

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА АВТОМАТИКИ И КОМПЬЮТЕРНЫХ СИСТЕМ

Курсовое проектирование

Разработка программного обеспечения

**Методические указания к выполнению курсовых проектов
для студентов специальностей**

**210100 «Управление и информатика в технических системах» и
220400 «Программное обеспечение вычислительной техники
и автоматизированных систем»**

**Сургут
2004**

Курсовое проектирование. Разработка программного обеспечения.

Методические указания к выполнению курсовых проектов для студентов специальностей 210100 «Управление и информатика в технических системах» и 220400 «Программное обеспечение вычислительной техники и автоматизированных систем». / Сост.: П.В. Гришмановский, И.Н. Даниленко. – Сургут: Издательство СурГУ, 2004. – 23 с.

Методические указания разработаны в соответствии с учебными планами специальностей подготовки дипломированного специалиста 210100 «Управление и информатика в технических системах» и 220400 «Программное обеспечение вычислительной техники и автоматизированных систем».

Рассмотрены вопросы организации курсового проектирования, этапы жизненного цикла программного обеспечения и их содержание, общие положения выполнения, представления и защиты курсового проекта, а также оформления пояснительной записки.

Предназначены для студентов 3–5 курсов Инженерно-физического факультета.

Табл. 3.

Печатается по решению редакционно-издательского совета Сургутского государственного университета.

Составители: доцент кафедры Автоматики и компьютерных систем, к.т.н. Гришмановский П.В., ассистент кафедры Автоматики и компьютерных систем, к.т.н. Даниленко И.Н.

Рецензент: доцент кафедры Автоматики и компьютерных систем, к.т.н. Тараканов Д.В.

© Сургутский государственный университет, 2004.

Содержание

Введение.....	4
Организация курсового проектирования	4
Этапы разработки программного продукта	6
Анализ	7
Проектирование.....	8
Кодирование и отладка	8
Тестирование	9
Сопровождение.....	10
Стратегии разработки программного обеспечения.....	10
Однократный проход	11
Инкрементная разработка.....	11
Содержание пояснительной записки	13
Графическая часть пояснительной записи	14
Оформление пояснительной записки	15
Стиль изложения	15
Нумерация страниц	16
Заголовки разделов	16
Приложения	16
Нумерация объектов	17
Использованная литература	18
Оформление листингов.....	18
Оформление содержания.....	19
Параметры макета	19
Вопросы для самопроверки	Ошибка! Закладка не определена.
Приложение. Пример оформления титульного листа	22

Введение

Настоящий документ регламентирует общие положения выполнения, оформления, представления и защиты курсовых проектов по общепрофессиональным и специальным дисциплинам, а также курсовых проектов по специальности, выполняемых в рамках учебных планов технических специальностей и связанных с разработкой программного обеспечения (ПО).

Целью курсового проектирования является закрепление и углубление теоретических знаний и приобретение практических умений и навыков по изучаемой дисциплине или блоку дисциплин в процессе разработки ПО для заданной предметной области. Кроме того, выполнение курсового проекта по специальности требует знаний в какой-либо предметной области, соответствующей циклам общепрофессиональных и специальных дисциплин.

В задачи курсового проектирования входят:

- изучение особенностей конкретной предметной области, относящейся к теме курсового проекта;
- анализ возможных подходов и методов решения задачи с обоснованием выбранного подхода;
- выбор или разработка математических, структурных, информационных и других моделей, необходимых для решения задачи;
- выбор или разработка алгоритмов с учетом их точности, устойчивости, сходимости, адекватности, эффективности и т.д.;
- реализация программного кода и его отладка;
- тестирование разработанного ПО;
- анализ полученных результатов работы ПО и оценка его характеристики;
- оформление технической документации на разработанное ПО.

Организация курсового проектирования

Основанием для разработки программного продукта (ПП) является задание на курсовое проектирование, выданное руководителем. В целом, процесс выполнения курсового проекта регламентируется общими требованиями к разработке, которыми являются настоящие методические указания, и специфическими для данного курсового проекта требованиями, предъявленными руководителем.

Выполнение курсового проекта условно можно разделить на работы двух направлений – собственно разработка ПО и оформление пояснительной записки, – которые завершаются защитой проекта. Рекомендуется выполнять указанные выше виды работ параллельно, а принимаемые на каждом этапе решения и получаемые результаты согласовывать с руководителем. При этом следует придерживаться общего графика выполнения курсового проекта, представленного в табл. 1, либо графика, предложенного руководителем. Содержание этапа «Разработка» определяется спецификой курсового проекта в каждом конкретном случае, а также выбранной стратегией разработки ПО (см. раздел «Стратегии разработки программного обеспечения»).

Таблица 1

График выполнения курсового проекта

№	Этап работы	Сроки выполнения (по неделям учебно- го семестра)		Длитель- ность (недель)
		осенний семестр	весенний семестр	
1	Выдача задания, анализ, обзор	1–3	1–2	2–3
2	Разработка	4–13	3–12	10
3	Оформление пояснительной за- писки, представление к защите, исправление и доработка	14–16	13–15	3
4	Защита курсового проекта	17–18	16–17	2

Для предварительной проверки руководителю предоставляется законченная пояснительная записка (с разрешения руководителя – не подшитая), выполненная в соответствии с настоящими методическими указаниями и специфическими требованиями к ее содержанию и оформлению. К пояснительной записке прилагается подписанная (группа, фамилия, инициалы) дискета или компакт-диск с файлами, необходимыми для компиляции и тестирования разработанного ПО. К таковым относятся файлы проекта и исходного кода разработанного программного продукта, а также файлы проекта приложения для его тестирования, если разработанный ПП, в соответствии с заданием, не является самостоятельным приложением. Кроме того, должны быть представлены файлы данных, необходимые для работы программного продукта или его тестирования, если таковые имеются. Особые рекомендации по компиляции, сборке, установке, размещению файлов исходного кода и данных, тестированию и т.п. приводятся в текстовом файле (рекомендуемое имя – *readme.txt*).

Установка и тестирование ПО с целью проверки результатов курсового проектирования осуществляется руководителем на основании задания и раздела «Руководство пользователя» пояснительной записки. При тестировании, в общем случае, основными критериями оценки являются соответствие заданию, устойчивость, адекватность, эффективность реализации и потребительские качества, но, в силу специфических особенностей темы курсового проекта и его предметной области, более высокий приоритет могут иметь отдельные из указанных или другие критерии. Также, если это предусматривается данным курсовым проектом, оцениваются представленное руководство пользователя, качество тестирования ПП разработчиком и независимым экспертом.

При проверке пояснительной записки основными критериями являются: обоснованность принятых решений на всех этапах разработки, четкость и стиль изложения материала. Обращается внимание на оформление и соблюдение правил языка, использованного при написании текста пояснительной записки.

После проверки, пояснительная записка с замечаниями, указанными руководителем, возвращается исполнителю для доработки или подготовке к защите. При защите курсового проекта должна быть представлена подшитая пояснительная записка. В случае, если при проверке руководитель выписывает замечания на обороте титульного листа или по тексту, все листы с замечаниями должны быть сохранены и предоставлены при следующей проверке или при защите курсового проекта. В случае повторных проверок пояснительной записи руководитель может сделать вывод о нецелесообразности дальнейшей ее доработки и, как следствие, либо о неудовлетворительном выполнении курсового проекта, либо о его представлении к защите. При несоблюдении установленных сроков выполнения и защиты курсового проекта считается, что он выполнен с оценкой «неудовлетворительно». При неудовлетворительной защите курсового проекта, его повторная проверка и защита возможна только при наличии разрешения деканата.

Этапы разработки программного продукта

Программным продуктом, который требуется разработать в рамках курсового проекта, может быть не только приложение (исполняемая программа), но и отдельные визуальные компоненты, классы и функции, поставляемые в виде динамических или статических библиотек кода. Тем не менее, при разработке ПО необходимо выполнение всех

этапов жизненного цикла (ЖЦ) ПО вне зависимости от его типа, используемой модели ЖЦ и стратегии разработки.

Анализ

Задание на курсовое проектирование, выданное руководителем, нуждается во всестороннем анализе. Основными целями этапа анализа являются разрешение противоречий и неточностей задания, определение функциональных требований, которым должен соответствовать ПП, множеств входной и выходной информации, типа интерфейса пользователя и т.п. Все дополнения к исходному заданию и его изменения должны быть согласованы с руководителем и фиксируются в тексте пояснительной записи. На основе проделанной работы формируется описание задачи, которое должно быть полным, последовательным, доступным для чтения и обзора различными заинтересованными сторонами. Таким образом, в задачи анализа входят:

- анализ предметной области данной задачи;
- обзор существующих решений подобных задач;
- обобщение частной задачи;
- уточнение требований, приведенных в задании на проектирование;
- разработка спецификаций на ПО.

Результатом выполнения этого этапа является набор спецификаций, содержащих конкретное описание разрабатываемого программного продукта. По степени формализации выделяют:

- словесные спецификации на естественном языке;
- модельные спецификации (схемы, диаграммы и другие информационно-графические представления);
- формальные спецификации (полученные формальным способом с использованием математических аппаратов, которые обеспечивают полное определение семантики).

Кроме того, в зависимости от семантической нагрузки, выделяют функциональные и эксплуатационные спецификации. Функциональные спецификации определяют функции ПП. Эксплуатационные спецификации должны содержать сведения о быстродействии ПО, затратах памяти, требуемых технических средствах, надежности и т.д. Спецификации должны быть полными, точными и ясными.

В ходе анализа и при выполнении последующих этапов, необходимо преследовать цель универсальности и унификации разрабатываемого ПП. Это предполагает использование стандартных библиотек функций и классов, понятных обозначений и имен, отвечающих смыслу объектов, общепринятых реализаций структур данных, алгоритмов, форм ввода и визуализации данных, стандартного (привычного для

пользователя) интерфейса, широко распространенных форматов файлов, унифициацию программных интерфейсов и т.п., если это не противоречит требованиям к разрабатываемому ПО.

Проектирование

Основной целью этого этапа является получение проектных решений, на основе которых может быть осуществлена реализация программного продукта. К сожалению, не существует однозначной формы представления результатов проектных решений – в зависимости от предметной области и целевой направленности курсового проектирования это могут быть, например, математические выражения, диаграммы, характеризующие структуру модулей разрабатываемого ПО, диаграммы иерархии классов и объектов, диаграммы взаимодействия, информационные модели, блок-схемы алгоритмов, спецификации программ и т.д. Вне зависимости от используемой модели ЖЦ, основными методологическими подходами в проектировании являются иерархическая декомпозиция сложной задачи создания ПО на подзадачи меньшей сложности и отыскание обобщенных решений, обеспечивающих гибкость последующей реализации и устойчивость процесса к модификациям ПП.

На этапе проектирования решаются следующие задачи:

- формирование структуры ПО и разработка алгоритмов, задаваемых спецификациями;
- определение состава модулей с разделением их на иерархические уровни;
- разработка структур данных и форматов их представления;
- выбор структуры информации в базе данных;
- фиксация межмодульных и других программных интерфейсов;
- разработка интерфейса пользователя, форматов ввода, вывода и визуализации информации.

Необходимо отметить, что в целях получения обобщенного решения, проектирование выполняется, как правило, без учета языка программирования, среды разработки, конкретных типов данных и библиотек.

Кодирование и отладка

Целью этапа кодирования и отладки является программная реализация проектных решений, полученных на предыдущем этапе, – создание работоспособного программного продукта. Если в задании не оговаривается язык программирования и среда разработки, с помощью которых должны осуществляться кодирование и отладка, то этот вопрос решается на этом этапе с учетом результатов анализа и проектирования.

В процессе программной реализации производится аргументированный выбор типов данных, классов, функций прикладного интерфейса, библиотек и т.п. При этом необходимо во всех случаях, когда это возможно, использовать те типы данных, которые применяются в библиотеках, используемых при разработке, а также учитывать существующие соглашения при формировании собственных идентификаторов и ограничения, действующие в этих библиотеках. Все принятые решения должны быть рациональны, эффективны и обоснованы с позиций производительности, объема кода и данных, функциональности ПП, либо его универсальности. Кроме того, все несоответствия реализации тем решениям, которые были приняты на этапах анализа и проектирования, должны быть обоснованы с точки зрения принципиальной невозможности или чрезвычайной сложности реализации, особенностей среды разработки, языка программирования, операционной системы.

Тестирование

Целями этапа тестирования являются устранение ошибок в разработанном ПО, выявление его соответствия или несоответствия заданию на проектирование и разработанным спецификациям, проверка адекватности и устойчивости функционирования ПП как в штатных, так и в нештатных ситуациях. В ходе этого этапа решаются следующие задачи:

- планирование тестирования;
- разработка тестов;
- подготовка данных для тестирования;
- испытание ПО.

Результатом работы на данном этапе является ПО, готовое к эксплуатации.

В случае, если программный продукт не является самостоятельным приложением (функции, классы, визуальные компоненты и т.п.), необходимо учитывать его специфику. Например, при разработке визуального компонента необходимо убедиться в адекватности его функционирования как во время проектирования, так и во время выполнения, во всех возможных режимах функционирования провести проверку всех новых свойств и методов, а также унаследованных, прямо или косвенно влияющих на состояние или поведение объекта и т.п.

Кроме внутреннего тестирования, проводимого самим разработчиком, в рамках данного этапа может выполняться и независимое тестирование. Независимое тестирование программного продукта проводится экспертом, назначенным руководителем. Для независимого тестирования разработчик предоставляет эксперту только готовый ПП,

задание на его разработку и руководство пользователя. Эксперт должен самостоятельно провести тестирование представленного ему ПО в полном объеме и составить акт независимого тестирования, который передается разработчику для доработки программного продукта (при необходимости). В акте независимого тестирования приводятся этапы и схема тестирования, обнаруженные недостатки и заключение о соответствии или несоответствии заданию. Разработчик может счесть тестирование, проведенное экспертом, неполным и, аргументировав причину, потребовать повторного тестирования.

Сопровождение

Так как разработка программного обеспечения в рамках курсового проектирования имеет учебный характер, то сопровождение ПП сводится к его документированию и разработке руководства пользователя.

Руководство пользователя представляет собой документ, которым должен комплектоваться готовый программный продукт при его передаче конечному пользователю. Руководство пользователя должно быть полным и достаточным для освоения и эффективного использования разработанного программного продукта и, как правило, содержит:

- требования к программно-аппаратной платформе, на которой должен функционировать ПП;
- инструкцию по установке и настройке;
- правила эксплуатации;
- сведения обо всех обнаруженных недостатках программного продукта, которые были обнаружены, но не исправлены;
- предупреждения о возможном неадекватном функционировании ПП, вызванном его особенностями или неправильной эксплуатацией, с указанием ситуаций, в которых оно наблюдается.

Примером руководства пользователя могут служить справочные системы ПО, подобного по назначению, описания функций, классов, компонентов и библиотек в справочных системах сред программирования, спецификации протоколов и форматов файлов и т.п.

Стратегии разработки программного обеспечения

Тип разрабатываемого программного обеспечения, его ожидаемые свойства и особые требования к программному продукту и к процессу его разработки, могут существенно влиять на выбор стратегии разработки, обеспечивающей наиболее эффективное решение поставленной задачи. Важно, что при любой стратегии разработки необходимо выполнение всех этапов жизненного цикла программного обеспечения,

но их последовательность и распределение времени может различаться. Собственно выбор стратегии осуществляется на этапе анализа.

В тех задачах, когда требования к разрабатываемому ПО известны или уточняются в начале разработки, что свойственно для курсового проектирования, наиболее применимы стратегии однократного прохода и инкрементной разработки, вне зависимости от используемого подхода – процедурного или объектно-ориентированного.

Однократный проход

Данная стратегия основана на классической (называемой также каскадной) модели жизненного цикла, при которой все этапы жизненного цикла выполняются строго последовательно, а переход к следующему производится только при полном завершении предыдущего.

При использовании стратегии однократного прохода в курсовом проектировании следует придерживаться сроков выполнения этапов разработки, указанных в табл. 2.

Таблица 2
График выполнения этапов разработки при стратегии однократного прохода

№	Этап разработки программного продукта	Сроки выполнения (по неделям учебного семестра)		Длительность (недель)
		осенний семестр	весенний семестр	
1	Проектирование	4–7	3–6	4
2	Кодирование и отладка	8–10	7–9	3
3	Тестирование (в т.ч. независимое)	11–12	10–11	2
4	Сопровождение	13	12	1

Достоинством данной стратегии является хорошая детерминированность и формализованность процесса разработки, его хорошая планируемость и наблюдаемость, но существенным недостатком является низкая адаптивность к изменениям требований, что, в случае курсового проекта, не столь существенно. Как правило, данная стратегия используется совместно с процедурно-ориентированными средствами проектирования и реализации ПО.

Инкрементная разработка

Стратегия инкрементной разработки (запланированного улучшения ПП) основана на спиральной модели жизненного цикла ПО, которая разрушает монотонность отдельных этапов каскадной модели и их стро-

гую последовательность. При этом все требования к разрабатываемому программному продукту выявляются на этапе анализа, а остальные этапы выполняются в виде последовательности итераций (инкрементов), на каждой из которых создается ПП, характеристики которого постепенно приближаются к требуемым. В результате последнего инкремента будет получен готовый программный продукт.

При использовании инкрементной стратегии разработки в рамках курсового проектирования рекомендуется руководствоваться графиком выполнения этапов, представленным в табл. 3.

Таблица 3
График выполнения этапов инкрементной разработки

№	Этап разработки программного продукта	Сроки выполнения (по неделям учебного семестра)		Длительность (недель)
		осенний семестр	весенний семестр	
1	Первый инкремент (включая проектирование архитектуры, структуры, алгоритмов и др.)	4–6	3–5	3
2	Второй инкремент	7–8	6–7	2
3	Третий инкремент	9–10	8–9	2
4	Четвертый инкремент (включая внутреннее и независимое тестирование)	11–13	10–12	3

Инкрементная стратегия эффективна, как правило, при объектно-ориентированном подходе (ООП) к разработке ПО на всех этапах ЖЦ. Цели и задачи этапов жизненного цикла при объектно-ориентированной разработке те же, что и рассмотренные выше, однако основное внимание уделяется объектной, а не функциональной, декомпозиции задачи.

Итогом выполнения объектно-ориентированного анализа является идентификация и моделирование основных классов и объектов, логических отношений и взаимодействий между ними. Полученная структура модифицируется на этапе объектно-ориентированного проектирования за счет ввода новых абстракций и механизмов, которые позволяют более эффективно использовать уже идентифицированные классы и объекты. Декомпозиция производится до уровня достаточно простых ключевых абстракций, не требующих дальнейшей декомпозиции. Как правило, используются и объектно-ориентированные средства реализации или, в некоторых случаях, процедурные, но в сочетании с принципами ООП.

Эволюция ПО при данной стратегии совмещает в каждом инкременте традиционные этапы: проектирование, кодирование, интеграция, тестирование и документирование. Таким образом, процесс разработки превращается в постепенное составление ряда прототипов, которые затем входят в конечную реализацию. Составление развивающихся прототипов, в свою очередь, стимулирует разработку и оценку альтернативных решений, что позволяет получить разумный компромисс, удовлетворяющий предъявленным ограничениям и функциональным требованиям.

Содержание пояснительной записи

В пояснительной записке к курсовому проекту, являющейся техническим документом, должен быть представлен процесс разработки программного продукта в соответствии с этапами жизненного цикла. Выполненный анализ, проектирование, кодирование и тестирование представляют собой собственно процесс разработки ПО и, как правило, в тексте пояснительной записи освещены в отдельных разделах вне зависимости от использованной стратегии разработки. Этап сопровождения ПП должен быть представлен документацией (руководством пользователя) на разработанный программный продукт. Необходимые графические и справочные материалы, нормативно-техническая документация, листинги программного кода и т.п., необходимые для изложения материала разделов, в зависимости от их объема, размещаются либо в соответствующих разделах, либо выносятся в приложения.

С учетом требований, предъявляемых к пояснительной записи в частности и к технической документации в целом, ее структура должна быть следующей (заголовки соответствующих разделов должны быть сформулированы согласно заданию на курсовое проектирование, его специфики и содержанию материала разделов):

1. Титульный лист.
2. Аннотация – краткая характеристика пояснительной записи как документа.
3. Задание – оригинальный (исходный) текст задания либо, если это предусмотрено спецификой курсового проекта, оригинальный бланк задания, выданный руководителем; изменение текста задания и замена оригинального бланка задания на иной недопустимы.
4. Содержание.
5. Введение – обозначает предметную область задачи, ее актуальность, ожидаемые результаты и их применение.
6. Анализ.

7. Проектирование.
8. Кодирование и отладка.
9. Тестирование – описание процесса тестирования, заключение о соответствии программного продукта заданным требованиям и его недостатках, описание предпринятых мер по устранению обнаруженных недостатков.
10. Сопровождение – документация на разработанный программный продукт, руководство пользователя. Руководство пользователя может быть также вынесено в отдельное приложение с целью сохранения его внутренней структуры, отличной от структуры пояснительной записи в целом, при этом данный раздел пояснительной записи может отсутствовать, но разделы «Кодирование и отладка» или «Тестирование» должны содержать ссылку на соответствующее приложение.
11. Заключение – обозначается результат разработки, возможности его применения, достоинства и недостатки полученного решения, перспективы дальнейшего развития данной задачи, направления расширения и улучшения функциональных и эксплуатационных характеристик ПП.
12. Список использованной литературы – упорядоченный перечень библиографических описаний документальных источников информации.
13. Приложения – полные листинги файлов исходного кода программного продукта и тестового приложения, если ПП не является самостоятельным приложением, акт независимого тестирования, если это предусмотрено спецификой данного курсового проекта, нормативно-техническая документация, иллюстративный материал.

Графическая часть пояснительной записи

Графическую часть пояснительной записи составляет иллюстративный материал (графики, диаграммы, схемы, рисунки), необходимый для изложения и более полного понимания текста пояснительной записи. В графической части курсового проекта могут быть представлены следующие результаты:

- функциональная структура ПО, показывающая функциональное назначение всей программы и ее отдельных частей;
- модульная (иерархическая) структура ПО, фиксирующая результаты проектирования ПО;
- диаграммы иерархии классов и объектов, диаграммы взаимодействия, диаграммы переходов и состояний и т.п., фиксирующие результаты объектно-ориентированного проектирования ПО;

- информационные модели баз данных;
- блок-схемы алгоритмов, иллюстрирующие работу основных функций и методов, реализуемых в ПП;
- результаты работы ПО в штатном режиме в форме графиков, таблиц, примеров выходных данных;
- экранные формы, демонстрирующие особенности пользовательского интерфейса и т. п.

Графический материал данного раздела может быть вынесен в приложения к пояснительной записке.

Оформление пояснительной записи

Пояснительная записка к курсовому проекту является техническим документом и должна быть оформлена соответствующим образом. Это следует учитывать и при использовании методических изданий по выполнению курсовых проектов и оформлению пояснительных записок, имеющихся в библиотеке СурГУ. В любом случае, в первую очередь необходимо обратить внимание на ряд отмеченных ниже аспектов подачи и оформления материала.

Стиль изложения

Технический документ предполагает использование соответствующего стиля изложения. Материал подается в безличной форме и без обращений к читателю. Таким образом, фразы, подобные «я выбрал», «мы решили» и т.п., являются недопустимыми и должны быть построены иначе, например, «выбран» или «был выбран», «принято решение» и т.д. Кроме того, особое внимание необходимо уделять терминологии – все используемые термины должны быть либо общепринятыми в данной предметной области, либо их необходимо пояснить при первом использовании. Недопустимо вводить новые термины вместо существующих общепринятых. В любом случае, используемые термины должны быть недвусмысленны. Все аббревиатуры и сокращения, кроме общепринятых, либо должны быть расшифрованы при первом их упоминании, либо их расшифровки должны быть вынесены в отдельный раздел «Список используемых сокращений», который размещается непосредственно за титульным листом. Термины, аббревиатуры и сокращения должны иметь один и тот же смысл в пределах всей пояснительной записи, иные случаи являются нежелательными и оговариваются отдельно. В целом, построение фраз и предложений, расстановка знаков препинания, кавычек, скобок, оформление перечислений и т.д. должны подчиняться правилам языка, используемо-

го для написания текста пояснительной записи (для России таковыми являются русский, как официальный государственный язык, и английский – официальный язык международного общения). Пояснительные записи, текст которых представлен на неустановленном языке, не могут быть приняты к рассмотрению.

Нумерация страниц

В пределах всей пояснительной записи должна быть использована сплошная нумерация страниц, включая и все страницы приложений (вне зависимости от формата бумаги). Страницей 1 является титульный лист, но на титульном листе, в аннотации и задании номера страниц не проставляются. Рекомендуемое положение номера – внизу страницы у внешней границы размещения текста, внутри поля.

Заголовки разделов

Для упрощения ориентации в документе и ссылок в тексте на материал других его разделов, используется сплошная нумерация заголовков разделов арабскими цифрами начиная с 1. В конце заголовка точка не ставится. Нумеруются либо все заголовки, начиная с задания, либо только заголовки основной части (разделы между «Введение» и «Заключение»). В каждом разделе могут быть выделены подразделы (параграфы), которые, в свою очередь, могут быть разбиты на пункты и т.д. Оптимальным считается наличие в документе заголовков 3-х уровней, но, при необходимости, количество уровней может быть иным. Заголовки подразделов должны иметь составную нумерацию, повторяющую номер заголовка более высокого уровня и дополняющую ее номером данного параграфа в разделе или пункта в параграфе, например, «6. Программная реализация визуального компонента ...», «6.1. Выбор базового класса ...», «6.1.1. Основные требования ...», «6.1.2. Назначение методов ...» и т.д. В тексте пояснительной записи могут присутствовать ссылки на другие ее разделы, например «... подробно изложены в п. 6.1.1 ...», «... описано в п. 7».

Приложения

Приложения нумеруются аналогично разделам, но с использованием слова «Приложение» и заглавных букв русского алфавита, и также могут содержать подзаголовки, например, «Приложение В. Листинги программного кода ...», «В.1. Заголовочный файл ...», «В.2. Файл кода модуля ...», «Приложение Г. Акт независимого тестирования ...» и т.п. В тексте основной части пояснительной записи должны присутствовать ссылки на материал приложений, в противном случае считается,

что этот материал не используется и размещен в приложения необоснованно или для фиктивного увеличения объема пояснительной записи. Примеры ссылок на материал приложений: «... существенно различаются по своей организации (Приложение А).», «... приведен в Приложении В.1, В.2 ...». Следует учесть, что в документах, размещенных в приложениях, может существовать своя структура заголовков. В этом случае не требуется изменять ее в соответствии со стилем оформления пояснительной записи. Тем не менее, возможны ссылки на разделы такого документа, например «... обнаружены серьезные недостатки (см. Приложение Г, п. 4.2), которые ...». Материал в приложениях группируется тематически, а сами приложения размещаются в конце пояснительной записи в порядке их упоминания в тексте основной части.

Нумерация объектов

К объектам, размещаемым в тексте пояснительной записи, относятся рисунки, таблицы, листинги и формулы. Все эти объекты, за исключением формул, должны быть пронумерованы и подписаны. При этом рекомендуется использовать составную нумерацию, включающую в себя номер раздела и порядковый номер объекта данного типа в этом разделе, например, «Таблица 4.1. Сравнительные данные ...», «Рисунок 5.3. Формат исходных данных», «Листинг 6.2. Пример перегрузки операции ...». Поясняющие надписи (названия) рисунков и листингов помещаются в одной строке с их нумерацией (как в примере выше). Для таблиц надпись «Таблица ...» без точки на конце, например, «Таблица 4.1», размещается в отдельной строке и выравнивается по правому краю, а название таблицы без точки на конце, например, «Сравнительные данные ...», – в следующей строке по центру. В тексте могут быть размещены ссылки на перечисленные объекты в виде «табл. 4.1», «рис. 5.3», «листинг 6.2», являясь частью фразы (например, «... приведены в табл. 4.1.») или дополняя ее (например, «... должен быть использован определенный формат (рис. 5.3), обеспечивающий ...»).

Рисунки, таблицы и листинги в тексте пояснительной записи размещаются после первого их упоминания (ссылки) на той же странице или, в крайнем случае, на следующей, но не далее. При большом объеме объекта (более 2/3 площади страницы), его желательно выносить в приложения. Ссылки на объекты, размещенные в приложениях, оформляются аналогично, например, «... как показано на рис. А.5, необходимо ...», «... выбраны в соответствии со справочными данными (табл. Б.4).». Любой объект, не упомянутый в тексте основных разделов, считается излишним и должен быть исключен из состава пояснительной записи.

Формулы размещаются непосредственно по тексту и либо не нумеруются, при отсутствии ссылок на них, либо номер размещается в круглых скобках на уровне формулы у правой границы размещения текста на странице. Рекомендуется составная нумерация, как и для других объектов. Ссылка на формулу представляет собой ее номер, заключенный в круглые скобки, и является частью фразы, например «... тогда из (5.3) может быть получено (5.4) и (5.5), что удовлетворяет заданному условию (5.1) ...».

Использованная литература

На всех этапах разработки ПП используются различные источники информации справочного, научного, учебно-методического, публицистического и др. характера с целью уточнения исходной информации или обоснования приводимых решений и выводов. Список этих источников приводится в разделе «Использованная литература» в общепринятой форме. При использовании информации на электронных носителях, их выходные данные указываются аналогично печатным изданиям, для Интернет-страниц указываются автор и название, а также адрес размещения в сети Интернет. Список источников информации формируется либо в порядке упоминания источников в тексте, либо по алфавиту. Для списка используется сплошная нумерация арабскими цифрами, начиная с 1. В тексте пояснительной записи должны присутствовать ссылки на указанные источники в тех местах, где непосредственно использован или заимствован их материал. Ссылки оформляются в виде номера источника (или нескольких номеров, разделенных запятыми), заключенного в квадратные скобки, например, «... подробно рассмотрены в [8].», «... данная структура является оптимальной [2, 14, 15], обеспечивая ...», «По мнению [5–8, 11], такое решение является наиболее эффективным». При необходимости, в ссылке может быть указана страница, на которой размещен интересующий материал, или соответствующий раздел, например, «[8, стр. 185]», «[14, «Архитектура клиент–сервер»]». Источник, не использованный в тексте основных разделов, должен быть исключен из списка используемой литературы.

Оформление листингов

Листинги программного кода необходимы для демонстрации конечной реализации ПП. При размещении листингов программного кода в приложениях, каждый файл должен быть снабжен заголовком. Строки каждого листинга нумеруются в следующем формате: у левой границы размещения текста ставится номер строки, затем символ-разделитель (обычно – двоеточие) и, после отступа, собственно текст строки про-

граммного кода. При описании программной реализации, в тексте пояснительной записи, кроме ссылок на соответствующее приложение, могут присутствовать ссылки на отдельные строки или группы строк листинга, например «... обеспечивает метод Paint (листинг В.2, строки 45–57), вызываемый при ...», «При определении свойства (листинг В.1, строка 23) указано, что ...». Для детального рассмотрения или большей наглядности, фрагменты программного кода могут быть представлены в виде листингов, размещенных непосредственно в тексте пояснительной записи. В этом случае нумерация строк может отсутствовать. Размещение отдельных фрагментов программного кода в тексте основной части не избавляет от необходимости представления полных листингов файлов в приложении.

Оформление содержания

Содержание предназначено для оперативной оценки состава пояснительной записи и быстрого поиска ее разделов. Содержание представляет собой перечисление заголовков разделов и подразделов, находящихся после содержания, в том числе приложений, в соответствии с их взаимной подчиненностью и с указанием номеров страниц, на которых они размещены. Обычно в содержании представляют до 3-х уровней заголовков, причем 2-й уровень имеет отступ слева относительно 1-го, а 3-й – относительно 2-го, и т.д. в соответствии со структурой пояснительной записи. Номера страниц в содержании размещаются у правой границы размещения текста вдоль одной вертикальной линии вне зависимости от уровня соответствующего заголовка. В качестве заполнителя между текстом заголовка и номером страницы рекомендуется использовать разреженную последовательность точек. Текст заголовка в содержании должен в точности соответствовать заголовку в тексте, включая нумерацию и пунктуацию. В случае набора и верстки пояснительной записи на компьютере, рекомендуется для оформления заголовков всех уровней использовать стили и автоматическую сборку содержания.

Параметры макета

Пояснительная записка к курсовому проекту должна быть выполнена на бумаге формата А4, а приложения при необходимости могут быть представлены на бумаге других стандартных форматов (А3, А2, А1, А0), подшиты к пояснительной записи и соответствующим образом сложены. Текст размещается с одной стороны листа. Необходимо, чтобы левое поле имело достаточную ширину для подшивки листов пояснительной записи; подшивка с другого края листа не допускается.

В случае набора и верстки пояснительной записки на компьютере, рекомендуются следующие параметры макета:

- поля страницы: сверху, снизу и справа – 15–20 мм, слева – 25–30 мм, расстояние от края до нижнего колонтитула – 10–15 мм, без верхнего колонтитула, без номера на первой странице;
- основной текст: шрифт Times New Roman, 12 пт (1,5 интервал) или 14 пт (одинарный интервал), отступ первой строки – 10 мм, выравнивание по ширине, отступы слева, справа, сверху и снизу абзаца – 0 мм, запрет висячих строк; параметры основного текста являются основой для оформления всех остальных элементов текста в пояснительной записке;
- подписи объектов: шрифт основного текста, выравнивание по центру, без отступа первой строки, возможно использование полужирного и курсивного начертания;
- заголовки: шрифт Times New Roman или Arial, возможно использование большего размера шрифта, полужирного начертания, курсива (для заголовков 3-го уровня), выравнивание текста абзаца влево или по центру, без отступа первой строки, не отрывать от следующего, не разрывать абзац, с новой страницы (только для заголовков 1-го уровня), отступы перед и после абзаца;
- листинги: шрифт Courier New, 9–10 пт., выравнивание влево, не разрывать абзац, рекомендуется автоматическая нумерация (желателен выступ первой строки 10–20 мм), допускается использование курсивного начертания для комментариев в тексте листингов; не следует разрывать строки, не умещающиеся по ширине листа, на отдельные абзацы (используется автоматическое разбиение абзаца на строки, действующее по умолчанию).

Образец оформления титульного листа пояснительной записки к курсовому проекту приведен в Приложении.

Контрольные вопросы

1. Перечислите основные этапы разработки программного продукта. Каково их содержание?
2. Какие задачи решаются на этапе анализа? Что является результатом этого этапа?
3. Какие стратегии разработки программного обеспечения могут применяться для создания программного продукта? В чем их сходства и отличия?
4. Какие цели преследуются на этапе проектирования? Какие задачи решаются на этом этапе?
5. Какие основные методологические приемы используются при проектировании программного обеспечения? В чем их сходства и отличия?
6. На основании каких результатов предыдущих этапов выполняется кодирование? Какую роль играют в процессе отладки средства разработки?
7. Какие цели преследуются на этапе кодирования? Какие задачи решаются на этом этапе?
8. Назовите виды тестирования. Почему является важным тестирование работы программного продукта как в штатных режимах, так и в нештатных? В чем заключаются достоинства стороннего тестирования?
9. Какой документацией комплектуется готовый программный продукт при его передаче конечному пользователю? Каково ее содержание?
10. Какие графические формы представления информации используются при разработке программного обеспечения?

Приложение. Пример оформления титульного листа

**Российская федерация
Ханты-Мансийский автономный округ – Югра
Департамент образования и науки
Сургутский государственный университет ХМАО**

Инженерно-физический факультет
Кафедра Автоматики и компьютерных систем

Пояснительная записка
к курсовому проекту
по дисциплине *(название дисциплины)*

Выполнил: студент(ка) группы *(номер)*
(фамилия и инициалы)

Проверил: *(должность преподавателя)*
(фамилия и инициалы)

Сургут
(год выполнения)

Примечания:

1. Вместо текста в скобках, набранного курсивом, указываются фактические данные без скобок и кавычек, шрифтом без курсива.
2. В случае курсового проекта по специальности, заголовок пояснительной записи оформляется следующим образом:

Пояснительная записка
к курсовому проекту по специальности *(шифр специальности)*
(семестр (номер семестра))

Методические указания

*Гришимановский Павел Валерьевич
Даниленко Иван Николаевич*

**Курсовое проектирование
Разработка программного обеспечения**

Оригинал-макет подготовлен
в редакционно-издательском отделе СурГУ

Подписано в печать _____. Формат 60×84/16.
Усл. печ. л. _____. Уч.-изд. л. _____.
Печать трафаретная. Тираж _____. Заказ № _____.

Сургутский государственный университет
626400, Россия, Ханты-Мансийский автономный округ,
г. Сургут, ул. Энергетиков, 14.
Тел. (3462) 243781, факс 243687

Отпечатано полиграфическим отделом СурГУ
ул. Лермонтова, 5.
Тел. (3462) 323305

Лицензия на издательскую деятельность ЛР № 071409
от 6 марта 1997 г.