

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Кафедра математического обеспечения информационных систем

И.В. ВЛАЦКАЯ, Н.С. ШАМСУДИНОВА

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Рекомендовано Редакционно-издательским советом
государственного образовательного учреждения
высшего профессионального образования
«Оренбургский государственный университет»

Оренбург 2008

УДК 004.4(076.5)

ББК 32.973 – 018.2я73

В 58

Рецензент:

кандидат педагогических наук, доцент Шухман А.Е.

Влацкая И.В.

Технология разработки программного обеспечения: методические указания для выполнения курсовой работы /И.В. Влацкая, Н.С. Шамсудинова, – Оренбург: ГОУ ОГУ, 2007. - 29 с.

Настоящие методические указания содержат рекомендации по выполнению и оформлению курсовой работы по дисциплине «Технология разработки программного обеспечения» (ТРПО). В пособии описываются порядок выполнения и защиты курсовой работы и требования к представляемым документам. Особое внимание обращено на оформление пояснительной записки. В приложении приводятся примеры оформления задания, оглавления пояснительной записки и титульных листов в соответствии со стандартами.

Методические указания предназначены для студентов второго курса специальности «Математическое обеспечение и администрирование информационных систем».

ББК 32.973 – 018.2я73

© Влацкая И.В., Шамсудинова Н.С., 2007
© Оренбург, ГОУ ОГУ, 2007

Содержание

Введение.....	4
1 Организация проведения курсовой работы.....	5
1.1 Темы курсовых работ	5
1.2 Порядок выполнения курсовой работы.....	5
1.3 Сроки выполнения отдельных этапов.....	6
1.4 Порядок защиты курсовой работы	7
2 Содержание пояснительной записки	8
3 Правила оформления пояснительной записки	15
3.1 Оформление текстового материала	15
3.2 Требования к содержанию структурных элементов текстовой части ...	15
3.3 Оформление иллюстраций	17
4 Основные стандарты разработки ПО	19
4.1 Нормативные документы по стандартизации и виды стандартов	19
4.2 Стандарты в области программного, обеспечения	20
Приложение А Титульный лист для курсовой работы	24
Приложение Б Форма бланка задания	25
Приложение В Примеры библиографических записей	26
Приложение Г Пример оформления содержания	29

Введение

Создание программного средства — динамически длительный и трудоемкий процесс. Современные технологии проектирования основаны на последовательной (поэтапной) разработке. По общности целей последовательности работ (этапы) обычно объединяются в стадии.

Технология разработки программного обеспечения (ТРПО) — система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах.

Курсовая работа по дисциплине ТРПО 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 области:

1) разработки:

- структурной и функциональной схем программного обеспечения;
- структур данных;
- алгоритмов и реализующих их программ;
- стратегии тестирования и тестовых данных;

2) отладки интерактивных систем программного обеспечения;

3) составления пояснительной записки, содержащей обоснование принятых проектных решений;

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

При разработке программного средства следует учитывать следующие требования:

– стандартизацию основных этапов жизненного цикла программных средств;

– обеспечение надежности и качества функционирования программного средства;

– тестирование нового программного средства.

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 сложных средств разработки, например, баз данных.

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

Примерные темы курсовых работ по дисциплине:

- Обработка экспертных оценок методом анализа иерархий.
- Проектирование обобщенного критерия оценки программного средства.
- Анализ и реализация криптографических алгоритмов (виды алгоритмов по выбору студентов).
- Автоматизированная система контроля знаний (по заказу кафедр).
- Мультиагентные системы принятия решений на предприятии.
- Автоматизация работы организации по трудоустройству населения.
- Автоматизированное рабочее место оператора туристической фирмы.
- Разработка Интернет-приложения «Студенческая интернет-газета».
- Автоматизация учета реализации книжной продукции.
- Анализ и реализация алгоритмов сортировки (виды алгоритмов по выбору студентов).
- Разработка автоматизированной системы для обработки социологических опросов.
- Разработка автоматизированного рабочего места для диспетчера справочной системы железнодорожных сетей.
- Анализ и реализация алгоритмов поиска кратчайшего пути в графах.
- Разработка редактора структурных химических формул.
- Автоматизированная система учета успеваемости студентов.
- Автоматизированная система учета и анализа демографической ситуации в Оренбургской области.
- Разработка автоматизированного рабочего места для работы администратора культурно-массовых заведений.

1.2 Порядок выполнения курсовой работы

В процессе выполнения курсовой работы студенты должны:

- 1) провести анализ предметной области;
- 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, наиболее удовлетворяющий проведенным разработкам;
- 7) разработать алгоритмы и реализовать их в выбранной среде разработки;
- 8) выполнить тестирование и отладку;
- 9) разработать необходимую документацию, указанную в техническом задании.

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 пояснительной записки и соответствующих текстовых или графических материалов, иллюстрирующих разработку.

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

Таблица 1 - Этапы выполнения курсовой работы

Этап	Содержание этапа	Сроки и объем	Представляемые результаты
1	2	3	4
1	Разработка технического задания и проекта программы	4 неделя (25%).	Готовое техническое задание
2	Выбор структур данных, разработка интерфейса пользователя, разработка иерархии объектов, подготовка данных для тестирования	8 неделя (50%).	Описание структур данных, структурная и функциональная схемы продукта, граф диалога интерфейса и описание основных экранных форм, стратегия тестирования и тестовые данные

Продолжение таблицы 1

1	2	3	4
3	Разработка алгоритмов и реализация программы на выбранном языке программирования.	12 неделя (75%).	Схемы алгоритмов, тексты программ системы, сам продукт на дискете для проверки функционирования
4	Подготовка расчетно-пояснительной записки	16 неделя (100%).	Полностью оформленная записка в рукописном или печатном варианте
5	Защита курсовой работы	17 неделя	Программа, записка, приложения

1.4 Порядок защиты курсовой работы

На защиту студент предоставляет:

1 Исследовательскую часть.

2 Программный продукт.

3 Пояснительную записку на 20-25 страницах в отпечатанном виде, содержащую описание разработки в соответствии с требованиями, указанными в разделе «Правила оформления пояснительной записки»

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 предметной области;

- работоспособность в различных режимах.

Защита курсовой работы проводится в открытой форме. На защиту приглашаются студенты и преподаватели выпускающих кафедр.

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

Оценка за курсовую работу выставляется с учетом:

- качества выполненного программного продукта;

- использования технологических приемов проектирования и разработки ПО;

- умением четко формулировать цель и излагать задачи курсовой работы;

- правильности оформления записки;

- результатов защиты.

2 Содержание пояснительной записки

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.

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

2) Анализ задания и выбор технологии, языка и среды разработки. В этом разделе записки обосновывается и осуществляется выбор одной из современных технологий программирования. А затем поясняется выбор языка и среды разработки.

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

3) Определение структуры программного продукта. В данном разделе проводится анализ предметной области задачи и ее разбиение (декомпозиция) в соответствии с выбранной технологией, т.е. создается структурная схема будущего продукта и описывается взаимодействие ее функциональных элементов.

Структурная схема - схема, отражающая состав и взаимодействие по управлению частей разрабатываемого продукта. При объектной декомпозиции такими частями являются объекты (рисунок 1), при структурной декомпозиции – подпрограммы (рисунок 2).

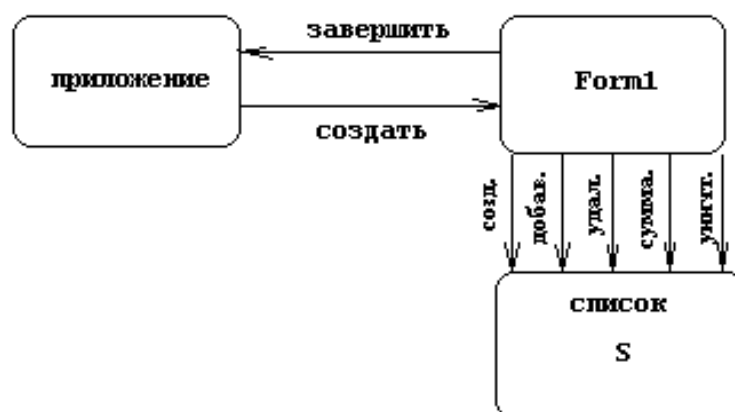


Рисунок 1 - Структурная схема программного продукта
(объектная декомпозиция)



Рисунок 2 - Структурная схема программного продукта (процедурная декомпозиция)

Для тем, связанных с нечисловой обработкой данных, этот раздел должен содержать информационную модель системы, которая может быть представлена функциональной схемой.

Функциональная схема - схема взаимодействия частей системы с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств.

Большое значение при разработке структуры программного продукта имеет выбор алгоритмов предметной области и структур данных.

Для заданий, при выполнении которых приходится использовать сложные математические методы, и заданий, при реализации которых студентом предлагались 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.

Выбор структур данных осуществляется на основе анализа основных процессов обработки информации (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.

Построение диаграммы вариантов использования. Вариант использования (или прецедент) применяется для спецификации общих особенностей поведения системы или любой другой сущности предметной области без рассмотрения внутренней структуры этой сущности. Каждый вариант использования определяет последовательность действий, которые должны быть вы-

полнены проектируемой системой при взаимодействии ее с соответствующим актером. Диаграмма вариантов может дополняться пояснительным текстом, который раскрывает смысл или семантику составляющих ее компонентов.

Цель варианта использования заключается в том, чтобы определить законченный аспект или фрагмент поведения некоторой сущности без раскрытия внутренней структуры этой сущности. В качестве такой сущности может выступать исходная система или любой другой элемент модели, который обладает собственным поведением, подобно подсистеме или классу в модели системы.

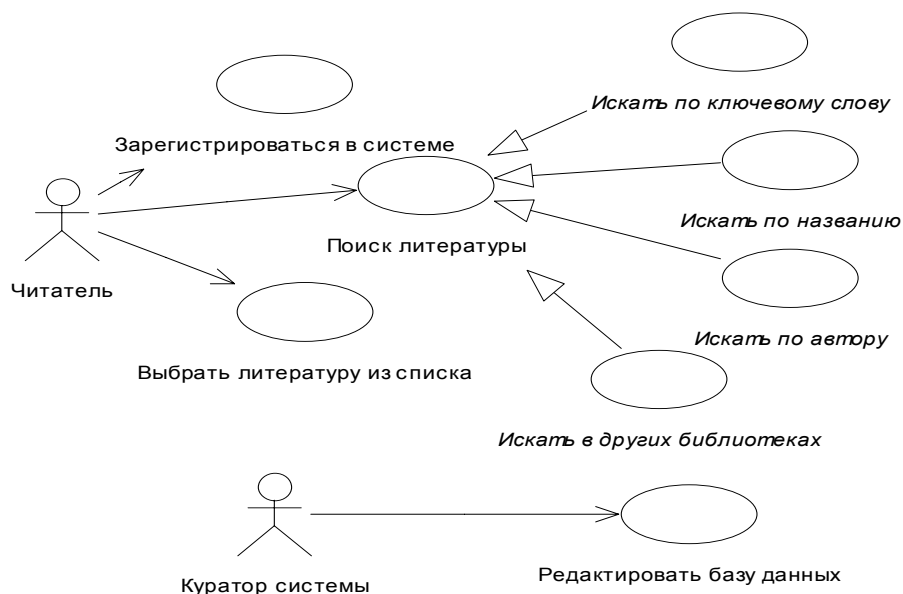


Рисунок 3 - Диаграмма вариантов использования для библиотечной базы данных

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы. Для удобства множество вариантов использования может рассматриваться как отдельный пакет.

Примерами вариантов использования могут являться следующие действия: проверка состояния текущего счета клиента, оформление заказа на покупку товара, получение дополнительной информации о кредитоспособности клиента, отображение графической формы на экране монитора и другие действия.

4) Разработка пользовательского интерфейса. Этот раздел содержит обзор различных способов и форм взаимодействия пользователя с системой и обоснование выбора определенной формы диалога (лежащего в основе любого взаимодействия) для общения с разрабатываемым программным продуктом. Проводится разработка структуры диалога и приводится граф диалога интерфейса, отражающий эту структуру. В случае табличной формы диалога дается описание всех оконных форм и меню (рисунки 4 и 5). В случае использования директивной или фразовой формы описываются основные команды.



Рисунок 4 - Иерархия меню

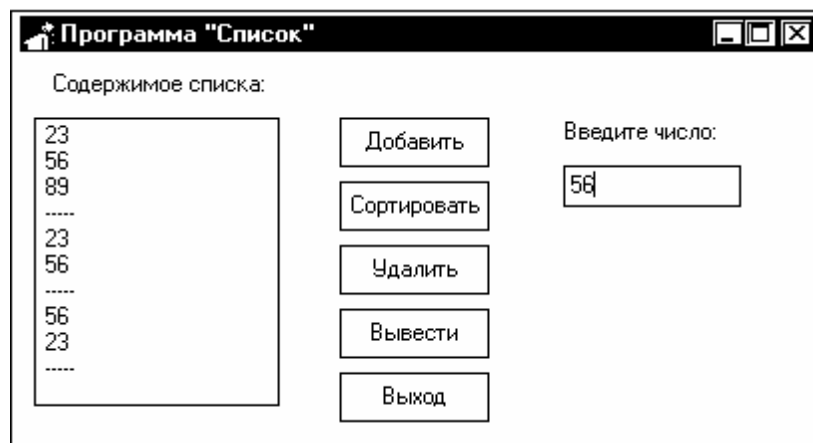


Рисунок 5 - Экранная форма

При использовании событийного программирования необходимо разработать и описать граф состояний интерфейса (рисунок 6), на основе которого затем проектируются обработчики задействованных событий.

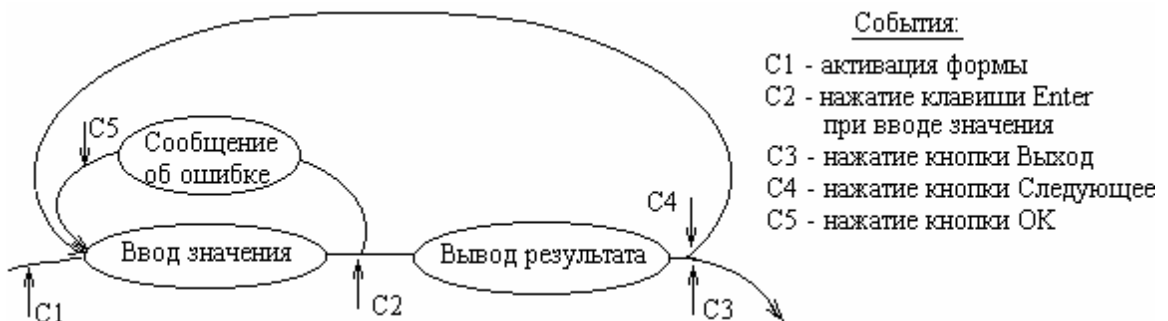


Рисунок 6 - Граф состояний интерфейса (при событийном программировании)

5) Описание реализации программного продукта. Для программы, использующей структурный подход к программированию, в данном разделе приводятся обобщенные алгоритмы, например, алгоритм основной программы (рисунок 7) и описывается межпрограммный интерфейс подпрограмм.

Для программы, при разработке которой использовалась объектно-ориентированная технология, обязательно должна описываться иерархия или диаграмма классов. Для каждого класса желательно указать дополнительные поля и методы, соответственно обосновывая их назначение и функции. При необходимости здесь же можно привести алгоритмы некоторых методов.

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.

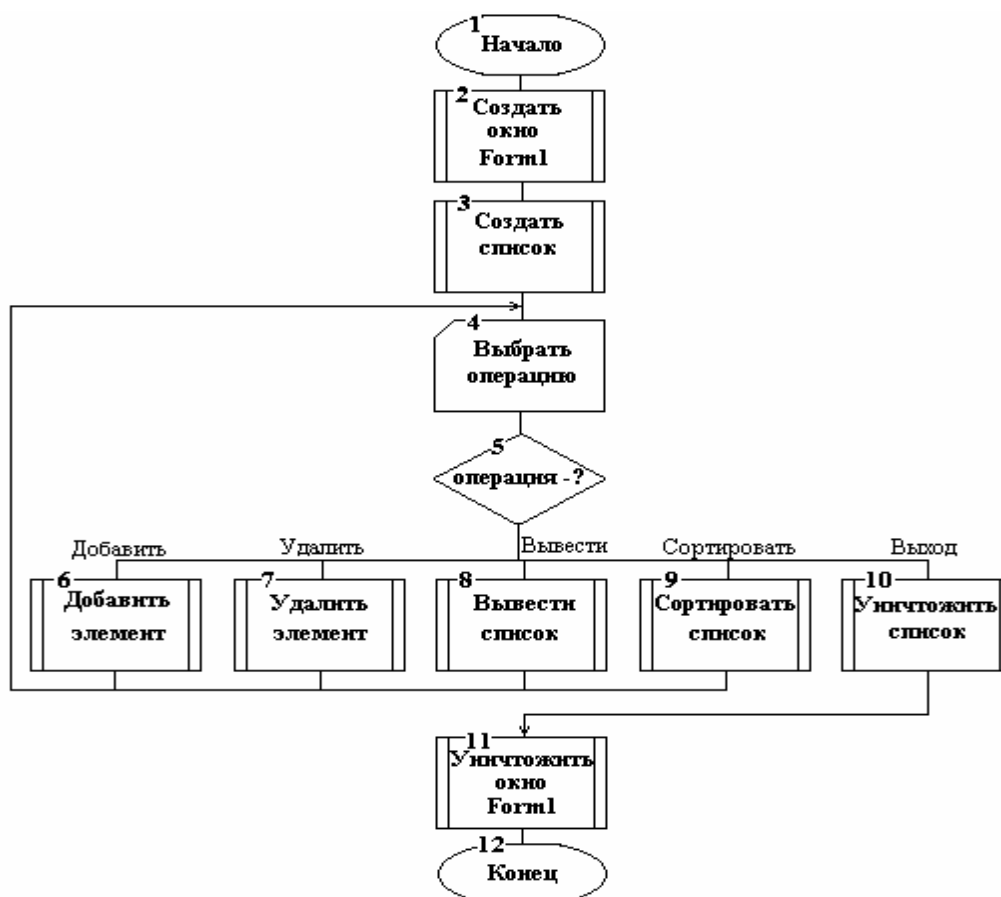


Рисунок 7 - Алгоритм программного продукта

(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 пояснения особенностей реализации классов или специфики событийной обработки можно использовать дополнительные иллюстрации (рисунок 8).

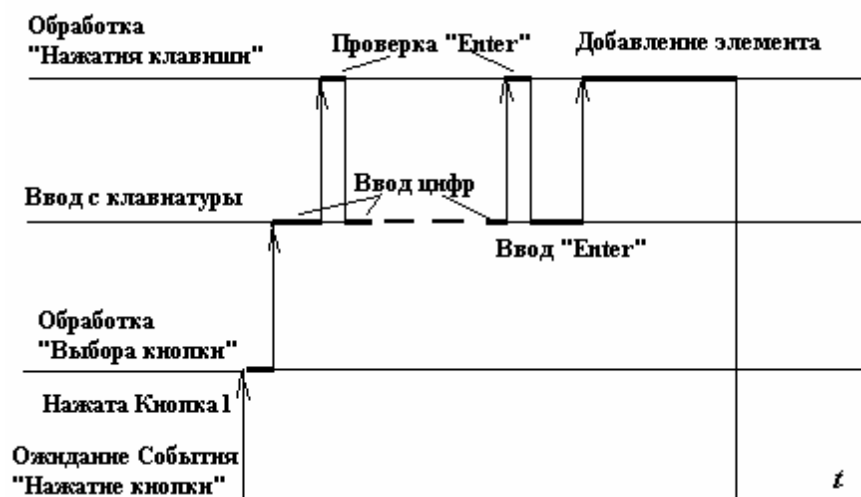


Рисунок 8 - Диаграмма выполнения операции

В завершении раздела реализации описывается декомпозиция разрабатываемой программы на модули и приводится схема взаимодействия модулей (рисунок 9).

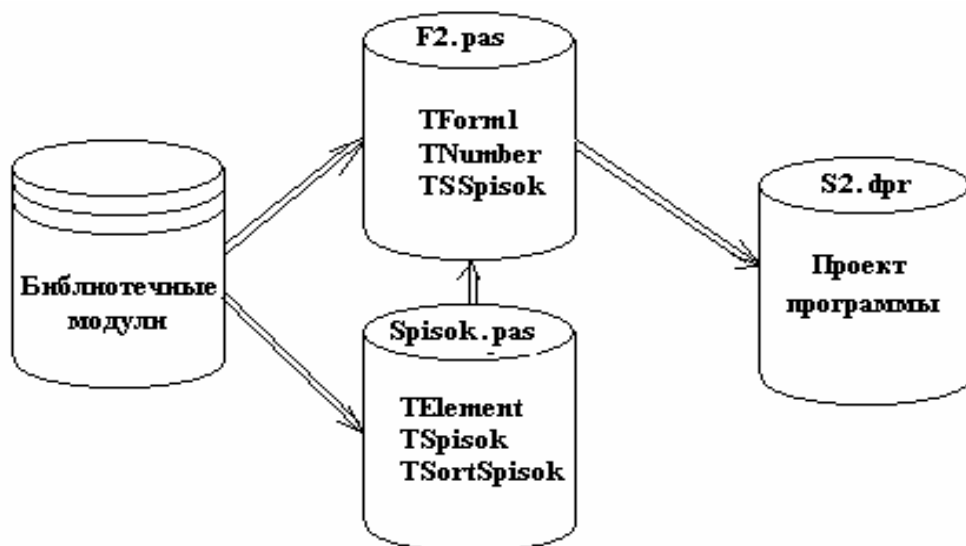


Рисунок 9 - Взаимодействие модулей программного продукта в процессе компиляции и компоновки

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

6) 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 целесообразно представить в виде таблиц.

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

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.

3 Правила оформления пояснительной записки

3.1 Оформление текстового материала

Текст должен быть оформлен в текстовом редакторе Word for Windows версии не ниже 6.0.

Тип шрифта: Times New Roman Cyr. Шрифт основного текста: обычный, размер 14 пт. Шрифт заголовков разделов: полужирный, размер 16 пт. Шрифт заголовков подразделов: полужирный, размер 14 пт.

Межсимвольный интервал: обычный. Межстрочный интервал: одинарный.

Формулы должны быть оформлены в редакторе формул Equation Editor и вставлены в документ как объект.

Размеры шрифта для формул:

- обычный - 14 пт;
- крупный индекс - 10 пт;
- мелкий индекс - 8 пт;
- крупный символ - 20 пт;
- мелкий символ - 14 пт.

Текст дипломной работы выполняется на листах формата А4 без рамки, соблюдая следующие размеры полей: левое - не менее 30 мм, правое - не менее 10 мм, верхнее - не менее 15 мм, нижнее - не менее 20 мм.

Страницы следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему тексту. Номер страницы проставляют в правом верхнем углу без точки в конце.

Опечатки, опiski и графические неточности, обнаруженные в процессе выполнения, допускается исправлять подчисткой или закрашиванием белой краской и нанесением в том же месте исправленного текста машинописным способом или черными чернилами, помарки и следы неполностью удаленного прежнего текста не допускаются.

3.2 Требования к содержанию структурных элементов текстовой части

1) Титульный лист.

Переносы слов в надписях титульного листа не допускаются.

На титульном листе указывают классификационный код (для специальности математическое обеспечение и администрирование информационных систем - 010503). Пример оформления титульного листа курсовой работы приведен в приложении А.

2) Задание.

Задание на курсовую работу должно включать: наименование кафедры, фамилию и инициалы студента, дату выдачи задания, тему работы, исходные

данные и краткое содержание работы, срок представления к защите, фамилии и инициалы руководителя. Задание подписывается руководителем, студентом и утверждается заведующим выпускающей кафедрой.

Задание составляется и утверждается на выпускающей кафедре.

Форма бланка задания приведена в приложении Б.

3) Основная часть.

Текст основной части разделяют на разделы, подразделы, пункты.

Разделы должны иметь порядковые номера в пределах всего текста, обозначенные арабскими цифрами без точки.

Подразделы должны иметь нумерацию в пределах каждого раздела, номера подразделов состоят из номера раздела и подраздела, разделенной точкой. В конце номера подраздела точка не ставится.

Подраздел допускается разбивать на пункты, нумерация которых выполняется аналогично.

Пример 1.2.3 - обозначает раздел 1, подраздел 2, пункт 3

Внутри пунктов или подпунктов могут быть приведены перечисления. Перед каждой позицией перечисления следует ставить дефис или, при необходимости ссылки в тексте на одно из перечислений, строчную букву, после которой ставится скобка. Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа.

Пример:

а) _____

б) _____

1) _____

2) _____

в) _____

Наименования разделов и подразделов должны быть краткими. Наименование разделов и подразделов записывают с абзацного отступа с первой прописной буквы без точки в конце, не подчеркивая. Переносы слов в заголовках не допускаются.

Расстояние между заголовками и текстом должно быть равно 15 мм. Расстояние между заголовками разделов и подраздела - 8 мм. Расстояние между последней строкой текста и последующим заголовком подраздела - 15 мм.

Каждый раздел рекомендуется начинать с нового листа.

Если в пояснительной записке принята особая система сокращения слов или наименований, то в ней должен быть приведен перечень принятых сокращений, который помещают в конце пояснительной записки.

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 которой они приведены в формуле. Первая строка расшифровки должна начинаться со слова «где» без двоеточия после него.

Пример - Статистическое определение среднего значения случайной величины

$$\tilde{M}(x) = \sum_{i=1}^n x_i / n, \quad (2)$$

где x_i — i -е опытное значение случайной величины;
 n —число измерений.

Формулы нумеруются сквозной нумерацией арабскими цифрами, в пределах записки, которые ставят на уровне формулы справа в круглых скобках. Одну формулу обозначают- (1).

Допускается нумерация формул в пределах раздела. В этом случае номер формулы состоит из номера раздела и порядкового номера формулы, разделенных точкой.

Пример - В формуле (3.1)

Ссылки в тексте на номер формулы дают в скобках, например, «...в формуле (1)».

После расшифровки формулы, с новой строки в неё подставляют числовые значения входящих параметров и приводят результат вычисления с обязательным указанием единицы физической величины.

3.3 Оформление иллюстраций

Иллюстрации (рисунки, чертежи, схемы, диаграммы) выполняют на листах пояснительной записки работы или на листах чертежной бумаги формата А4 (210x297 мм) ГОСТ 2.301 карандашом или черной тушью. Разрешается выполнять на печатающих и графических устройствах вывода ЭВМ.

Допускается выполнять иллюстрации на листах формата А3 (297x420 мм).

Иллюстрации располагают после первой ссылки на них.

Допускается помещать иллюстрации вдоль длинной стороны текста с поворотом документа по часовой стрелке для чтения.

Все иллюстрации нумеруют арабскими цифрами сквозной нумерацией. Если один рисунок в тексте, то следует указать «Рисунок 1».

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

При ссылках на иллюстрации следует писать: «...в соответствии с рисунком 2» при сквозной нумерации и «...в соответствии с рисунком 1.2» при нумерации в пределах раздела.

Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисовочный текст). Слово «Рисунок» и наименование помещают после пояснительных данных и располагают следующим образом:

Рисунок 1 - Элементы формы

Пример оформления иллюстрации приведен в приложении Е.

Построение таблиц.

Цифровой материал оформляют в виде таблиц согласно ГОСТ 2.105.

Таблицы следует нумеровать арабскими цифрами сквозной нумерацией. Если в тексте одна таблица, она должна быть обозначена «Таблица 1». Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Пример - Таблица 1.1

Слово «Таблица» и наименование помещают над таблицей следующим образом:

Таблица 1 – Показатели работы транзистора в разных режимах

На все таблицы должны быть ссылки в тексте. При ссылке пишут слово «Таблица» с указанием её номера.

Таблица может иметь заголовки и подзаголовки. Заголовки граф и строк таблицы следует писать с прописной буквы, а подзаголовки - со строчной буквы, если они составляют одно предложение с заголовком.

Графы таблицы допускается нумеровать для облегчения ссылок в тексте, при делении таблицы на части, а также при переносе части таблицы на следующую страницу.

Если таблица не размещается на одном листе, допускается делить её на части. Слово «Таблица» указывают один раз слева над первой частью таблицы, над другими частями пишут слова «Продолжение таблицы» с указанием номера таблицы.

4 Основные стандарты разработки ПО

4.1 Нормативные документы по стандартизации и виды стандартов

Термин «стандарт» (англ. standard — норма) в широком смысле понимается как образец, эталон или модель, принимаемые за исходные для сопоставления с ними других подобных объектов.

Стандарты как нормативно-технические документы устанавливают комплекс норм, правил, требований к объекту стандартизации. Применение стандартов способствует улучшению качества программных средств, повышению развития информатизации процессов, росту эффективности внедрения и эксплуатации программных средств и устраняет разноречивость при создании их различными разработчиками.

В процессе стандартизации вырабатываются нормы, правила, требования, характеристики, касающиеся объекта стандартизации, которые оформляются в виде нормативного документа.

Рассмотрим разновидности нормативных документов, которые рекомендуются руководством 2-й Международной организации по стандартизации и Международной электротехнической комиссии (ИСО/МЭК), а также принятые в государственной системе стандартизации Российской Федерации (РФ). Руководство ИСО/МЭК рекомендует: стандарты, документы технических условий, своды правил, регламенты (технические регламенты), положения (рисунок 10).

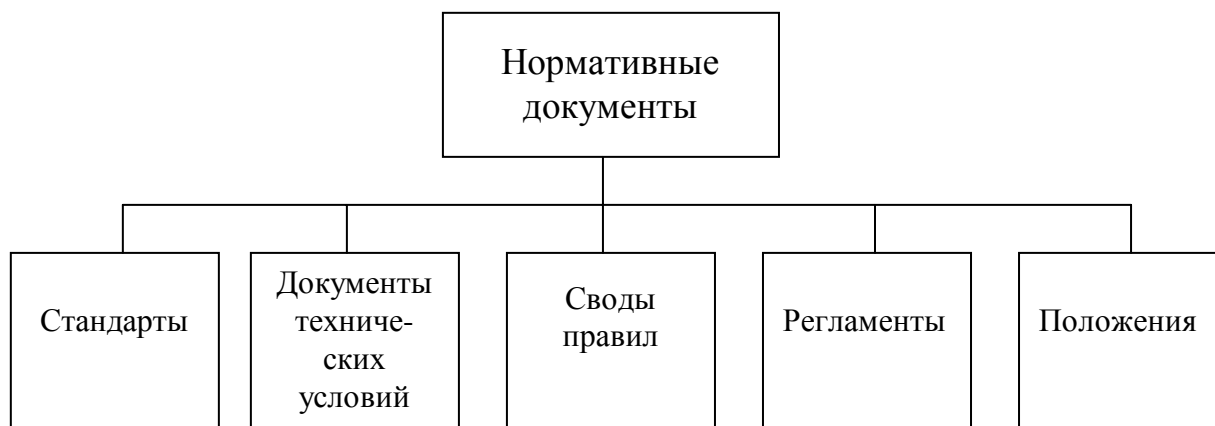


Рисунок 10 – Схема разновидностей нормативных документов

Стандарт как нормативно-технический документ устанавливает комплекс норм, правил, требований к объекту стандартизации. Стандарт может быть разработан как на материальные предметы (продукцию, эталоны, образцы веществ), так и на нормы, правила, требования в различных областях. В переносном смысле — шаблон, трафарет, не содержащий ничего оригинального.

Документ технических условий (technical specification) устанавливает технические требования к продукции, услуге, процессу. Обычно в документе технических условий должны быть указаны методы или процедуры, которые следует использовать для проверки соблюдения требований данного нормативного документа в таких ситуациях, когда это необходимо.

Свод правил, как и предыдущий нормативный документ, может быть самостоятельным стандартом либо самостоятельным документом, а также частью стандарта. Свод правил обычно разрабатывается для процессов проектирования, монтажа оборудования и конструкций, технического обслуживания или эксплуатации объектов, конструкций, изделий. Технические правила, содержащиеся в документе, носят рекомендательный характер.

Все указанные выше нормативные документы являются рекомендательными. В отличие от них регламент носит обязательный характер. *Регламент* — это документ, в котором содержатся обязательные правовые нормы.

Кроме стандартов нормативными документами являются также ПР — правила по стандартизации, Р — рекомендации по стандартизации и ТУ — технические условия. Особое требование предъявляется к нормативным документам на продукцию, которая согласно российскому законодательству подлежит обязательной сертификации. В них должны быть указаны те требования к продукции (услуге), которые подтверждаются посредством сертификации, а также методы контроля (испытаний), которые следует применять для установления соответствия, правила маркировки такой продукции и виды сопроводительной документации.

4.2 Стандарты в области программного, обеспечения

Стандарты занимают все более значительное место в направлении развития индустрии информационных технологий. Более 250 подкомитетов в официальных организациях по стандартизации работают над стандартами в области информационных технологий. Более 1000 стандартов или уже приняты этими организациями, или находятся в процессе разработки. Процесс стандартизации информационных технологий далеко не закончен (да, по нашему мнению, вряд ли когда-либо будет закончен, так как область информационных технологий постоянно динамично развивается).

Все компании-разработчики должны обеспечить приемлемый уровень качества выпускаемого программного обеспечения (ПО). Для этих целей предназначены стандарты качества программного обеспечения или отдельные разделы в стандартах разработки программного обеспечения, посвященные требованиям к качеству программного обеспечения.

Необходимость стандартизации разработки программного обеспечения наиболее удачно описана во введении в стандарт ISO/ IEC 12207: «Программное обеспечение является неотъемлемой частью информационных технологий и традиционных систем, таких, как транспортные, военные, меди-

цинские и финансовые. Имеется множество разнообразных стандартов, процедур, методов, инструментальных средств и типов операционной среды для разработки и управления программным обеспечением. Это разнообразие создает трудности при проектировании и управлении программным обеспечением, особенно при объединении программных продуктов и сервисных программ. Стратегия разработки программного обеспечения требует перехода от этого множества к общему порядку, который позволит специалистам, практикующимся в программном обеспечении, «говорить на одном языке» при разработке и управлении программным обеспечением. Этот международный стандарт обеспечивает такой общий порядок».

Международная организация по стандартизации создана в 1946 г. двадцатью пятью национальными организациями по стандартизации.

При создании организации и выборе ее названия учитывалась необходимость того, чтобы аббревиатура наименования звучала одинаково на всех языках. Для этого было решено использовать греческое слово «isos» — равный. Вот почему на всех языках мира Международная организация по стандартизации имеет краткое название ISO (ИСО).

Сфера деятельности ИСО касается стандартизации во всех областях, кроме электротехники и электроники, относящихся к компетенции Международной электротехнической комиссии (МЭК). Некоторые виды работ выполняются совместными усилиями этих организаций. Кроме стандартизации ИСО занимается и проблемами сертификации.

Согласно руководству ИСО/МЭК деятельность по стандартизации осуществляют соответствующие органы и организации.

Национальным органом по стандартизации в России является Государственный комитет Российской Федерации по стандартизации и метрологии (Госстандарт России). Это федеральный орган исполнительной власти, осуществляющий межотраслевую координацию, а также функциональное регулирование в области стандартизации, метрологии и сертификации.

В России в качестве базового принят стандарт ГОСТ Р ИСО/МЭК 12207. Данный документ введен в действие с 1 июля 2000 г., тесно взаимосвязан с рядом стандартов, принятых ранее, и с некоторыми стандартами, разрабатываемыми в данное время на основе прямого применения стандартов ИСО.

Актуальность стандарта ГОСТ Р ИСО/МЭК 12207 для современных условий настолько высока, что принятие в ISO его исходного, международного варианта вскоре вызвало самую положительную оценку российских экспертов. Был дан ряд рекомендаций по его использованию в реальных условиях.

В России государственные стандарты на программную документацию — это комплекс стандартов 19-й серии ГОСТ ЕСПД (Единой системы программной документации — серия ГОСТ 19.XXX), который отвечает на вопросы:

– Что должно быть сделано, кроме собственно программы?

- Что и как должно быть оформлено в виде документации?
- Что передавать пользователям, а что — службе сопровождения?
- Как управлять всем этим процессом?
- Что должно входить в само задание на программирование?

Стандарты ЕСПД в основном охватывают ту часть документации, которая создается в процессе разработки ПС, и связаны, по большей части, с документированием функциональных характеристик ПС. Следует отметить, что стандарты ЕСПД (ГОСТ 19) носят рекомендательный характер. Впрочем, это относится и ко всем другим стандартам в области ПС (ГОСТ 34, международному стандарту ISO/IEC и др.). Дело в том, что в соответствии с Законом РФ «О стандартизации» эти стандарты становятся обязательными на контрактной основе, т.е. при ссылке на них в договоре на разработку (поставку) ПС.

Обозначение стандарта ЕСПД должно состоять из:

- числа 19 (присвоенных классу стандартов ЕСПД);
- одной цифры (после точки), обозначающей код классификационной группы стандартов, указанной в таблице;
- двузначного числа (после тире), указывающего год регистрации стандарта.

Вообще перечень документов ЕСПД очень обширен. В него, в частности, входят следующие ГОСТы:

ГОСТ 19.001-77 ЕСПД. Общие положения.

ГОСТ 19.005-85 ЕСПД. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения.

ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.

ГОСТ 19.102-77 ЕСПД. Стадии разработки.

ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.

ГОСТ 19.104-78 ЕСПД. Основные надписи.

ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.

ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом.

ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.

ГОСТ 19.202-78 ЕСПД. Спецификация. Требования к содержанию и оформлению.

ГОСТ 19.301-79 ЕСПД. Порядок и методика испытаний.

ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.

ГОСТ 19.402-78 ЕСПД. Описание программы.

ГОСТ 19.403-79 ЕСПД. Ведомость держателей подлинников.

ГОСТ 19.404-79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.

ГОСТ 19.501-78 ЕСПД. Формуляр. Требования к содержанию и оформлению.

ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению.

ГОСТ 19.503-79 ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.

ГОСТ 19.504-79 ЕСПД. Руководство программиста. Требования к содержанию и оформлению.

ГОСТ 19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению.

ГОСТ 19.506-79 ЕСПД. Описание языка. Требования к содержанию и оформлению.

ГОСТ 19.507-79 ЕСПД. Ведомость эксплуатационных документов.

ГОСТ 19.508-79 ЕСПД. Руководство по техническому обслуживанию. Требования к содержанию и оформлению.

ГОСТ 19.601-78 ЕСПД. Общие правила дублирования, учета и хранения.

ГОСТ 19.602-78 ЕСПД. Правила дублирования, учета и хранения программных документов, выполненных печатным образом.

ГОСТ 19.603-78 ЕСПД. Общие правила внесения изменений.

ГОСТ 19.604-78 ЕСПД. Правила внесения изменений в программные документы, выполняемые печатным способом.

ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения.