (разработка структуры программного обеспечения, структур данных, алгоритмов и реализующих их программ, разработка тестов и т.п.);
- 2) тестирование и отладка интерактивных систем программного обеспечения;
- 3) составление пояснительной записки, содержащей обоснование принятых проектных решений;

4) применение нормативных документов, регламентирующих состав, содержание и форму технической документации на разработанный программный продукт.

## **1 Организация проведения курсовой работы**

### **1.1 Темы курсовых работ**

Темы курсовых работ выбираются таким образом, чтобы при выполнении работы студенты могли приобрести практические навыки проектирования программных продуктов среднего уровня сложности. В связи с этим желательно наличие у разрабатываемого продукта развитого пользовательского интерфейса и нежелательно использование специальных сред разработки программного обеспечения, например, баз данных.

Объем курсовой работы сильно зависит от сложности задачи: чем сложнее задача, тем меньше может быть объем. Средний объем составляет примерно 1–2 тыс. операторов Паскаля или C++. (Язык программирования и среду студент выбирает самостоятельно, исходя из выбранной технологии программирования, решаемой задачи, а также личных предпочтений).

*Примерные темы курсовых работ:*

Графический редактор для изображения схем алгоритмов.

Система представления табличной информации в графическом виде.

Специализированный текстовый редактор.

Система расчета простейших электрических схем.

Информационно-поисковая система «Астроном».

Система учета успеваемости студентов.

Автоматизированная система складского учета.

Консультирующая медицинская экспертная система.

Система распознавания графической информации.

Web – приложение:

Корпоративный сайт.  
Информационный портал.  
Интернет-магазин.  
Система управления содержимым сайтов.  
Система управления документооборотом.  
Система управления продажами.  
Система бронирования билетов и др.

## 1.2 Порядок выполнения курсовой работы

Порядок выполнения курсовой работы зависит от класса разрабатываемого программного продукта.

При разработке большинства локальных приложений в процессе выполнения курсовой работы студенты должны:

- 1) разработать развернутое техническое задание на программный продукт;
- 2) выполнить анализ задания, выбрать технологию проектирования и разработать проект программного продукта (логическую и физическую модели с учетом динамических аспектов);
- 3) выбрать структуры данных для реализации предметной области программного продукта;
- 4) разработать интерфейс пользователя;
- 5) выбрать стратегию тестирования и разработать тесты;
- 6) выбрать язык и среду программирования;
- 7) разработать алгоритмы и реализовать их в выбранной среде разработки;
- 8) выполнить тестирование и отладку;
- 9) разработать необходимую документацию, указанную в техническом задании.

При разработке веб-приложений студенты должны:

- 1) разработать развернутое техническое задание на веб-приложение;
- 2) провести анализ предметной области и существующих способов реализации задачи;
- 3) выполнить проектирование и разработку дизайна интерфейса пользователя (проектирование логической структуры страниц сайта и логики перехода между ними; выбор способов управления сайтом — меню, всплывающие закладки и пр.; дизайн — подбор расположения элементов, цветовых схем, рисунков и др.);
- 4) выполнить проектирование логики обработки данных;
- 5) выбрать средства для реализации сервера;
- 6) выполнить проектирование серверных компонентов и выбрать методы их тестирования (например, с использованием фреймворка RubyOnRails; выделение основных сущностей — объектный принцип проектирования; проектирование контроллеров, вьюеров, маршрутов; оценка возможности использования стандартных генераторов Rails; проектирование моделей, ограничений и связей; наполнение тестов контроллеров др.);
- 7) выполнить проектирование БД;
- 8) выбрать средства для реализации интерфейса пользователя;
- 9) провести проектирование компонентов интерфейса пользователя;
- 10) реализовать серверные и клиентские компоненты;
- 11) выполнить тестирование веб-приложения (тестирование компонентов; тестирование моделей данных; написание функциональных тестов, тестирование со стороны интерфейса пользователя при помощи Selenium Webdriver и др.);
- 12) разработать необходимую документацию, указанную в техническом задании.

При разработке экспертной системы студенты должны:

- 1) разработать развернутое техническое задание;
- 2) исследовать предметную область (определить основные факты, выполнить логическую декомпозицию, и др.);
- 3) определить структуру системы;
- 4) разработать модель представления знаний для заданной предметной области (определить способы представления декларативных, процедурных и управляющих знаний; разработать схему фреймовой модели др.);
- 5) разработать алгоритмы работы механизма логического вывода;
- 6) разработать семантический интерфейс с пользователем (проработка синтаксиса, семантики, прагматики и др. );
- 7) разработать структуру базы знаний;
- 8) выбрать инструментальное средство разработки системы;
- 9) выполнить кодирование, тестирование и отладку компонентов системы и системы в целом;
- 10) разработать необходимую документацию, указанную в техническом задании.

\*\*\*Конкретный и детальный план разработки программного продукта должен быть согласован с руководителем курсовой работы.

### **1.3 Сроки выполнения отдельных этапов**

Выполнение курсовой работы осуществляется в несколько этапов. В конце каждого этапа студент демонстрирует преподавателю результаты выполнения работы в виде фрагментов расчетно-пояснительной записки и соответствующих текстовых или графических материалов, иллюстрирующих разработку.



Основные этапы курсовой работы, сроки их выполнения и представляемые преподавателю результаты приведены в таблице 1.

**Таблица 1 - Этапы выполнения курсовой работы**

Этап	Содержание этапа	Сроки и объем	Представляемые результаты
1	Разработка технического задания	2 неделя (5 %)	Готовое техническое задание
2	Разработка концептуальной модели предметной области	3 неделя (10 %)	Результаты исследований и декомпозиции предметной области
3	Разработка структуры системы	4 неделя (15%)	Структурная схема и назначение основных компонентов
4	Разработка эскизного проекта	5 неделя (20 %)	Документация, отражающая принципиальные и конструктивные решения (среда функционирования, принципы работы, методы решения, технология разработки и др.)
5	Разработка структуры данных	6 неделя (30 %)	Схемы структур данных, описание основных компонентов данных, фрагменты кодов и др.
6	Разработка интерфейса пользователя	8 неделя (40 %)	Диаграмма состояний интерфейса, формы интерфейса, описание форматов сообщений и др.
7	Разработка модулей ядра	10 неделя (60 %)	Демонстрация работы функций ядра программного продукта, описание технологических реше-

			ний, алгоритмов и др.
8	Разработка вспомога- тельных компонент	12 неде- ля (80 %)	Демонстрация работы вспомога- тельных модулей и их описание и др.
9	Тестирование системы в целом	13 неде- ля (85 %)	Рабочая версия программного продукта, результаты тестирова- ния и др.
10	Оформление докумен- тации	14 неде- ля (98 %)	Полностью оформленная записка в рукописном или печатном ва- рианте.
11	Подготовка к защите и защита курсовой работы	15 Неделя (100%)	Доклад (3-5 минут)

#### 1.4 Порядок защиты курсовой работы

На защиту студент предоставляет:

Техническое задание.

Программный продукт.

Расчетно-пояснительную записку на 20-30 страницах в рукописном или отпечатанном виде (шрифт 12 через 1.5 интервала), содержащую описание разработки и соответствующие иллюстрации.

Программную документацию, указанную в разделе «Требования к программной документации» технического задания.

Программный продукт студент предварительно демонстрирует и сдает преподавателю, который дает отзыв на работу и допускает студента к защите.

В процессе демонстрации программного продукта проверяется:

- соответствие программы техническому заданию;

- работоспособность в различных режимах.

Защита курсовой работы выполняется комиссией, состоящей не менее чем из двух преподавателей.

На защите студент коротко (3–5 мин.) докладывает об основных проектных решениях, принятых в процессе разработки, и отвечает на вопросы членов комиссии.

Оценка за курсовую работу выставляется с учетом:

- качества выполненного программного продукта,
- правильности оформления записки;
- результатов защиты (доклад и ответы на вопросы).

## 2 Содержание расчетно-пояснительной записки

Расчетно-пояснительная записка должна содержать обоснование основных проектных решений, принятых студентом на каждом этапе разработки. Решения должны приниматься исходя из особенностей проектируемого продукта и специфики области его применения. Не должно быть обоснований типа «удобнее», «целесообразнее» и т. п. Необходимо пояснить, чем удобнее, почему целесообразно. По возможности необходимо четко формулировать основания для принятия того или иного решения.

**Пример содержания расчетно-пояснительной записки (технология ООП):**

[Реферат (в содержании не указывается)..... 2]

[Содержание (в содержании не указывается) ..... 3]

Введение..... 4

1. Анализ требований и уточнение спецификаций ..... 6

1.1. Анализ задания и выбор технологии, языка и среды разработки  
..... 6

1.2. Разработка диаграммы вариантов использования 7

1.3. Разработка концептуальной модели предметной области	9
1.4. Выбор методов решения задачи и разработка основных алгоритмов предметной области .....	11
2. Проектирование структуры и компонентов программного продукта	13
2.1. Разработка интерфейса пользователя .....	13
2.1.1. Посторонние графа (диаграммы) состояний интерфейса .....	13
2.1.2. Разработка форм ввода-вывода информации	15
2.3. Разработка структурной схемы программного продукта	18
2.4. Проектирование классов для реализации интерфейса и предметной области .....	20
3. Выбор стратегии тестирования и разработка тестов	21
Заключение .....	24
Литература .....	25
Приложение А. Техническое задание.	
Приложение Б. Руководство пользователя.	

**Пример содержания расчетно-пояснительной записки (структурный подход)**

[Реферат (в содержании не указывается) .....	2]
[Содержание (в содержании не указывается) .....	3]
Введение .....	4
1. Анализ требований и уточнение спецификаций .....	6
1.1. Анализ задания и выбор технологии, языка и среды разработки .....	6
1.2. Анализ процесса обработки информации и построение функциональных диаграмм.....	7
1.3. Анализ хранимой информации и выбор структур данных для ее представления .....	9

1.3. Выбор методов и алгоритмов решения задачи	11
2. Проектирование структуры и компонентов программного продукта	12
2.1. Разработка интерфейса пользователя .....	12
2.1.1. Построение графа (диаграммы) состояний интерфейса	12
2.1.2. Разработка форм ввода-вывода информации	14
2.2. Разработка алгоритма основной программы и структурной схемы программного продукта	16
2.3. Разработка основных алгоритмов программного продукта	18
3. Выбор стратегии тестирования и разработка тестов	21
Заключение .....	24
Литература .....	25
Приложение А. Техническое задание.	
Приложение Б. Руководство пользователя.	

Ниже приводится описание содержимого каждого раздела расчетно-пояснительной записки.

Примечание - В зависимости от выбранной технологии и особенностей разрабатываемого продукта могут иметь место отклонения от рекомендуемой структуры записки. Разделы записки могут объединяться, опускаться, меняться местами. Возможно добавление новых разделов, если этого требует логика изложения. Все интересные материалы, выходящие за пределы рекомендуемого объема записки, выносятся в приложение. Объем приложения не лимитируется.

## 2.1 Реферат

Реферат должен содержать:

- сведения об объеме, количество иллюстраций, таблиц, приложений, количество частей, количество используемых источников
- перечень ключевых слов

- текст реферата.

Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний из текста записки, которые в наибольшей мере характеризуют его содержание и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже и печатаются строчными буквами в строку через запятые.

Текст реферата должен отражать:

- объект исследования или разработки;
- цель работы;
- результаты;
- область применения.

Пример реферата приведен в приложении В

## **2.2 Введение**

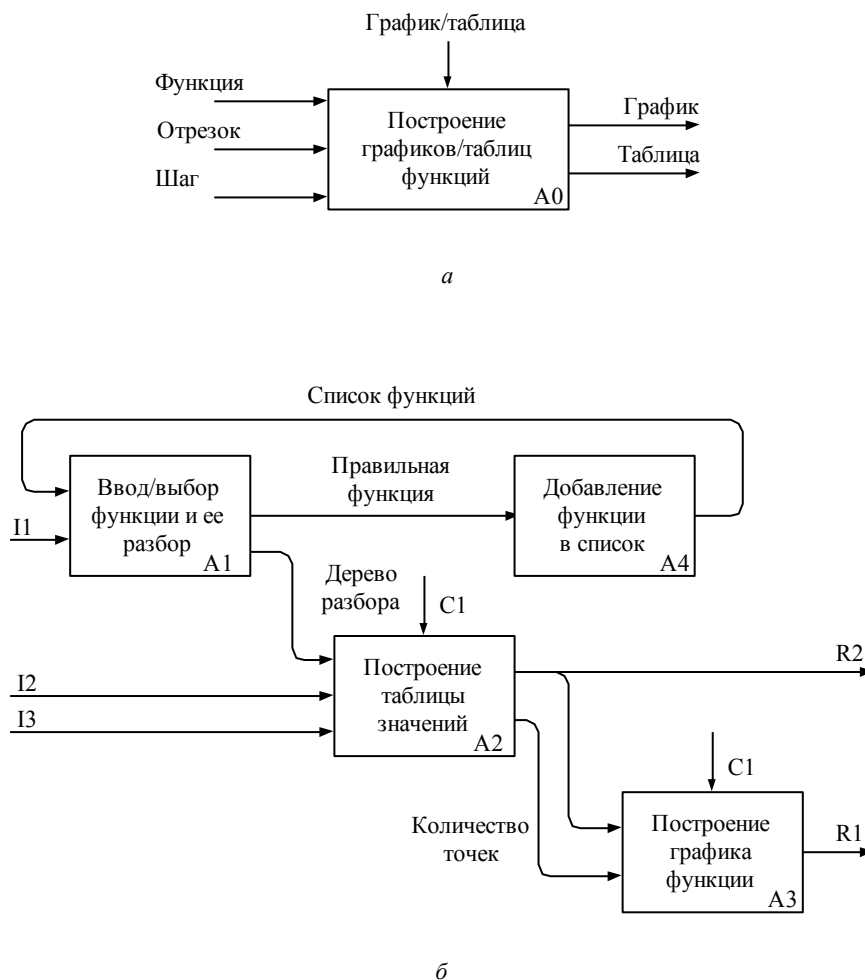
В этом разделе обосновывается актуальность разработки, по возможности, анализируются существующие аналоги, и проводится их сравнение с разрабатываемой системой.

## **2.3 Анализ технических требований и уточнение спецификаций**

Раздел начинают с подраздела анализа задания и выбора технологии, языка и среды разработки. В этом подразделе записки обосновывается и осуществляется выбор одной из современных технологий программирования. А затем поясняется выбор языка и среды разработки.

Примечание - При выборе технологии предпочтение следует отдавать объектно-ориентированному подходу, как обеспечивающему максимальную эффективность разработки.





**Рисунок 3 - Функциональные диаграммы**

Для заданий, при выполнении которых приходится использовать сложные математические методы, и заданий, при реализации которых студентом предлагался собственные оригинальные алгоритмы, обязательным является присутствие в записке обоснование выбора метода (по вычислительной сложности или другим соображениям). В этом же разделе приводятся и сами алгоритмы (в виде схем или псевдокодов), и пояснения к ним.

При наличии сложно организованной информации необходим выбор структур для представления данных, который осуществляется на основе анализа основных процессов обработки данных (статические или динамические, массивы или другие структуры). При необходимости создаются новые структуры данных или модифицируются уже известные. Обычно при выборе структур учитываются следующие параметры: объем и типы данных, а также основные операции над



данными (хранение, поиск, сортировка) и частота обращения к ним в процессе выполнения программы. Если возможны варианты, то производится их оценка по объему требуемой памяти и вычислительной сложности выполнения основных операций.

## 2.4 Проектирование структуры и компонентов программного продукта

При обоих подходах проектирование в большинстве случаев целесообразно начинать с разработки пользовательского интерфейса.

Этот раздел должен начинаться с обзора различных способов и форм взаимодействия пользователя с системой и обоснования выбора определенной формы диалога (лежащего в основе любого взаимодействия) для общения с разрабатываемым программным продуктом. Далее должна определяться структура диалога и приводиться диаграмма диалога интерфейса, отражающая эту структуру. Кроме того, определяется набор необходимых форм (рисунок 4) и строится граф или диаграмма состояний интерфейса (рисунок 5).

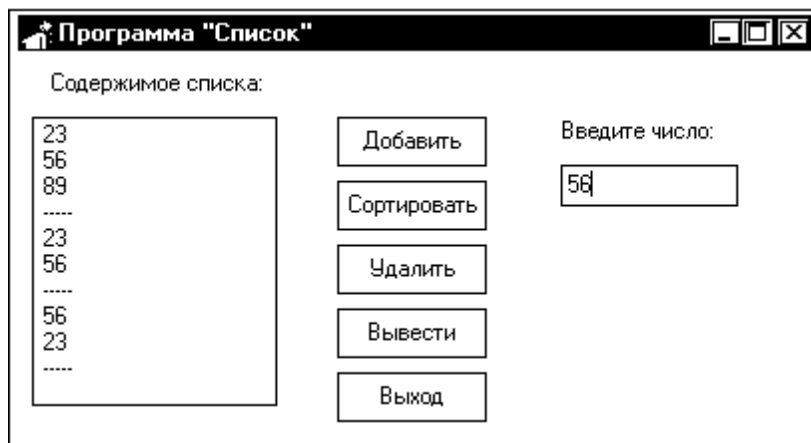
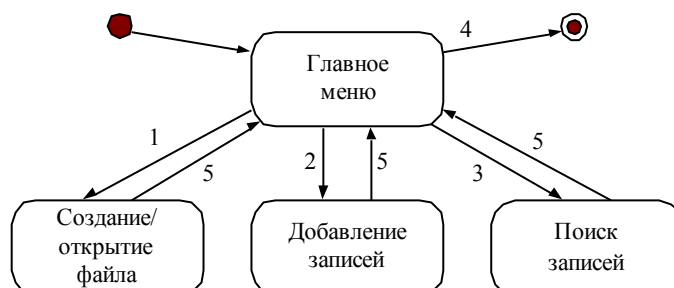


Рисунок 4 - Экранная форма



### Рисунок 5 - Диаграмма состояний интерфейса

В случае табличной формы диалога производится описание всех оконных форм и меню (рисунок 6). В случае использования директивной или фразовой формы описываются основные команды.



Рисунок 6 - Иерархии меню

При использовании событийного программирования необходимо разработать и описать диаграмму состояний интерфейса конкретной формы (рисунок 7), на основе которой затем проектируются обработчики задействованных событий.

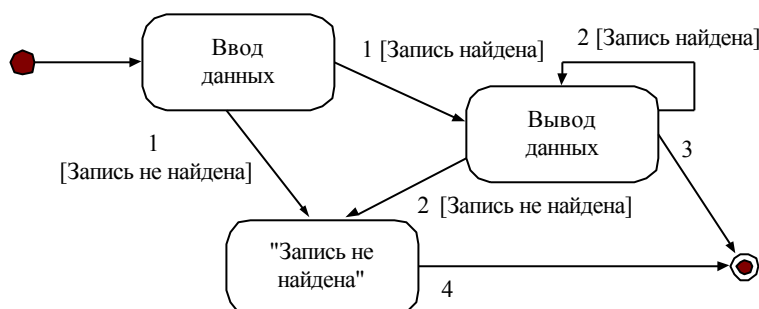


Рисунок 7 - Диаграмма состояний формы интерфейса (при событийном программировании)

После уточнения интерфейса выполняется декомпозиция предметной области задачи в соответствии с выбранной технологией, т. е. создается структурная схема будущего продукта и описывается взаимодействие его функциональных элементов.

*Структурная схема* – схема, отражающая состав и взаимодействие по управлению частей разрабатываемого продукта. При объектной декомпозиции такими частями являются объекты (рисунок 8), при структурной декомпозиции – подпрограммы (рисунок 9).

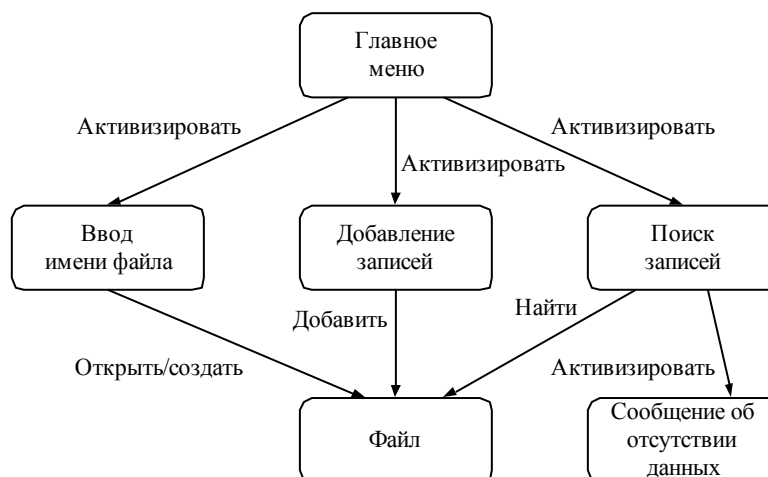


Рисунок 8 - Структурная схема программы (объектная декомпозиция)

Для тем, связанных с нечисловой обработкой данных, этот раздел также должен содержать информационную модель системы, которая может быть представлена функциональной схемой или диаграммой потоков данных. *Функциональная схема* – схема взаимодействия частей системы с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств.

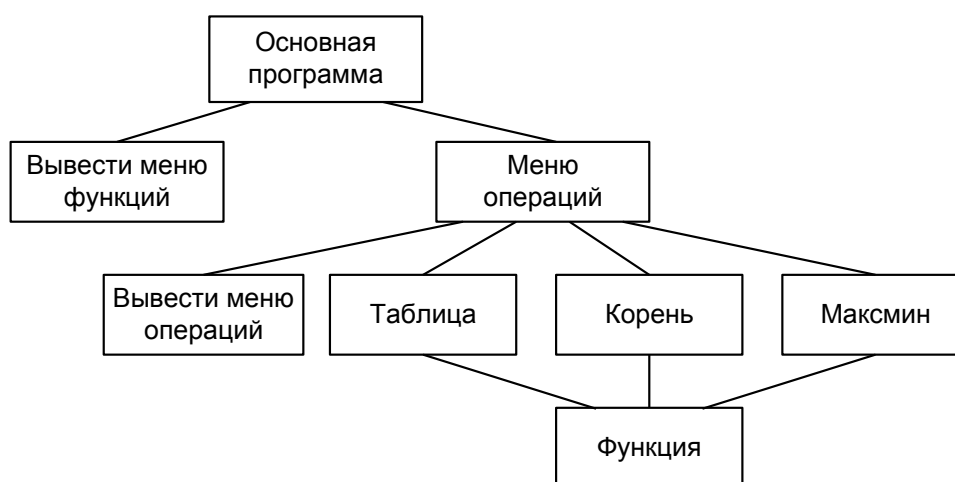


Рисунок 9 - Структурная схема программного продукта (процедурная декомпозиция)

Далее описывается проектирование компонентов в соответствии с выбранной технологией. Для программы, использующей структурный подход к программированию, в данном разделе приводятся обобщенные алгоритмы, напри-

мер, алгоритм основной программы (рисунок 10) и описывается межпрограммный интерфейс подпрограмм.

Всего в записке приводятся два-три из наиболее интересных алгоритмов, включая алгоритм основной программы при структурном подходе.

Примечание - Каждый алгоритм должен быть представлен:

- таблицей и (или) списком используемых в нем глобальных переменных;
- схемой алгоритма, использующей имена переменных, приведенных в таблице или списке; (алгоритм может быть представлен в псевдокодах, если разработчику он более знаком или удобен);
- описанием процесса обработки данных в соответствии с приведенной схемой алгоритма.

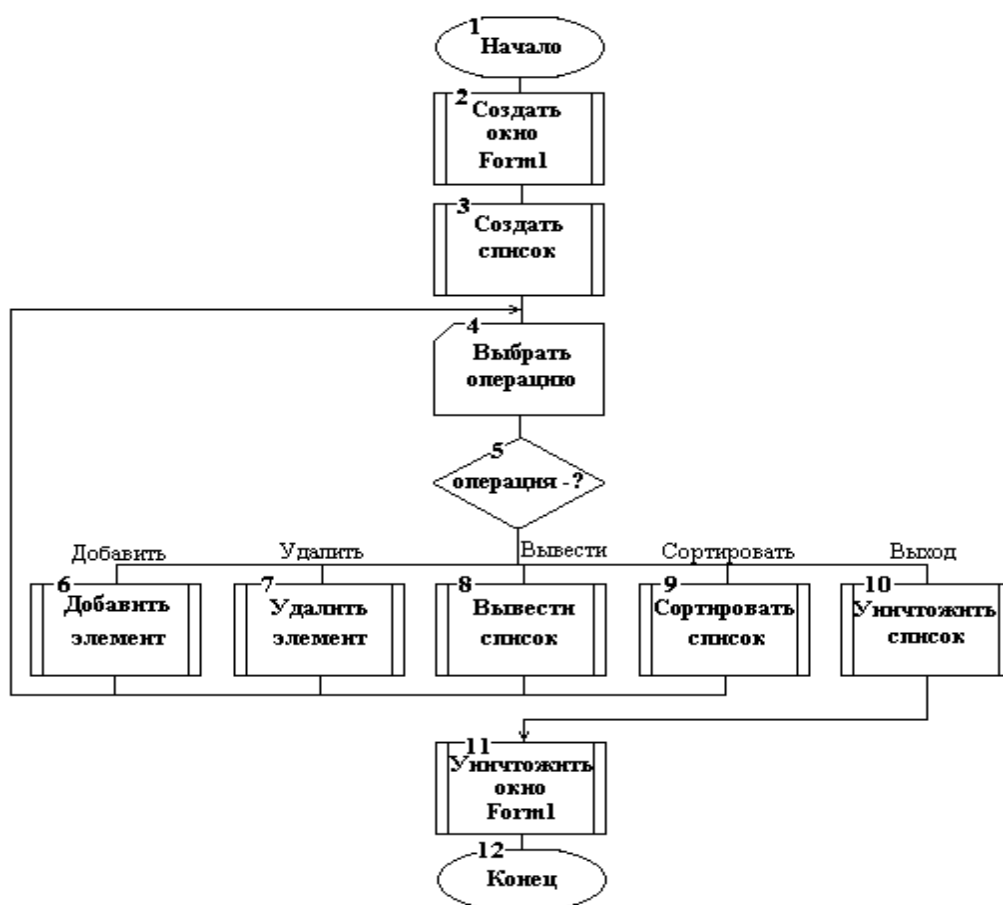


Рисунок 10 – Схема алгоритма основной программы (процедурное программирование)

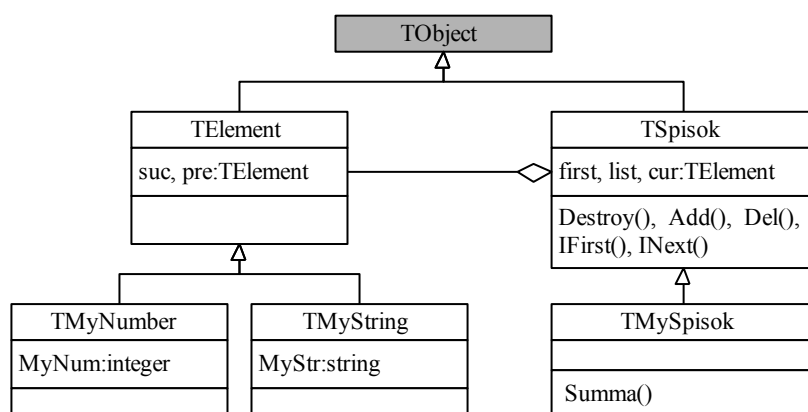
Описание каждого алгоритма должно включать:

- функциональное назначение алгоритма;
- входные и выходные данные (результаты выполнения);
- список формальных параметров и их назначение;
- пример вызова модуля или подпрограммы;
- используемые технические средства;
- ссылку на таблицу переменных алгоритма;
- ссылку на рисунок со схемой алгоритма;
- описание процесса обработки данных в соответствии со схемой;
- если имеется приложение с полным текстом программы, то ссылку на соответствующую страницу приложения.

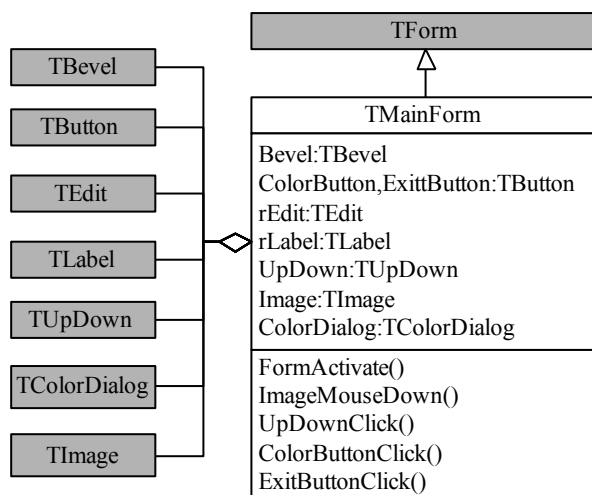
При описании процесса обработки данных в соответствии со схемой алгоритма необходимо пояснить все циклы, каждую альтернативу ветвления, принятое решение по результатам анализа альтернатив и последующие действия.

Тексты описания алгоритмов должны быть структурными, предложения короткими. Описание алгоритма должно отражать суть процесса обработки.

Для программы, при разработке которой использовалась объектно-ориентированная технология, обязательно должна описываться иерархия или диаграмма классов (рисунок 11).



*a*



б

Рисунок 11 - Диаграммы классов предметной области (а)  
и интерфейсной части (б)

Для каждого класса желательно указать дополнительные поля и методы, соответственно обосновывая их назначение и функции. При необходимости здесь же можно привести алгоритмы некоторых методов.

Для пояснения особенностей реализации классов или специфики событийной обработки можно использовать дополнительные иллюстрации, например, диаграммы последовательности действий (рисунок 12).

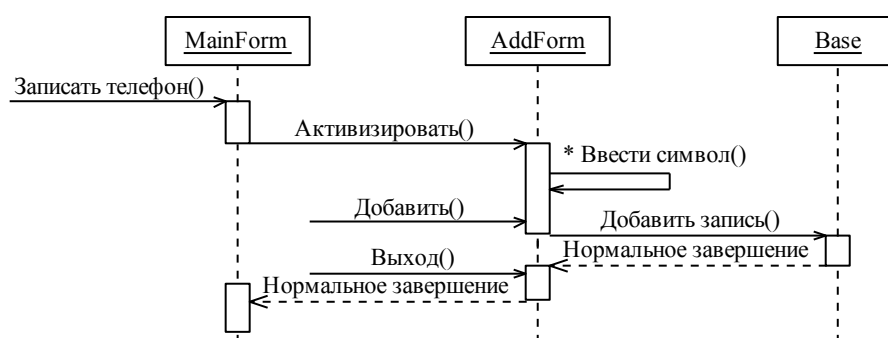


Рисунок 12 - Диаграмма последовательности действий

В завершении раздела описывается декомпозиция разрабатываемой программы на модули и приводится диаграмма компоновки программного продукта (рисунок 13).

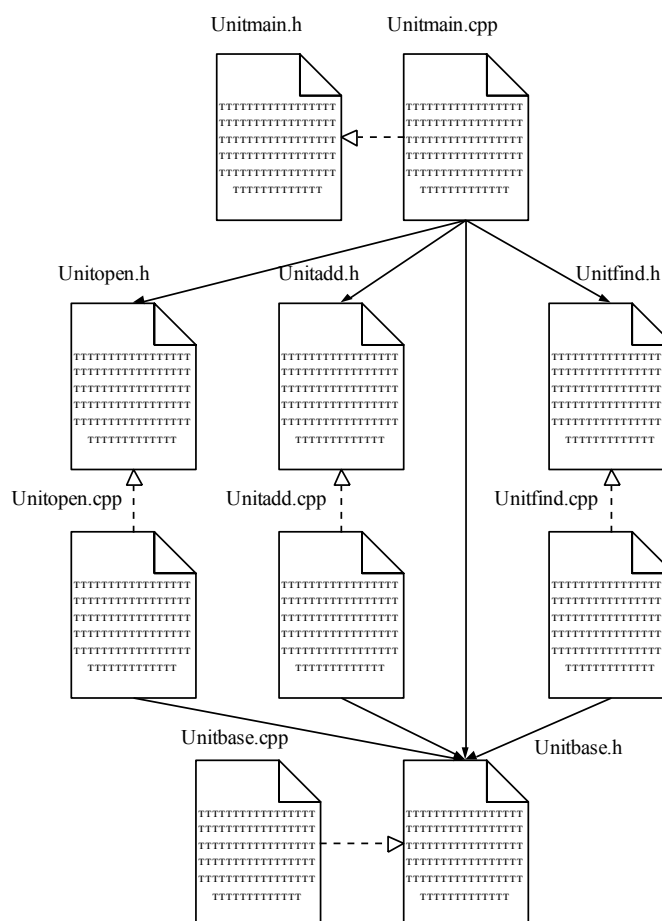


Рисунок 13 - Диаграмма компоновки программного продукта (C++ Builder)

Кроме того, в этом же разделе желательно указать вариант разработки («восходящая» или «нисходящая») и обосновать свой выбор.

## 2.5 Подготовка тестовых данных

В этом разделе выбирается стратегия и методы тестирования. В соответствии с выбранной стратегией и методами строятся примеры тестов (обязательно с предполагаемыми результатами тестирования). Данные тестов рекомендуется представить в виде таблиц.

## **2.6 Заключение**

Раздел содержит выводы по разработанному продукту, рекомендации по его использованию и возможные направления дальнейшего совершенствования.

## **3 Правила оформления пояснительной записки**

### **3.1 Оформление текстового и графического материала**

Пояснительная записка оформляется на листах формата А4. Графический материал можно оформлять на листах формата А3. Поля на листе определяются в соответствии с общими требованиями. При использовании текстовых редакторов для оформления записки параметры страницы заказываются в зависимости от устройства печати. При ручном оформлении выбираются из соображений удобства.

Нумерация страниц – сквозная. Номер проставляется сверху справа арабской цифрой. Страницами являются листы с текстами, рисунками и текстами приложения.

Первая страница – титульный лист расчетно-пояснительной записки. Номер страницы на титульном листе не проставляется. Образец титульного листа представлен в Приложении Б.

Вторая страница – реферат.

Третья страница – оглавление, отражающее содержание изложенного материала. Ни реферат, ни само оглавление в содержании не упоминаются.

Затем следуют разделы записки в порядке, определенном логикой изложения материала.

Записка завершается списком литературы.

Далее могут следовать приложения, содержащие материал, не вошедший в записку по причине ее ограниченного размера, но интересный для более глубо-



кого понимания назначения и возможностей разработки. Расчетно-пояснительная записка может содержать одно и более приложений.

Наименование разделов и подразделов пишется строчными буквами, кроме первой прописной. Расстояние между заголовками и текстом, а также между заголовками раздела и подразделов должно быть равно:

- при выполнении документа машинописным способом – двум интервалам;
- при выполнении рукописным способом – 10 мм;
- при использовании текстовых редакторов – определяется возможностями редактора.

Расстояние между последней строкой текста предыдущего раздела и последующим заголовком при расположении их на одной странице должно быть равно:

- при выполнении документа машинописным способом – трем интервалам;
- при выполнении рукописным способом – не менее 15 мм;
- при использовании текстовых редакторов – определяется возможностями редактора.

Разделы и подразделы нумеруются арабскими цифрами с точкой. Разделы должны иметь порядковые номера 1, 2, и т. д. Номер подраздела включает номер раздела и порядковый номер подраздела, входящего в данный раздел, разделенные точкой. Например: 2.1., 3.5.

Перечисления надо нумеровать арабскими цифрами со скобкой; Например: 2), 3) и т. д. – с абзацного отступа. Допускается выделять перечисление простановкой дефиса перед пунктом текста или символом, его заменяющим, в текстовых редакторах.

### 3.2 Оформление рисунков, схем алгоритмов, таблиц и формул

Иллюстрации (графики, схемы, диаграммы) могут быть приведены как в основном тексте, так и в приложении. Все иллюстрации именуются рисунками. Все рисунки, таблицы и формулы нумеруются арабскими цифрами последовательно (сквозная нумерация). В приложении – в пределах приложения. Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные. Чертежи, графики, диаграммы, схемы должны соответствовать требованиям ЕСКД.

Рисунки, за исключением рисунков приложений следует нумеровать арабскими цифрами сквозной нумерацией.

Допускается нумеровать рисунки в пределах раздела. В этом случае номер рисунка состоит из номера раздела и порядкового номера рисунка, разделенных точкой.

Если рисунок один, то он обозначается «Рисунок 1». Рисунок может иметь наименование и пояснительные данные (подрисуночный текст). Слово «рисунок» и его наименование располагают посередине строки. Если есть подрисуночный текст, то слово «рисунок» и его наименование помещают после пояснительных данных.

Например:

Рисунок 12 - Форма окна основного меню

На все рисунки, таблицы и формулы в записке должны быть ссылки в виде: «(рисунок 12)» или « форма окна основного меню приведена на рисунке 12».

Слово «рисунок» и его наименование располагается посередине строки.

Рисунки и таблицы должны размещаться сразу после той страницы, на которой, в тексте записки, она упоминается в первый раз. Если позволяет место, рисунок (таблица) может размещаться в тексте на той же странице, где на него дается первая ссылка.

Если рисунок занимает более одной страницы, на всех страницах, кроме первой, проставляется номер рисунка и слово «Продолжение». Например:

### Рисунок 12 - Продолжение

Рисунки следует размещать так, чтобы их можно было рассматривать без поворота записки. Если такое размещение невозможно, рисунки следует располагать так, чтобы для рассматривания надо было повернуть записку по часовой стрелке. В этом случае верхним краем является левый край страницы. Расположение и размеры полей сохраняются в соответствии с установленными.

Схемы алгоритмов должны быть выполнены в соответствии со стандартом ЕСПД. Толщина сплошной линии при вычерчивании схем алгоритмов должна быть в пределах от 0,6 до 1,5 мм. Надписи на схемах должны быть выполнены чертежным шрифтом. Высота букв и цифр должна быть менее 3,5 мм.

Название таблицы, при его наличии, должно отражать ее содержание, быть точным, кратким. Название следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире. Заголовок, кроме первой буквы, выполняется строчными буквами. В аббревиатурах используются только заглавные буквы. Например: ПЭВМ.

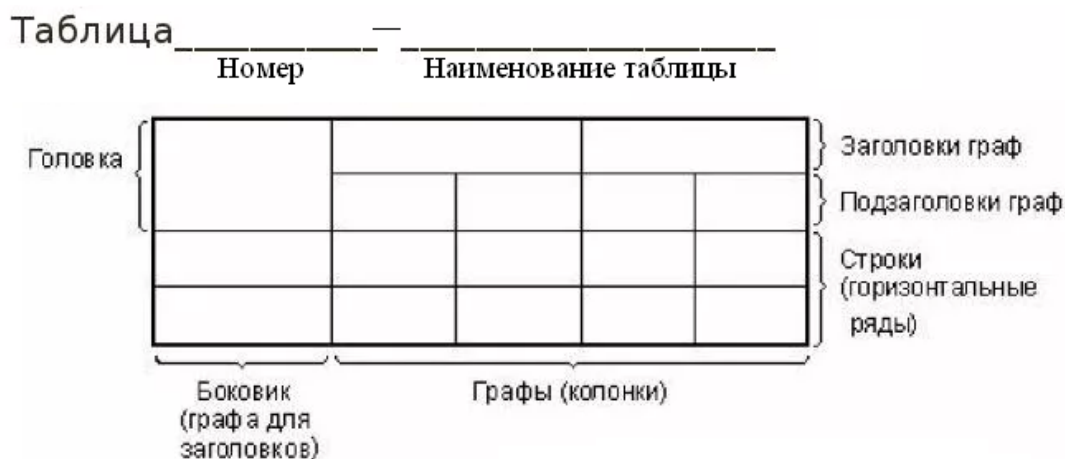
При переносе части таблицы название помещается только над первой частью таблицы, нижнюю горизонтальную черту, ограничивающую таблицу, не проводят.

Ссылки на таблицы в тексте пояснительной записки должны быть в виде слова «таблица» с указанием ее номера. Например, «Результаты тестов приведены в таблице 4».

Таблицу с большим количеством строк допускается переносить на другой лист (страницу). При переносе слово «Таблица» и ее номер указывается один раз справа над первой частью таблицы, а над другими частями пишут слово «Продолжение» и указывают номер таблицы, например: «Продолжение таблицы 1». При этом заголовок помещают только над ее первой частью.

Оформление таблиц должно соответствовать ГОСТ 1.5 и ГОСТ 2.105.

Пример оформления таблицы приведен на рисунке 14.



**Рисунок 14** - Пример оформления таблицы

Таблицы, за исключением таблиц приложений следует нумеровать арабскими цифрами сквозной нумерацией.

Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Уравнения и формулы следует выделять из текста в отдельную строку. Ниже и выше каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Пояснения значений символов и числовых коэффициентов следует проводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Формулы следует нумеровать порядковой нумерацией в пределах записки арабскими цифрами. Номер формулы ставится в крайне правом положении на строке в круглых скобках на уровне формулы. Например:

$$z := \sin(x) + \ln(y);$$

(12)

Ссылка на номер формулы дается в скобках. Например: «расчет значений производится по формуле (12)».

### *Оформление примечаний.*

Слово «Примечание» следует печатать с прописной буквы с абзаца и не подчеркивать. Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается с прописной буквы и примечание не номеруется. Несколько примечаний нумеруются по порядку арабскими цифрами без представления точки. Примечание к таблице помещают в конце таблицы над линией, обозначающей окончание таблицы.

Пример одного примечания:

Примечание - \_\_\_\_\_

Пример нескольких примечаний:

Примечания

- 1 \_\_\_\_\_
- 2 \_\_\_\_\_
- 3 \_\_\_\_\_

### **3.3 Оформление текстов программ**

Тексты программ должны оформляться в соответствии с «хорошим стилем» программирования, т.е. должны быть легко читаемы и хорошо документированы. В текстах должны быть комментарии:

- 1) после заголовка программы или подпрограммы приводится общая информация: назначение, входные данные, результаты, метод решения; данные о программисте, дата написания, версия;
- 2) при объявлении данных - назначение переменных;
- 3) в начале и в конце определенной функционально законченной части программы;
- 4) для пояснения логических частей программы (ветвлений, циклов).

Однако комментарии не должны затенять структуру текста и должны быть ясными и краткими. Наименование программ и подпрограмм должны отражать их назначение. Логическая структура программы должна быть отражена в ее тексте с помощью:

- 1) пустых строк между текстами подпрограмм и отдельных ее функционально законченных частей;
- 2) сдвигами текста в строке при написании:
  - заголовков вложенных циклов;
  - тела цикла после его заголовка;
  - альтернатив разветвлений процесса обработки данных.

### 3.4 Оформление приложений

Каждое приложение должно начинаться с новой страницы с указанием наверху посередине страницы слова «ПРИЛОЖЕНИЕ» прописными буквами и иметь тематический заголовок, который записывается симметрично относительно текста с прописной буквы отдельной строкой. Приложения обозначаются заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, Щ, Ч, Ъ, Ы, Ь. Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O. Если в документе одно приложение, оно обозначается «ПРИЛОЖЕНИЕ А».

Например:

#### ПРИЛОЖЕНИЕ В

Титульный лист расчетно-пояснительной записки.

Рисунки и таблицы, помещаемые в приложении, нумеруются арабскими цифрами в пределах каждого приложения с добавлением буквы приложения. Например:

- рисунок А.12 - 12-й рисунок приложения А;
- рисунок В.2 - 2-й рисунок приложения В;

- таблица Г.3 - 3-я таблица приложения Г.

Каждый файл в приложении оформляется как рисунок с наименованием файла и его назначением. Например:

Рисунок А.3 - Файл **mod1.pas** – исходные тексты библиотеки процедур обработки.

Рисунок В.4 - Файл **menuran.pas** – программа движения курсора основного меню.

### 3.5 Оформление списка литературы

Список литературы должен включать все использованные источники. Сведения о книгах (монографиях, учебниках, пособиях, справочниках и т.д.) должны содержать: фамилию и инициалы автора, заглавие книги, место издания, издательство, год издания. При наличии трех и более авторов допускается указывать фамилию и инициалы только первого из них со словами «и др.». Наименование места издания надо приводить полностью в именительном падеже: допускается сокращение названия только двух городов: Москва (М.) и Ленинград (Л.).

Сведения о статье из периодического издания должны включать: фамилию и инициалы автора, наименование статьи, наименование издания (журнала), наименование серии (если она есть), год выпуска, том (если есть), номер издания (журнала) и номера страниц, на которых помещена статья.

При ссылке на источник из списка литературы (особенно при обзоре аналогов) надо указывать порядковый номер по списку литературы, заключенный в квадратные скобки; например: [5].

#### 4 Критерии оценки курсовой работы

№	Раздел оценки, срок	Кто оценивает	Баллы	Пункты оценки	Критерии оценки
1	Рабочая версия программного продукта, 12 неделя	Руководитель	30-50	Соответствие программного продукта требованиям технического задания	Выполнение требований, предъявляемых к функционалу программного продукта, временным характеристикам, организации входных и выходных данных, защиты.
				Технологичность программного продукта	Проработанность модели, стиль программирования, степень независимости модулей и степень повторного использования кодов.
				Универсальность программного продукта	Количество и сложность дополнительных функций реализованных в программном продукте.
2	Оформление документации, 14 – 16 недели	Комиссия	18-30	Соответствие требованиям ГОСТ, ЕСКД и ЕСПД	Выполнение требований, предъявляемых к объему и оформлению представленной документации.
				Полнота представления информации	Наличие основных разделов и степень их информативности.
				Точность представленной информации (формулировки и др.)	Однозначность формулировок и последовательность представления информации.
3	Защита курсовой работы, 14 – 16 недели	Комиссия	12-20	Доклад (5-6 мин)	Полнота и последовательность изложения полученных результатов по основным этапам (исследование, проектирование и реализация)
				Ответы на вопросы	Лаконичность ответов (краткость и ясность изложения мыслей)
			60-100		

При получении за любую часть работы баллов менее, чем минимально определенное в таблице количество, студент получает «неудовлетворительно».

Итоговая оценка проставляется с использованием стандартной таблицы соответствия:

Рейтинг	Оценка на дифференцированном зачёте
85 – 100	отлично
71 – 84	хорошо
60 – 70	удовлетворительно
0 – 59	неудовлетворительно



## ПРИЛОЖЕНИЕ А. Техническое задание на курсовую работу

Ниже приведен титульный лист и пример технического задания.

---

**Государственное образовательное учреждение высшего профессионального образования**



**«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ \_\_\_\_\_

КАФЕДРА \_\_\_\_\_

**«Система учета текущей успеваемости студентов»**

Техническое задание на курсовую работу.

**Студент гр. ИУ6-51:** \_\_\_\_\_ (Иванов И. И.)

**Руководитель:** \_\_\_\_\_ (Петров П.П.)

Москва 2018

## **1 Введение**

Напряженный график контрольных мероприятий учебного семестра требует от студента регулярной ежедневной работы. Наличие контроля выполнения графика со стороны деканата и куратора позволяет существенно повысить успеваемость студентов. Автоматизированная система учета текущей успеваемости позволить улучшить качество контроля успеваемости со стороны куратора и деканата, так как позволит получать сведения о динамике работы каждого студента.

## **2 Назначение**

Система предназначена для хранения и обработки сведений о текущей успеваемости студентов учебной группы.

Система может быть использована при проведении смотров и для оценки текущей успеваемости каждого студента и группы в целом.

## **3 Требования к программе или программному изделию**

### *3.1 Требования к функциональным характеристикам*

Система должна обеспечивать возможность выполнения следующих функций:

- \* инициализацию системы (ввод списка группы и т.п.);
- \* ввод и коррекцию текущей информации о ходе выполнения учебного графика конкретным студентом;
- \* хранение информации в течение длительного времени;
- \* получение сведений о текущем состоянии выполнения учебного графика студентами в следующих вариантах:

- а) процент успеваемости по конкретному студенту по всем предметам;
- б) процент успеваемости по всем студентам по конкретному заданию;
- в) проценты успеваемости по всем студентам и группе в целом по всем предметам;
- г) список студентов, не сдавших конкретное задание;

д) процент успеваемости каждого студента и группы по конкретному предмету.

Исходные данные:

- ◆ список студентов группы;
- ◆ перечень предметов, перечень заданий по каждому предмету и сроки их сдачи;
- ◆ текущие данные (еженедельно): процент выполнения каждым студентом каждого задания учебного графика.

### *3.2 Требования к надежности*

1. Программный продукт должен соответствовать современному уровню требований к разработке программного обеспечения (структурному и объектно-ориентированному подходам).

2. Предусмотреть контроль вводимой информации.

3. Предусмотреть блокировку некорректных действий пользователя при работе с системой.

4. Обеспечить сохранение результатов предыдущей корректировки файла данных в файле с расширением .bak.

### *3.3 Требования к составу и параметрам технических средств*

Система должна работать на IBM совместимых персональных компьютерах.

Минимальная конфигурация:

- |                    |            |
|--------------------|------------|
| ◆ тип процессора   | Intel 386; |
| ◆ объем ОЗУ        | 8 Мб;      |
| ◆ тип монитора     | EGA;       |
| ◆ тип манипулятора | мышь.      |

### *3.4 Требования к информационной и программной совместимости*

Система должна работать под управлением MS DOS 6.2 и выше.

## **4 Требования к программной документации**

- ◆ Разрабатываемые программные модули должны быть самодокументированными, т.е. тексты программ должны содержать все необходимые комментарии.
- ◆ В состав сопровождающей документации должны входить:
  - а) расчетно-пояснительная записка;
  - б) руководство пользователя.

**ПРИЛОЖЕНИЕ Б. Титульный лист расчетно-пояснительной записки**

Государственное образовательное учреждение высшего профессионального образования



**«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ \_\_\_\_\_

КАФЕДРА \_\_\_\_\_

**РАСЧЁТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

\_\_\_\_\_  
\_\_\_\_\_

Студент \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Руководитель  
курсовой работы \_\_\_\_\_  
(Подпись, дата) (И.О.Фамилия)

Москва, 2018

## ПРИЛОЖЕНИЕ В. Пример реферата на расчетно-пояснительную записку

### Реферат

Записка 25с., 1ч., 15 рис., 4 табл., 12 источников, 2 прил.

ФУНКЦИЯ, ПРОИЗВОДНАЯ, ЭКСТРЕМУМ, ТОЧКА ПЕРЕГИБА, ГРАФИК, ТАЛИЦА ЗНАЧЕНИЙ, ШАГ, ОТРЕЗОК, РАЗБОР ФУНКЦИИ.

Объектом разработки является автоматизированная система исследования функций одной переменной.

Цель работы – проектирование автоматизированной системы исследования функций одной переменной, позволяющая:

- ввод и разбор функции одной переменной в аналитическом виде;
- построение графиков и таблиц исследуемых функций на выбранном отрезке;
- нахождение производной исследуемой функции в аналитическом виде на выбранном отрезке;
- нахождение экстремумов и точек перегибов исследуемой функции.

В результате разработки была спроектирована и реализована автоматизированная система исследования функций одной переменной.

Пользователями системы могут быть школьники, студенты и инженеры, занимающиеся исследованиями указанных функций.