

Федеральное агентство по образованию РФ

Дальневосточный федеральный университет

Кафедра информатики, математического и компьютерного
моделирования

Кленин А. С.

Методические указания по подготовке и защите работ по программированию

Версия 2.0

Владивосток, 2002-2015

Содержание

Содержание 2

Введение 4

Требования к оформлению отчёта 4

Формат и именование 4

Бумажная копия отчёта 5

Страницы и поля 5

Шрифт и пунктуация 5

Абзацы и разделы 6

Таблицы и рисунки 6

Диаграммы 7

Формулы 7

Мета-заголовки 8

Титульный лист 8

Раздел «Содержание» 9

Раздел «Аннотация» 9

Раздел «1. Введение» 9

Раздел «1.1. Глоссарий»	9
Раздел «1.2. Описание предметной области»	9
При наличии заказчика (прикладная задача)	9
При отсутствии заказчика (исследовательская проблема)	10
Описание совместной деятельности	10
Раздел «1.3. Неформальная постановка задачи»	11
Раздел «1.4. Математические методы»	11
Раздел «1.5. Обзор существующих методов решения»	11
Аналогичные (конкурирующие) решения	11
Описание предшествующих работ	12
Вывод	12
Раздел «1.6. План работ»	12
Раздел «2. Требования к окружению»	12
Раздел «2.1. Требования к аппаратному обеспечению»	12
Раздел «2.2. Требования к программному обеспечению»	13
Раздел «2.3. Требования к пользователям»	13
Раздел «2.4. Организационные требования»	13
Раздел «3. Архитектура системы (Общие требования)»	13
Раздел «4. Спецификация данных»	14
Раздел «4.1. Описание формата или структуры данных»	14
Раздел «4.2. Описание сущности»	15
Раздел «4.3. Описание протокола»	15
Раздел «5. Функциональные требования»	15
Библиотека подпрограмм (классов)	16
Раздел «6. Требования к интерфейсу»	16
Раздел «7. Прочие требования»	17
Раздел «7.1. Требования к надёжности»	17
Раздел «7.2. Требования к безопасности»	17

[Раздел «7.3. Требования к производительности» 18](#)

[Раздел «8. Проект» 18](#)

[Раздел «8.1. Средства реализации» 18](#)

[Раздел «8.2. Структуры данных» 19](#)

[Раздел «8.3. Модули и алгоритмы» 19](#)

[Раздел «8.4. Стандарт кодирования» 19](#)

[Раздел «8.5. Проект интерфейса» 19](#)

[Раздел «9. Реализация и тестирование» 19](#)

[Раздел «9.1. Вычислительный эксперимент» 20](#)

[Раздел «Заключение» 20](#)

[Раздел «Список литературы» 21](#)

[Раздел «Приложение» 21](#)

[Исходный код 22](#)

[Форматирование кода 22](#)

[Именованние объектов 22](#)

[Внутренняя документация 23](#)

[Доклад 23](#)

[Презентация 24](#)

[Общее описание 24](#)

[Структура 24](#)

[Оформление 25](#)

[Раздаточный материал 25](#)

[Список литературы 26](#)

[Приложение 1. Примерный текст Акта о внедрении 27](#)

[Акт о внедрении 27](#)

Введение

Настоящий документ описывает требования, предъявляемые к письменным работам (отчётам), выполняемым студентами специализации «Прикладная математика и информатика. Системное

программирование» кафедры информатики, математического и компьютерного моделирования ДВФУ. Требования составлены в соответствии с государственными стандартами и учебным планом. При этом учитывались также индустриальные стандарты и аналогичные требования других вузов (например, [10]). Подробно эти требования излагаются в рамках курса «Технология программирования» [9].

Наиболее полно описаны курсовые и дипломные работы, однако большинство положений применимо и к случаю отчёта по учебно-вычислительной практике или практикуму на ЭВМ. Некоторые положения, в частности правила оформления, относятся также и к рефератам и другим письменным работам.

Отчёт по практикуму, учебно-вычислительной, курсовой или дипломной работе является документом, предназначенным для описания целей и задач работы, предпринятых действий и достигнутых результатов. Настоящий документ предполагает, что работа защищается в рамках специализации «Прикладная математика и информатика. Системное программирование», и поэтому цель работы состоит в разработке и реализации программной системы. Целевая аудитория для отчёта состоит из преподавателей и студентов указанной специализации, что определяет предполагаемый уровень компетентности читателя.

Текущую версию данного документа можно найти по адресу <https://github.com/klenin/dvfu-repplan/>.

Требования к оформлению отчёта

Формат и именование

Отчёт должен быть подготовлен с использованием текстового редактора Microsoft Word или другой системы, предоставляющей аналогичные возможности, например, TEX. Все изложенные требования к внешнему виду работы должны выполняться вне зависимости от используемого программного средства. Отчёт должен быть представлен в электронной форме, пригодной к просмотру на компьютерах с ОС Windows, например, в формате RTF/DOC (Microsoft Word), ODF (Open Office), PDF или HTML. Для документов формата PDF необходимо убедиться, что используются векторные, а не растровые шрифты.

Весь текст отчёта должен содержаться в одном файле. Имя файла отчета должно состоять из номера группы, фамилии автора и вида работы, разделённых знаком подчёркивания «_». Например: «239_Петров_курсовая.doc». Изображения могут быть либо встроены (форматы DOC, ODF, PDF), либо содержаться в каталоге, префиксом имени которого является имя файла отчёта (HTML, RTF). Файл презентации должен иметь то же имя, что и файл отчёта, с соответствующим расширением.

При использовании редактора Microsoft Word желательно сохранять документ в формате RTF, а не DOC. Исключение составляет случай, когда формат RTF значительно (более чем в пять-шесть раз) проигрывает в объёме. Это обычно происходит при внедрении в документ крупных изображений. В любом случае документ *не должен* содержать никаких макросов, даже разработанных автором документа. При использовании формата DOC следует сохранять документы в режиме совместимости со старыми версиями Word (т. е. не использовать формат DOCX).

Бумажная копия отчёта

Отчёт представляется комиссии в распечатанном виде. Если работа не защищается перед комиссией (например, реферат), то, по согласованию с преподавателем, отчёт можно представить только в электронном виде. При этом правила его оформления остаются неизменными.

Бумажная копия отчёта должна быть скреплена при помощи скрепок или прошивки и помещена в пластиковую папку. Рекомендуется использовать как можно более тонкие папки, поскольку это упрощает складирование большого количества распечатанных работ в архиве.

Страницы и поля

Отчёт должен быть напечатан на стандартных листах белой бумаги формата A4 (297 x 210 мм) плотностью не менее 60 г/м². Поля составляют 3 см слева (на переплёт), по 1 см сверху, снизу и справа.

Страницы отчёта, кроме титульного листа, должны быть пронумерованы, при этом титульный лист считается первой страницей, и поэтому нумерация начинается со страницы 2. Номера должны быть расположены внизу страницы по центру.

Шрифт и пунктуация

Текст должен быть набран шрифтом Times New Roman или аналогичным шрифтом с засечками, кеглем 12, прямым начертанием. Тексты программ и другая машинно-ориентированная информация должна быть набрана шрифтом Courier New или другим моноширинным шрифтом.

Для выделения текста следует использовать *наклонное* и **полужирное** начертание. Не рекомендуется использовать подчёркивание, а также одновременно применять более одного способа выделения одного и того же текста. В случае необходимости выделить слово внутри большого отрывка, набранного курсивом, следует набрать это слово прямым начертанием. Для цитирования и других целей следует заключать текст в парные кавычки типа «ёлочки» либо “лапки”. Первые предпочтительнее использовать в русскоязычных, а вторые — в англоязычных текстах. Непарные «прямые» кавычки рекомендуется использовать только в текстах программ и т. п. местах.

Пробелы вокруг знаков препинания расставляются в соответствии с обычными правилами: после запятых, точек, и двоеточий, снаружи, но не внутри скобок, и т. д. В тексте не должно встречаться последовательностей из двух и более пробелов подряд.

Обратите внимание на использование символов - (дефис), – (минус), и — (тире). Дефис применяется для разделения частей сложных слов и обозначения переноса, минус — для обозначения операции вычитания и разделения границ числового диапазона, а тире — как знак препинания для разделения частей предложения. При использовании текстового редактора Microsoft Word, знак дефиса получается нажатием клавиши «минус», знак минус — нажатием комбинации клавиш Ctrl + «серый минус», а знак тире — нажатием комбинации клавиш Ctrl + Alt + «серый минус».

При использовании системы подготовки текстов TEX для знаков дефис, минус и тире используется соответственно один, два и три подряд идущих символа «минус».

В тексте не должно быть переносов слов, сделанных при помощи разрыва строки. Рекомендуется вместо этого включить автоматический перенос. При использовании редактора Microsoft Word для указания места переноса в слове, неизвестном автоматической системе, можно использовать «мягкий перенос», получаемый нажатием клавиш Ctrl + «минус» (команда «\» в системе TEX).

Напротив, предотвратить разрыв строки можно при помощи «неразрывного пробела», получаемого нажатием клавиш Ctrl + Shift + «пробел» (символ «тильда» в системе TEX). Неразрывный пробел рекомендуется вставлять там, где разрыв строки запрещён правилами русского языка, то есть между:

- словом и следующим за ним знаком тире,
- инициалами людей, в т. ч. авторов в списке литературы («Иванов И. И.»),
- точкой и следующей буквой сокращений типа «т. е.», «и т. д.»,
- сокращениями названий и именами собственными («г. Владивосток»),
- числовыми величинами и единицами измерений («3 см»).

Проконтролировать наличие и расположение невидимых символов в редакторе Microsoft Word можно с помощью соответствующего пункта настроек либо путём нажатия клавиш Ctrl + Shift + 8.

Абзацы и разделы

Абзацы должны быть выровнены по ширине, и иметь отступ первой строки 1 см («красную строку»). В отчёте о дипломной работе требуется полуторный интервал между строками. В прочих видах работ полуторный интервал следует заменять одинарным с целью экономии бумаги. С той же целью в отдельных случаях допускается (но не рекомендуется) распечатка отчёта на обеих сторонах листа.

При оформлении нумерованных списков следует использовать для нумерации арабские цифры, отделяя номер пункта от текста точкой. Вложенные списки нумеруются иерархически («1.1.», «1.2.», ...). Не рекомендуется создавать списки с более чем двумя уровнями вложенности.

Пункты нумерованных списков выделяются с помощью одного из специальных символов (буллитов). Все списки одного уровня в пределах одного документа должны использовать один и тот же буллит.

Разделы отчёта должны быть озаглавлены с использованием соответствующих стилей, и набраны шрифтом Arial или аналогичным шрифтом без засечек, кеглем 12–14, полужирным начертанием. Текст заголовка любого уровня должен быть выровнен влево и набран с отступом, равным отступу красной строки (1 см).

Все разделы, кроме специально отмеченных ниже, должны быть иерархически пронумерованы. В Microsoft Word для этого следует изменить в стилях «Заголовок 1», «Заголовок 2», и т. д. нумерацию на «многоуровневую». Текст заголовка должен быть отделён от номера точкой и ровно одним пробелом (а не символом табуляции).

Таблицы и рисунки

Все таблицы и рисунки должны быть снабжены подписями вида «Табл. *N*. *Название таблицы*» или «Рис. *N*. *Название рисунка*». Нумерация таблиц и рисунков сквозная и независимая. На каждую таблицу или рисунок должна быть ссылка в тексте, например «смотри Табл. 7». При использовании текстового редактора Microsoft Word вставку ссылки следует производить при помощи команды «Вставка / Ссылка / Перекрестная ссылка». Подписи к таблицам и рисункам должны быть

оформлены как абзацы обычного текста. При этом подпись к таблице должна находиться непосредственно *перед* таблицей, а подпись рисунка — непосредственно *после* него.

Не допускается появление разрыва страницы между подписью и таблицей или рисунком. Также недопустимо появление рисунка или таблицы в качестве первого элемента раздела — они должны быть отделены от заголовка раздела по крайней мере одной строкой текста.

При очень большом (более 20–30 штук) количестве таблиц и/или рисунков возможна их иерархическая нумерация по номеру раздела (например, «Рис. 4.23»). В этом случае рекомендуется также составление списка иллюстраций и списка таблиц.

Ширина таблицы должна совпадать с шириной основного текста отчёта, т. е. боковые стороны внешней рамки таблицы должны совпадать с границами полей листа. Если содержимое строки таблицы по ширине значительно меньше листа, следует прижать таблицу к левому краю (см., например, Табл. 1).

Если таблица содержит большой объём текста (более половины страницы), рекомендуется набирать его шрифтом кегля 11 или кегля 10 без засечек.

Заголовки таблиц следует выделять полужирным начертанием того же шрифта, что и текст. Дополнительно возможно выделение заголовка таблицы заливкой 10–20% серого цвета. Ячейки таблицы должны быть разделены линиями толщиной 0,5–0,75 пункта. Рамку вокруг таблицы, горизонтальную линию, отделяющую заголовков, а также, при большом количестве строк в таблице, каждую третью-пятую горизонтальную линию между ячейками, рекомендуется проводить толщиной 1–1,5 пункта.

Диаграммы

Для иллюстрации отчёта следует использовать диаграммы, соответствующие стандарту UML . Диаграммы подписываются как рисунки, например «Рис. 3. Диаграмма состояний объекта Диспетчер». При этом в название диаграммы должно содержать информацию помимо типа диаграммы. Например, недопустимо сокращение приведённого выше названия до «Рис. 3. Диаграмма состояний».

Диаграммы можно рисовать либо при помощи встроенных средств текстового редактора, либо с использованием специальных систем (например, Microsoft Visio, Magic Draw UML, ArgoUML, dot). Диаграмма должна полностью уместиться на странице и, по возможности, не иметь пересечений в линиях. Если этого не удаётся достичь, следует абстрагировать диаграмму и/или разбить её на несколько частей. Толщина соединительных и граничных линий на диаграмме не должна превышать 1 пункта (рекомендуется оставлять стандартную толщину 0,75 пункта). Подписи на диаграмме должны быть набраны шрифтом без засечек 10–12 кегля или, в зависимости от содержания, моноширинным шрифтом 8–10 кегля. При рисовании с помощью встроенных средств редактора Microsoft Word следует сгруппировать готовую диаграмму.

Стандартные стереотипы следует записывать на английском языке. Использование нестандартных стереотипов и обозначений разрешено только по согласованию с руководителем и должно быть описано в специальном разделе отчёта.

Большинство диаграмм, за исключением чрезмерно громоздких, следует включать в текст согласно приведённым ниже указаниям. Остальные диаграммы следует собрать в отдельном приложении.

Формулы

Для вставки в текст формул используется один следующих методов:

- при использовании Open Office и TEX — встроенные средства набора;
- при использовании HTML — формулы, сохранённые как изображения в отдельных файлах либо, в простых случаях, отформатированные стандартными средствами HTML;
- при использовании Microsoft Word — формулы, набранные с помощью Equation Editor либо отформатированные стандартными средствами Word.

Стабильность и эффективность работы Microsoft Word значительно снижается при наличии в документе большого количества внедрённых объектов. Поэтому для вставки простых формул, особенно отдельных имён переменных, рекомендуется применять следующие правила форматирования:

- имена переменных должны быть набраны курсивом того же кегля, что и основной текст;
- числа, знаки препинания (например, запятые, скобки), и знаки операций должны быть набраны прямым шрифтом;
- знаки операций и знак равенства должны быть окружены (обычно неразрывными) пробелами;

Пример правильно набранной формулы: $c^2 = a^2 + b^2$. Сложные дроби, знаки радикала, интегралы и матрицы следует набирать при помощи Equation Editor.

Формулу не следует разрывать между строками, если её длина меньше одной строки. Длинные либо особо важные формулы рекомендуется выносить на отдельную строку. Для ссылки на такие формулы при небольшом их общем количестве они должны иметь подписи аналогично рисункам, например «Формула 1. Теорема Пифагора». Если ссылок на формулу нет, подпись можно опустить.

Мета-заголовки

Последующие разделы настоящего документа соответствуют разделам отчёта, и имеют заголовки вида *Раздел «Номер раздела. Название раздела»*, означающие, что раздел с таким номером и названием должен появиться в отчёте.

В тексте под этими заголовками излагаются требования к отчёту. Часть требований, изложенных в повелительной форме («указать», «перечислить») являются обязательными. Другие, помеченные словами «по возможности», «желательно», и т. п., имеют статус рекомендаций.

Титульный лист

Оформление титульного листа дипломной работы см. в .

Титульный лист должен содержать в верхней части выровненные по центру строки: «Федеральное агентство по образованию РФ», «Дальневосточный государственный университет», «Институт математики и компьютерных наук», «Кафедра информатики». В центре листа должно находиться

название работы без слова «Тема» и обрамляющих кавычек. Название работы не должно, как правило, содержать фраз «разработка программы», «программная система для ...» и т. п., поскольку они подразумеваются для всех работ.

Далее после промежутка должны идти выровненные вправо строки, содержащие:

- вид работы (например, «курсовая работа», «отчёт по преддипломной практике»),
- фразу «студента *NNN* группы», где *NNN* — номер группы,
- Фамилию И. О. исполнителя в родительном падеже,
- слово «руководитель»,
- ученую степень, должность и место работы руководителя,
- Фамилию И. О. руководителя.

В нижней части должна содержаться выровненная по центру строка «Владивосток, *YYYY*», где *YYYY* — четыре цифры года завершения работы.

Все надписи должны быть выполнены шрифтом без засечек 16 кегля прямого нормального начертания. Название работы должно быть набрано тем же шрифтом 20 кегля, возможно, с использованием «малых прописных букв» (small caps).

Раздел «Содержание»

Раздел не нумеруется. Содержание должно быть сгенерировано автоматически средствами используемого текстового редактора на основании стилей документа. В содержании должны указываться заголовки первого, второго, и, возможно, третьего уровня. Номера страниц должны быть прижаты к правому краю и отделены от названий разделов последовательностью точек.

Раздел «Аннотация»

Раздел не нумеруется. Цель раздела — дать краткий обзор данной работы, на основании которого читатель, заинтересованный заголовком работы, может принять решение о дальнейшем изучении данной работы. Раздел должен состоять из не более 100 слов (приблизительно два-три предложения). В аннотации должны быть охарактеризованы цели работы и основные достигнутые результаты, особенно следует подчеркнуть отличие от аналогичных работ. Раздел *не* должен являться точной копией начальных предложений из раздела «Введение».

Раздел «1. Введение»

Цель введения — сформировать у читателя общее представление о предметной области и решаемой задаче, а также обосновать необходимость и целесообразность её решения путём создания программной системы. Введение состоит из подразделов, некоторые из которых могут быть опущены при отсутствии релевантной информации в работе.

Раздел «1.1. Глоссарий»

Если необходимо, привести список используемых в работе терминов и сокращений, специфичных для данной предметной области и/или для используемых программных технологий. Не следует указывать общераспространённые в программировании сокращения типа «СУБД», поскольку их смысл предполагается известным читателю. И наоборот, необходимо указать все используемые в работе сокращения, специфичные для предметной области, даже если автор отчёта привык к ним за время исполнения работы и считает их очевидными. Термины в глоссарии упорядочить по алфавиту.

Раздел «1.2. Описание предметной области»

Описание должно содержать, по меньшей мере, *две ссылки на литературу*.

При наличии заказчика (прикладная задача)

Кратко описать деятельность предприятия или организации-заказчика в целом и более подробно — в аспектах, связанных с описываемой работой. Отметить также сходство и различия между деятельностью данного предприятия и других в данной отрасли / виде деятельности.

Описать *проблему*, подлежащую решению — неэффективность или невозможность выполнения определённых операций, возникающие в связи с этим технические и экономические ограничения деятельности предприятия. Указать *не менее трёх* возможных подходов к решению проблемы, в том числе подход, применяемый до внедрения результатов описываемой работы. Сделать вывод о необходимости и/или выгоды применения программной системы.

Рассмотреть *пределы применения* предполагаемой программной системы — очертить границы класса задач, аналогичных поставленной, которые возможно решить с её помощью и оценить масштаб требующихся для этого изменений в системе. Например, для программы складского учёта в конкретной фирме рассмотреть возможность применения для других фирм того же профиля, для всех фирм, имеющих склад, и т. п.

При отсутствии заказчика (исследовательская проблема)

Кратко описать вид деятельности и/или направление исследований, в рамках которого выполняется описываемая работа. Перечислить основные исторические моменты и достижения в указанном направлении. Более подробно описать текущее состояние, особенно в области, имеющей непосредственное отношение к описываемой работе.

Описать *проблему*, подлежащую решению — неизученный или недостаточно изученный вопрос, прояснение которого позволило бы достичь продвижения в той или иной области науки или прикладного знания. Сделать вывод о необходимости или полезности для решения данной проблемы проведения вычислительного эксперимента или другого исследования с использованием программной системы.

Работа относится к категории исследовательских, только если её результаты имеют научную ценность сами по себе, а не являются утилитарной частью какого-либо научного проекта. Например, база данных, хранящая результаты измерений, — прикладная задача, в то время как алгоритм, строящий на основании этой базы данных какие-либо прогнозы и выводы, может быть отнесён к научно-исследовательской работе.

Рассмотреть *пределы обобщения* предполагаемых результатов.

Описание совместной деятельности

В случае если описываемая система является частью более крупного проекта, описать цели, задачи и структуру проекта, чётко выделить подзадачу, решаемую в рамках защищаемой работы. Привести ссылки на литературу, описывающую проект в целом и другие его части (например, на отчёты однокурсников, страницу сайта проекта).

При совместном участии в открытых проектах рекомендуется предварительно согласовать тему работы с основными участниками проекта. В работе привести ссылки на материалы обсуждения, например архив списка рассылки.

Защита одной и той же работы двумя студентами не допускается. Каждый защищающийся должен представить отдельный отчёт, отличающийся как по названию, так и содержательно. Рекомендуется прибавлять к общей части названия конкретизирующее слово или предложение (например, «серверная часть» и «клиентская часть»). Допустимо, хотя и нежелательно, дословное совпадение начальной части раздела «Описание предметной области».

При совместной реализации крупных проектов несколькими студентами необходимо выделить каждому максимально независимые подзадачи со строго определёнными областями ответственности и интерфейсами. Желательно при этом достичь такой степени независимости, чтобы возможна была реализация и защита любой из работ даже в случае полного отсутствия остальных. Невозможность такого разделения следует рассматривать как фатальное препятствие для защиты.

Раздел «1.3. Неформальная постановка задачи»

Привести *список пожеланий (неформальных требований)* к системе, решающей проблему, поставленную в п. 1.1. Следует ограничиться общими утверждениями о наиболее крупных возможностях, и не проводить их детализацию. Однако должны быть указаны все свойства и особенности, необходимые для дальнейшего обсуждения в п. 1.5. При совместной разработке убедиться, что требования к системе согласованы и не совпадают с требованиями к другим подсистемам того же проекта.

В случае если среди требований можно выделить нетривиальные отношения (например, включения, расширения), и/или в системе имеется несколько акторов (ролей пользователей), привести *диаграмму вариантов использования* верхнего уровня.

Пункты списка могут быть сформированы по правилам функциональных требований (п. 5), реже — требований к данным (п. 4). Если количество пунктов превосходит 5–6, следует при помощи средств оформления визуально отделить эти виды требований. В целом количество неформальных требований не должно быть более 10–15, а каждое требование должно занимать 1–2 строки текста.

Отдельно указать предполагаемую политику распространения программного продукта (только в организации-заказчике, продажа на открытом рынке, свободное распространение). В случае свободного распространения желательно выбрать и указать лицензию (BSD, GPL, Creative Commons и т. п.) и версию этой лицензии.

Раздел «1.4. Математические методы»

Если для решения задачи используются математические методы, модели и алгоритмы, имеющие самостоятельное значение и/или являющиеся новыми для данной предметной области, описать их и привести необходимые ссылки на литературу.

Также привести все определения, модели и доказательства, разработанные автором в процессе выполнения работы либо предложенные руководителем в качестве основополагающих для данной работы. (Например, доказательство корректности предложенного алгоритма или математическую модель, оптимизации параметров которой посвящена работа).

При достаточной сложности и объёме может потребоваться выделение раздела первого уровня.

Раздел «1.5. Обзор существующих методов решения»

Аналогичные (конкурирующие) решения

Привести список, содержащий *не менее трёх* систем, аналогичных или близких по тематике к описываемой системе. Описать их основные возможности, сравнительные достоинства и недостатки, опираясь на список неформальных требований п. 1.2. Описание каждой системы должно содержать ссылку на литературу.

Привести *таблицу сравнительных характеристик*, в которую будут включены как наиболее репрезентативные конкурирующие решения, так и планируемый к разработке программный продукт. В зависимости от конкретной задачи рекомендуется рассмотреть в качестве кандидатов на включение в таблицу следующие характеристики:

- выделяющиеся функциональные возможности,
- виды пользовательского интерфейса, наличие локализованных версий;
- ограничения по ресурсам и размеру решаемых задач,
- поддерживаемые платформы,
- лицензионную политику (proprietary, open source, free software), желательно с указанием конкретной лицензии,
- модель продаж (commercial, freeware, shareware) и стоимость.

Документировать использованные методы поиска и обосновать, таким образом, полноту обзора аналогичных систем в мировом масштабе, либо привести аргументы в пользу невозможности выполнения полного обзора (например, дороговизна наиболее известных систем и отсутствие при этом их обзоров в литературе).

В случае если необходимость создания системы аргументируется в основном высокой стоимостью коммерческих аналогов, подробно изучить имеющееся свободно распространяемое программное обеспечение и обосновать неэффективность его использования или модификации для решения поставленной задачи. Работы, дублирующие функциональность имеющихся свободных систем и не имеющие перед ними никаких преимуществ, могут быть не допущены к защите.

В то же время, рекомендуется и приветствуется доработка существующих свободных систем, особенно с последующей интеграцией внесённых изменений в основную версию. Если доступны

несколько свободных систем, привести их сравнительный анализ и обосновать выбор системы для доработки.

Описание предшествующих работ

В случае если в той же организации либо под управлением того же руководителя в прошлом были предприняты попытки решения задачи, близкой или совпадающей с поставленной, описать их. Поместить ссылки на описание этих работ (например, отчёты) в список литературы.

Указать результаты предшествующих работ, их взаимосвязь. Перечислить причины их неудачного завершения либо неполного соответствия результатов потребностям и предполагаемые отличия данной работы.

Вывод

Сделать вывод о несоответствии всех имеющихся систем предъявленным требованиям и необходимости, поэтому, разработки собственной системы.

Раздел «1.6. План работ»

Перечислить в хронологическом порядке виды и примерные сроки работ, необходимых для реализации системы. План работ является необходимой частью отчёта по преддипломной практике. В прочих случаях план разрабатывается по специальному требованию руководителя или заказчика.

Раздел «2. Требования к окружению»

Раздел требований к окружению содержит, в отличие от последующих разделов, не требования к системе, а, напротив, требования, предъявляемые системой к окружающему миру.

Раздел «2.1. Требования к аппаратному обеспечению»

Описать *минимально необходимую* и, возможно, рекомендуемую аппаратную конфигурацию для *эффективной* работы описываемой системы (т. е. такую конфигурацию, в которой система не только запускается, но и способна производить полезные действия). Особенно выделить нестандартное оборудование, необходимое для работы, например, звуковую карту или интерфейс специализированного измерительного прибора. Если нестандартные устройства необходимы для выполнения лишь некоторых функций системы, то указать эти устройства и функции отдельно. Если система требует более одного аппаратного узла (например, клиент-серверная или кластерная архитектура), описать требования к каждому узлу по отдельности, а также требования к каналам связи между узлами.

Раздел «2.2. Требования к программному обеспечению»

Описать базовое программное обеспечение, установка которого необходима для функционирования описываемой системы, например, операционную систему, веб-браузер, СУБД и т. п. Для каждого компонента ПО указать аппаратный узел из п. 2.1, на котором он будет выполняться. Привести *точные версии* компонентов, на которых производилось тестирование,

использовать фразу «и выше» для указания на то, что описываемая система будет, вероятно, работать и с более поздними версиями.

В случае разработки библиотеки требования к программному окружению включают компилятор или среду программирования, для которой предназначена библиотека.

Если система переносима между различными платформами, то указать, во-первых, полные и точные программные конфигурации платформ, на которых производилось тестирование, и, во-вторых, базовый компонент, от переносимости которого зависит переносимость описываемой системы. (Например, «и другие ОС, поддерживающие стандарт POSIX» или «и другие платформы, на которых работает PHP 4»).

Раздел «2.3. Требования к пользователям»

Описать категории пользователей системы, при этом указать для каждой категории:

- название категории (используется также термин «роль»);
- краткое обозначение исполняемых функций как в системе, так, где это применимо, и в организации (например «ввод первичных данных»);
- доступные возможности системы;
- требования, предъявляемые к знаниям, умениям и навыкам данного пользователя.

Не указывать тривиальные навыки (например, умение читать) там, где это очевидно. Напротив, специально указать возможное отсутствие какого-либо базового навыка у пользователей (например, «программа предназначена для детей дошкольного возраста, не умеющих читать», «программа поддерживает работу пользователей с дефектами зрения»). Обратить особое внимание на знание пользователями языка и терминологии, используемой в интерфейсе.

Раздел «2.4. Организационные требования»

Указать изменения в структуре и производственном процессе организации-заказчика, необходимые для успешного внедрения системы. Например, «требуется введение ставки системного администратора с приблизительной занятостью в половину рабочего дня» или «ежедневное резервное копирование данных необходимо вменить в обязанность главному менеджеру». В случае отсутствия или незначительности таких изменений раздел опустить.

Раздел «3. Архитектура системы (Общие требования)»

Описать разделение системы на видимые с точки зрения пользователя подсистемы (например, клиентскую и серверную часть), кратко указать функции каждой подсистемы и перечислить общие для всех подсистем требования. Привести *диаграмму пакетов* и/или *диаграмму развёртывания*. Если система создаётся группой разработчиков, указать общие требования либо сослаться на документ, описывающий их, а также указать подсистемы, разрабатываемые автором отчёта. Все изложенные в последующих разделах требования должны относиться только к этим подсистемам.

Данный раздел можно опустить при описании простой системы, не подразделяющейся на более мелкие подсистемы.

Раздел «4. Спецификация данных»

Описать внешние (видимые пользователю и/или другим программам) протоколы, форматы и структуры данных, разработанные в рамках описываемой работы. Сюда входят форматы файлов, протоколы обмена информацией между подсистемами, описанными в п. 3, сущности, хранимые в базе данных и имеющие смысл в предметной области.

Не следует включать в данный раздел протоколы, форматы и структуры:

1. Стандартные — описанные в существующих и не относящихся к

описываемой системе документах, например, формат RTF, протокол TCP/IP. Вместо описания указать ссылку на соответствующий документ (например, RFC). Если некоторый формат или протокол имеет большое значение для понимания структуры и функциональности программы, можно добавить его краткое (не более одной страницы) описание. Если, кроме этого, описание протокола или формата труднодоступно (например, это внутренний формат, принятый в рамках описываемой организации), то его можно привести в *Приложении*.

2. Строго внутренние — в том числе структуры данных в оперативной

памяти, протоколы, форматы и сущности, которые никогда не видны внешним пользователям и программам и не упоминаются в функциональных требованиях. Эти данные должны быть описаны в разделе «Проект».

3. Секретные и закрытые — попадающие под действие коммерческой или

государственной тайны. Наличие таких данных может служить препятствием для допуска работы к защите и должно специально оговариваться с руководителем. Следует различать случаи закрытых данных (распространённая ситуация, не являющаяся помехой защите работы) и закрытого формата.

Раздел состоит из подразделов, по одному на каждый описываемый объект. Названия подразделов должны начинаться со слов «Протокол», «Формат», «Сущность» и т. п., за которыми следует название соответствующего объекта данных. Если объекты могут быть содержательно сгруппированы по подсистемам, выделенным в п. 3 (например, сущности, хранимые в БД каждой подсистемой), и таких объектов в среднем не менее трёх в каждой группе, то следует ввести подразделы третьего уровня.

Раздел «4.1. Описание формата или структуры данных»

Описать назначение формата, указать предполагаемый носитель (например, файл или оперативная память).

Указать, текстовый или двоичный формат. Для текстового формата указать: кодировку символов, правила использования символов перевода строки и возврата каретки (CR/LF), возможно, базовый

формат (например, XML). Для двоичного формата указать способ кодировки многобайтных целых чисел (big- или little-endian), формат вещественных чисел (например, IEEE-754 Single Precision).

Подробно описать синтаксис и семантику каждого элемента данных, привести примеры. Если возможно, привести *таблицу полей* или другое структурированное изображение формата. Описать возможные ошибки в представлении данных и реакцию на них.

Раздел «4.2. Описание сущности»

Термин сущность (таблица) обычно используется при разработке баз данных и объектно-ориентированных систем.

Описать назначение сущности («предназначена для хранения ...» либо «реализует понятие ...»).

Перечислить атрибуты сущности, имеющие смысл в предметной области. При этом не указываются служебные поля БД, невидимые для пользователя (например, искусственный первичный ключ, время последней модификации или номер реплики). Для каждого атрибута указать:

- полное название по-русски (а не имя соответствующего поля БД),
- тип (без детализации, например, не «VARCHAR(200)», а «Строка»),
- ограничения на допустимые значения,
- обязательность,
- доступность для редактирования,
- уникальность (для составных ключей указать «комбинация значений атрибутов А, В, ... должна быть уникальной»),
- значение по умолчанию (если есть).

Если атрибут используется для создания отношения «один ко многим», следует указать его тип как «ссылка на *Название сущности*», а не «Целое число».

Оформить список атрибутов в виде таблицы, содержащей все или часть из перечисленных характеристик. Часто повторяющиеся значения вынести в текст после таблицы. Например «Значения атрибута *Фамилия* должно быть уникальным» или «Для редактирования доступны все атрибуты, кроме атрибута *Дата создания объекта*».

Описать дополнительные требования к данным сущности. Перечислить объекты сущности, автоматически создаваемые при установке системы, и значения их атрибутов.

Привести *диаграмму классов* (или объектов).

Раздел «4.3. Описание протокола»

Описать назначение протокола.

Указать стороны, сообщающиеся при помощи данного протокола, а также базовый протокол более низкого уровня (например, согласно модели OSI [2]). Формализовать отображение описываемого протокола на базовый. Описать формат сообщений протокола и общие правила их интерпретации,

аналогично п. 4.1. Указать динамические требования к протоколу (задержки, тайм-ауты, повтор сообщений и т. д.). Указать наличие хранимого состояния в протоколе (stateful или stateless).

Перечислить список всех сообщений протокола, для каждого сообщения указать содержащиеся в нём значения фиксированных полей (например, тип пакета) и семантику переменных полей, а также сопутствующие изменения в состояниях сторон. Если формат сообщений достаточно сложен, описать его аналогично структуре данных.

Обратить внимание на возможные ошибки и непредвиденные ситуации, в частности, непредвиденное прекращение функционирования одной из сторон и разрыв канала связи.

Привести, по меньшей мере, одну *динамическую диаграмму* (например, *диаграмму состояний*).

Раздел «5. Функциональные требования»

Перечислить требования в формате нумерованного списка, предварённого фразой «Система должна:» или «Система должна позволять пользователю:». Если система состоит из подсистем согласно п. 3, то данный раздел должен быть соответственно разбит на подразделы, объединяющие требования к каждой из подсистем.

Отдельные требования должны начинаться с глагола в неопределённой форме (например, «создавать», «редактировать») либо с условия, за которым следует такой глагол (например, «в случае сбоя выдавать сообщение об ошибке»). Допускается объединение нескольких требований в одно предложение (например, «добавлять и удалять записи»). Следует, где это возможно, использовать глаголы несовершенного вида («просматривать», но не «просмотреть»), поскольку этим подчёркивается повторяемость действий.

Требования должны быть чётко сформулированными и иметь однозначную интерпретацию. В случае необходимости следует привести уточняющее предложение (например, «редактировать данные о пользователе, при этом изменять значения всех полей, кроме идентификатора пользователя»). Для сокращения повторяющихся наборов требований рекомендуется вводить дополнительные определения (например, «далее под термином *работать со списком* понимается возможность добавлять, удалять и изменять элементы списка»).

В случае если у системы или подсистемы имеется несколько категорий пользователей с различными возможностями, разбить список требований на отдельные списки для каждого пользователя, и указать перед ними соответствующие фразы (например, «Система должна позволять Бухгалтеру...»). При этом названия категорий должны точно соответствовать разделу 2.3. Альтернативным вариантом является указание списка пользователей для каждого требования в отдельности. Следует выбрать один из вариантов (обычно из соображений компактности представления), и последовательно придерживаться его на протяжении всего раздела. Если права категорий пользователей организованы иерархически, следует упорядочить их в порядке увеличения доступных возможностей и заменять повторяющиеся части фразами вида «позволять Администратору: ... совершать все действия, доступные Пользователю».

Привести, по мере необходимости, детализированные *диаграммы вариантов использования*, *диаграммы кооперации*, и другие.

Библиотека подпрограмм (классов)

Перечислить возможности, предоставляемые библиотекой пользователю. Описание конкретных классов и функций поместить в требования к интерфейсу. Для небольших библиотек допускается объединение разделов 5 и 6. Алгоритм следует называть только в случае, если он фиксирован внешними требованиями (например, «решать систему линейных уравнений», либо «решать систему линейных уравнений методом Гаусса»).

Раздел «6. Требования к интерфейсу»

Определить требования к общему виду и платформе интерфейса (например, «интерфейс командной строки» или «Windows-интерфейс», или «Web-интерфейс»). Если система обладает несколькими интерфейсами, перечислить их и указать, какая часть функциональности доступна через каждый из них.

Указать специальные требования к интерфейсу, накладываемые составом целевой аудитории, ограничениями технических средств либо требованиями заказчика. Например: «интерфейс должен быть доступен для использования слепыми людьми», «интерфейс должен функционировать при использовании браузера Internet Explorer 3.0», «все функции интерфейса должны быть доступны при использовании для ввода только клавиатуры» и т. п.

Описать структуру интерфейса, основные команды или органы управления, наиболее тесно связанные с функциональностью системы. При необходимости использовать *модели экрана*.

Если описывается интерфейс библиотеки подпрограмм, указать прототип каждой доступной внешнему пользователю функции, класса или структуры данных. Для каждой функции указать назначение и, где необходимо, смысл и диапазон допустимых значений параметров. В случае небольшого объёма библиотеки допустимо объединение с разделом 5, а для библиотеки классов и с разделом 4. В таком случае структурирование подразделов производится по функциям или классам.

При необходимости привести, динамическую диаграмму (например, *диаграмму состояний*).

Раздел «7. Прочие требования»

Указать требования к нефункциональным характеристикам системы, отличающиеся от общепринятых либо имеющие критическое значения для успешной реализации проекта. Любой из подразделов, а также и весь раздел можно опустить в случае отсутствия соответствующих требований.

Раздел «7.1. Требования к надёжности»

Оценить вероятность возникновения, возможные последствия и убытки от нештатных ситуаций (рисков), могущих возникнуть в связи с использованием системы, в том числе:

- отказ оборудования;
- обнаружение ошибок разной степени тяжести;

- утеря или повреждение данных;
- физическое нарушение безопасности;
- резкое изменение экономических условий или производственного процесса.

Сделать вывод о наличии или отсутствии необходимости мероприятий по управлению данными рисками. Перечислить и описать список мероприятий, для каждого мероприятия указать: процедуру проведения, предполагаемую периодичность, категории пользователей, ответственных за его выполнение. Например: «ежедневное резервное копирование базы данных на магнитную ленту производится в конце рабочего дня Оператором» или «для обеспечения физической безопасности сервера он должен быть помещён в охраняемое помещение, доступ к которому имеет только Администратор и Директор».

Отдельно указать требования к устойчивости программного продукта к различным нештатным ситуациям, действиям по обработке ошибок и восстановлению после сбоев. Например, «неожиданное отключение питания должно приводить к потере только данных только из тех записей БД, которые редактируются в этот момент». Если значительная часть требований к надёжности выполняется за счёт базового программного обеспечения, например, СУБД или ОС, указать это.

Раздел «7.2. Требования к безопасности»

Если система предназначена для хранения и обработки ценных данных либо для управления критическими процессами, рассмотреть отдельно риски, связанные со злоумышленным несанкционированным доступом к системе, в том числе:

- доступ на чтение к конфиденциальным данным (например, финансовые, медицинские записи);
- доступ на запись к открытым, но важным данным (например, данные об успеваемости, результаты выборов);
- перехват управления критическими процессами;
- разрушение данных и вандализм;
- временное нарушение работы системы без разрушения данных (Denial-of-Service).

Указать возможные векторы атаки и необходимые для предотвращения атак изменения функциональности (например, шифрование передаваемых или хранимых данных, проверка паролей пользователя на сложность, перенос административных функций из Web-интерфейса в локальное приложение и т. д.).

Раздел «7.3. Требования к производительности»

Определить диапазоны количественных параметров размеров задач, для решения которых предназначена система, в том числе объём базы данных, количество одновременно работающих пользователей, скорость поступления внешних сигналов и т. п.

В предположении, что система будет использовать рекомендуемую конфигурацию оборудования из раздела 2.1, указать минимальные требования к количественным показателям эффективности системы. В число показателей могут входить: время отклика, скорость выполнения транзакций,

время обработки запроса, объём используемой оперативной памяти и дискового пространства и др. Для каждого из показателей может быть указано предельно допустимое и ожидаемое среднее значение. Например, «среднее время отклика системы на действие пользователя должно быть 2–3 секунды, максимальное — не более 5 секунд». Если часть показателей зависит от внешних факторов, (например, качества TCP/IP соединения), отметить это и привести требования для случая их оптимального сочетания. Дополнительно возможно указание ожидаемых реальных показателей в различных условиях.

Раздел «8. Проект»

Раздел содержит данные о внутренней структуре программной системы и принятых при разработке решениях, не проистекающих непосредственно из требований. В отличие от требований, предложения в разделе «Проект» должны быть сформулированы не в побудительной, а в утвердительной форме, (например, вместо фразы «система *должна* состоять из 5 модулей» следует написать «система *состоит* из 5 модулей»). Некоторые подразделы могут отсутствовать, если реализация соответствующих требований тривиальна или непосредственно определяется самими требованиями.

Привести во всех подразделах все необходимые *статические и динамические диаграммы* (например, *диаграмму компонентов*).

Раздел «8.1. Средства реализации»

Перечислить и сравнить языки программирования, среды разработки, библиотеки программ, СУБД и т. п., рассматривавшихся в качестве возможных средств реализации системы. Учесть характеристики открытости и цены средства, скорости и простоты разработки, надёжности и эффективности конечного программного продукта. Если проводилось сравнительное тестирование или апробация средств, привести результаты. Документировать и обосновать осуществлённый выбор средств.

Раздел «8.2. Структуры данных»

Описать внутренние структуры данных в порядке, аналогичном спецификации данных. Описать, если необходимо, низкоуровневую структуру базы данных (в терминах таблиц и полей вместо сущностей и атрибутов). Привести точные и детализированные типы данных и описания форматов.

Описать нетривиальные структуры данных, используемые алгоритмами программной системы. Аргументировать их выбор.

Раздел «8.3. Модули и алгоритмы»

Описать общую структуру системы или каждой из подсистем. Перечислить, в зависимости от размера и сложности системы, все модули и классы и/или процедуры и функции с кратким описанием назначения каждого пункта. Описать внутренние интерфейсы между частями системы.

Если при разработке были применены или созданы нетривиальные алгоритмы, описать их. Для известных алгоритмов привести ссылки, для новых — сравнение с существующими и анализ их свойств, эффективности, преимуществ и недостатков.

Раздел «8.4. Стандарт кодирования»

Кратко описать используемый стандарт кодирования, указать его источник (например, принят в организации-заказчике, рекомендуется разработчиками используемого языка программирования), привести ссылку на литературу. Обратить внимание на уникальные для данного проекта элементы стандарта, особенно соглашения об именовании.

Если для оформления кода использовались автоматические инструменты (например, среды программирования с автоматической генерацией кода, программы автоматического форматирования кода), указать эти инструменты и параметры их вызова.

Раздел «8.5. Проект интерфейса»

Перечислить и обосновать дизайнерские решения, принятые в процессе разработки интерфейса. В том числе цветовую гамму, шрифты, размеры, форму и расположение элементов управления, и т. д. Привести, по меньшей мере, по одному *снимку экрана (screenshot)* для каждого существенно отличающегося модуля или режима работы. При получении снимков экрана следует:

- следить за соответствием версии программы, изображаемой на снимке, с версией, описываемой в отчёте;
- включать в снимок только окно программы, а не весь экран операционной системы (в частности, следует удалять изображение Windows taskbar и меню/панели инструментов браузера);
- уменьшать окно программы, особенно по вертикали, таким образом, чтобы оно не содержало пустого пространства;
- по возможности подбирать цветовую схему, хорошо отображающуюся на черно-белую печать (в частности, желательно избегать крупных участков с чёрным фоном, а также низкоконтрастного текста на светлом фоне).

Раздел «9. Реализация и тестирование»

Привести данные о физических характеристиках текущей версии системы:

- объём *написанного автором* кода в килобайтах и строках, отдельно по каждому языку программирования,
- для работ, заключающихся в изменении существующей системы, объём *изменённого автором* кода в строках,
- объём *автоматически сгенерированного* кода в килобайтах и строках, отдельно по каждому языку программирования (сюда входят, например, файлы *.dfm в системе Delphi),
- количество модулей, форм, экранов, страниц сайта и т. п.,
- количество автоматических тестов,
- количество и объём, в килобайтах, программных компонент,

- фактическое быстроедействие и затраты оперативной памяти, на нескольких примерах, сравнить с требованиями п. 7.2.
- фактическое количество пользователей, кроме разработчиков системы.

Если данная система разрабатывалась как часть более крупной, привести количественную (например, «около 15% кода») и качественную оценку вклада данной подсистемы.

Указать методику тестирования: по белому или чёрному ящику, бета-тестирование, случайное тестирование. Описать процедуру тестирования (вручную или автоматически), его объём и результаты.

Если проводилось внедрение, описать процесс внедрения, полученный технико-экономический эффект, отзывы пользователей. Сослаться на *Акт о внедрении*, который должен прилагаться к работе.

Сделать вывод об успешности реализации программной системы.

Если сложность описываемой системы невелика, данный раздел можно опустить, и перенести данные о характеристиках системы и её внедрении в «Заключение».

Раздел «9.1. Вычислительный эксперимент»

Данный раздел наиболее характерен для научно-исследовательских работ и часто является центральной частью таких работ.

Следует указать цели эксперимента, экспериментальную гипотезу (если есть). Типичными целями являются: демонстрация возможности (приблизительного) решения поставленной задачи разработанным алгоритмом, подбор оптимальных параметров алгоритма, оценка производительности и оптимизация, сравнение различных вариантов алгоритма, оценка качества прогнозирования, выполняемого алгоритмом.

Описать методику проведения эксперимента, обратить особое внимание на возможность её воспроизведения независимыми исследователями. При исследовании производительности — подробно описать использованную программно-аппаратную конфигурацию и методику измерений.

Привести результаты эксперимента в виде набора таблиц и графиков. При построении графиков, особенно сравнительных, обратить внимание на корректный выбор масштаба по осям.

Проанализировать результаты, сделать выводы о достижении целей эксперимента. Если целью эксперимента был подбор оптимальных параметров или вариантов алгоритма, перечислить полученные рекомендации.

Раздел «Заключение»

Раздел не нумеруется. Начать раздел фразой «Таким образом, в процессе курсовой / дипломной / др. работы мною было ...», за которой перечислить виды деятельности, выполненные в рамках работы. Отделить в списке учебную деятельность («изучено», «углублены знания / повышены навыки в области...») от производственной («разработано», «спроектировано», «реализовано», и т.п.)

В перечислении избегать общих выражений («изучена предметная область»), а вместо этого использовать конкретные («изучены основы банковского дела и схема работы на примере банка Х»).

Кратко перечислить основные характеристики и достоинства разработанной системы, привести данные о её внедрении и достигнутом за счёт него эффекте, указать пути дальнейшего развития системы. В случае отсутствия п. 8 «Реализация», привести краткие сведения об объёме и сложности системы.

Раздел «Список литературы»

Раздел не нумеруется. Содержит данные о печатных работах и Internet-ресурсах, относящихся к описываемой работе. Форматируется в виде нумерованного списка, упорядоченного *в алфавитном порядке названий*. Ссылки на литературу в тексте оформляются как номер, заключённый в квадратные скобки. Желательно, чтобы на каждый элемент списка литературы была, по крайней мере, одна ссылка в тексте.

Элементы списка формируются согласно стандарту [5] и состоят из следующих частей:

1. списка авторов (Фамилия И. О.),
2. названия на русском языке или языке оригинала,
3. издательства для печатных работ, (например, М.: Просвещение) либо
держателя копирайта для электронных документов (например, (с) Microsoft Corp.),
4. года либо, если этого требует политика публикации документа, даты
(например, октябрь 1999),
5. объёма в страницах — для печатных документов либо электронных с
постраничным форматированием (например, PDF),
6. для электронных документов или печатных документов, доступных также
в Internet — URL с указанием протокола
(например, http://imcs.dvfu.ru/lib/get_connected.pdf).

В случае если URL содержит изменяющуюся часть (например, аргументы скрипта для выбора документа из БД), допустимо привести неполный путь (возможно, с указаниями по получению нужного документа).

Приводить ссылку на корневой каталог сайта нежелательно, даже если его тематика соответствует теме работы, вместо этого следует указать ссылки на конкретные документы внутри сайта, и оформить их согласно приведённым правилам. Предполагается, что читатель имеет квалификацию, достаточную для выделения из URL имени сайта.

В случае указания ссылок на архивы дискуссий в сети интернет (списки рассылки, форумы, wiki-сайты) следует:

1. объединить такие ссылки в отдельный подраздел;
2. использовать тему дискуссии как название работы;
3. ссылаться на конкретное обсуждение (thread, тему форума, talk-страницу wiki).

Раздел «Приложение»

Приложения нумеруются отдельно, «Приложение 1», «Приложение 2» и т. д.

Примеры исходного текста программы, а также входных и выходных данных могут быть включены в приложения. Полный исходный текст может быть приведён в отчёте только по специальному требованию руководителя и для небольших систем.

Исходный код

Исходный код системы обычно не является частью отчёта и не демонстрируется во время защиты. Однако руководитель перед защитой просматривает исходный код с целью определения общего качества кодирования и степени самостоятельности защищающегося, анализа личного вклада каждого участника в совместную разработку.

Поскольку выработка стиля кодирования является важной компонентой образования на специальности «Системное программирование», правильное оформление исходного кода является необходимым условием для допуска к защите работы.

Требования к оформлению исходного кода могут определяться корпоративными стандартами организации-заказчика. Этот факт должен быть отражен в тексте работы, а сами требования — включены в список литературы.

При отсутствии внешних требований следует придерживаться приведённых ниже рекомендаций, а также рекомендаций, изложенных в литературе (см., например, [7]). В любом случае, принятые соглашения о кодировании должны быть документированы в отчёте, в разделе «Общие требования» либо «Проект».

Форматирование кода

Исходный код на должен быть оформлен в едином в пределах системы стиле. Форматирование кода включает:

- выделение блочной структуры — все блоки должны быть выделены отступом от 1 до 8 пробелов (рекомендуется 2);
- использование пробелов в выражениях и операторах (рекомендуется расставлять пробелы максимально близко к правилам естественного языка, например, после запятых и точек с запятыми, вокруг знаков операций, снаружи скобок, не являющихся вызовом функции);
- группировка операторов по строкам (рекомендуется размещать не более одного оператора в строке);

- разбиение длинных операторов на несколько строк (рекомендуется выделять перенесённую часть отступом);
- форматирование табличных структур.

Именованние объектов

Объекты, используемые в программе, должны именоваться согласно единой схеме, включающей соглашения о следующем:

- разделение слов внутри идентификатора — при помощи подчёркивания либо написания каждого слова с заглавной буквы;
- оформление идентификаторов различных видов (имён констант, переменных, функций, классов), использование заглавных и прописных букв, специальных префиксов;
- использование частей речи и устойчивых словосочетаний (например, имена логических функций начинаются на «is»);
- язык идентификаторов — рекомендуется использовать правильный английский либо, либо для систем, которые это допускают, (например, 1C или MS Access), правильный русский язык; не рекомендуется использовать транслитерацию и некорректные английские слова и выражения;
- наличие в имени информации о типе (венгерская нотация).

Внутренняя документация

Комментарии в тексте программы должны быть написаны на одном и том же естественном языке (русском или английском) и единообразно оформлены. Отметить использование системы автоматизированного документирования (например, JavaDoc, Doxygen [1])

Доклад

Защита работы производится в форме доклада студента на заседании кафедры или, для дипломной работы, на заседании ГАК. Необходимым условием для выхода на защиту является допуск от руководителя работы. Докладчик должен представить отчет, оформленный в соответствии с изложенными требованиями, распечатанный и подшитый и, желательно, упакованный в пластиковую папку. Заявление, отзыв, рецензия и тому подобные прилагающиеся документы не подшиваются к основному тексту работы.

Продолжительность доклада должна укладываться в интервал, указанный в Табл. 1

Табл. . Продолжительность доклада

Вид работы **Продолжительность, мин**

Реферат	10–80	Летняя практика	3–5	Курсовая работа (предзащита)	8–10	Курсовая работа (защита)	7
Преддипломная практика	5–10	Дипломная работа (предзащита)	8–12	Дипломная работа (защита)	7–10		

Доклад должен начинаться фразой «Защищается (докладывает) студент *NNN* группы *Фамилия Имя Отчество* по теме *Название работы*; руководитель *научная степень должность Фамилия Имя Отчество*», и заканчиваться фразой «спасибо за внимание». Доклад может быть прерван в случае истечения отведённого времени или по решению председателя комиссии в связи с явной неготовностью докладчика.

Докладывать следует ясным, достаточно громким голосом, с соблюдением правил культуры речи и дикции [6]. В случае возникновения трудностей с дикцией или связным изложением мыслей следует провести несколько тренировок, особенно во время предварительных защит.

Примерный план доклада соответствует структуре отчёта. При этом докладчик должен сконцентрировать изложение на наиболее важных моментах, опуская детали и упоминая о малозначительных моментах лишь поверхностно. В случае если у комиссии возникнут вопросы, они будут заданы защищаемому после доклада.

Рекомендуется подготовить текст доклада заранее и согласовать его с руководителем. Текст или план доклада разрешается использовать как вспомогательный материал во время выступления, однако не рекомендуется заменять доклад чтением.

Во время выступления докладчик должен большую часть времени сидеть или стоять лицом к комиссии, и смотреть на членов комиссии, по мере необходимости оглядываясь на демонстрационный материал.

Доклад должен сопровождаться демонстрацией наглядного материала одного из следующих видов:

1. Плакатами, отпечатанными или нарисованными от руки на листах формата

A2 или A1. Допускается также склеивание плакатов из отдельных листов меньшего формата. Плакаты должны быть пронумерованы и подписаны, например «Плакат 1. Общие требования к системе». Данный вид сопровождения является устаревшим и допускается только в исключительных случаях.

2. Презентацией с раздаточным материалом — предпочтительный вид

демонстрации, более подробно описан в разделе «Презентация».

3. Демонстрацией программной системы — допускается в исключительных

случаях на ранних этапах обучения, либо как *дополнение* к одной из предыдущих форм демонстрации.

Презентация

Общее описание

Компьютерная презентация является предпочтительной формой демонстрационного материала. Презентация предназначена для иллюстрации излагаемого докладчиком материала, облегчению его восприятия слушателями, более полного раскрытия отдельных вопросов. Презентация состоит из последовательности слайдов, демонстрируемых один за другим по мере продвижения доклада.

Для создания и демонстрации слайдов используется компьютер со специальным программным обеспечением (например, Microsoft PowerPoint или Open Office Impress; допустимы также презентации в формате PDF). В качестве устройств вывода применяется либо монитор с достаточно большой диагональю (не менее 17", предпочтительно 21" и более), либо специальный проектор. Докладчик совместно с руководителем должен заблаговременно позаботиться о надлежащем техническом обеспечении презентации.

Преимущества презентаций состоят в том, что они:

- обеспечивают визуальный канал передачи информации в дополнение к звуковому,
- облегчают структурирование доклада, снижают вероятность ошибок и пропуска тем,
- отвлекают внимание слушателей от возможных недостатков позы и жестикуляции докладчика,
- обладают средствами для ограничения времени показа слайдов, что позволяет жёстко и непрерывно контролировать скорость доклада,
- позволяют иллюстрировать изложение наглядно и динамично, с возможностью использованием анимационных эффектов.

Презентация является дополнительным иллюстративным материалом, и не должна подменять собой доклад. В частности, не следует выносить на слайды целиком фразы и абзацы из отчёта.

Структура

Презентация должна начинаться со слайда, содержащего информацию с титульного листа работы, и заканчиваться слайдом с информацией из заключения. Чтобы упростить обращение к докладчику для членов комиссии, фамилия имя отчество докладчика и руководителей на титульном слайде должны быть указаны полностью.

Каждый слайд, кроме титульного, должен иметь заголовок и номер и быть рассчитан на показ в течение 40–60 секунд, но не менее 20 секунд и не более 2 минут. Отсюда общее количество слайдов в зависимости от продолжительности доклада должно составлять 10–20 штук.

Для сокращения времени доклада, по согласованию с руководителем, допускается не описывать подробно отдельные слайды, содержащие сложную и непринципиальную для защиты информацию. Например, «на данном слайде изображена архитектура вспомогательной подсистемы NNN, более подробно описанная в отчёте». Если у комиссии возникнет такое желание, подробности могут быть изложены в качестве ответа на вопрос.

После окончания доклада следует оставлять на экране слайд с заключением.

Оформление

Текст на слайдах должен быть кратким и крупным, с использованием шрифта без засечек кеглем не менее 28, а желательно 32–40. Общий объем текста на слайде не должен превышать 20–25 слов, а желательно не более 10 слов. Текст на слайде должен представлять одно, редко — два предложения.

В случае необходимости перечисления следует оформить его как нумерованный или маркированный список, содержащий не более 5–6 пунктов, причём каждый пункт описать 2–3

словами. Например: «Достоинства системы:

1. Высокое быстродействие
 - 2) Малые системные требования
 - 3) Дружественный интерфейс».
- Во время доклада следует раскрыть каждый из пунктов более подробно. Возможно использование анимационных эффектов для последовательного вывода на экран пунктов перечисления. При этом предпочтение следует отдавать простым эффектам, сосредотачивающим внимание на появившемся пункте, а не на самой анимации. Кроме того, не рекомендуется использовать несколько различных эффектов в одной презентации.

Ни в коем случае не следует использовать звуковые эффекты, а также *повторяющуюся* анимацию и мигание. Нежелательно также использовать эффекты перехода между слайдами.

При изображении на слайде схем и диаграмм следует ограничивать число одновременно изображённых объектов до 5–6. Сами объекты должны быть достаточного размера и иметь чёткие границы. Надписи на объектах должны иметь тот же размер, что и остальной текст. Линии и стрелки, соединяющие объекты, должны быть чётко видимыми и достаточно широкими. При изображении на слайде диаграмм UML рекомендуется, в отличие от печатных документов, использовать соединительные линии толщиной в 3–4 пункта.

В целом изображение презентации должно быть достаточно ярким и контрастным для вывода на проектор поверх различных поверхностей, в том числе приводе на проекционный экран, стену и классную доску. Рекомендуется использовать чёрный текст на белом или бледном цветном фоне. Не допускается использование низкоконтрастных сочетаний, например «чёрный на синем» или «жёлтый на белом». Не рекомендуется использовать градиентные переходы между сильно различающимися цветами.

Более подробно вопросы подбора цветовой гаммы и композиции рассмотрены, например, в [8]. Рекомендуется пользоваться стандартными шаблонами, имеющимися, например, в программе Microsoft PowerPoint.

В презентацию рекомендуется включать снимки экрана, иллюстрирующие пользовательский интерфейс и процесс работы с программой. Требования к снимкам аналогичны требованиям, изложенным для снимков в отчёте. Часто размер интерфейсных элементов программы недостаточен для хорошей видимости на слайде. В таком случае из снимка следует выделить части, соответствующие основным областям интерфейса, и показать их необходимой степенью увеличения на нескольких слайдах или с использованием выносок.

Раздаточный материал

Основные недостатки презентаций — это

- часто неидеальная видимость из-за недостаточного размера монитора, низкой резкости проектора, неудачного освещения и т. п., и
- последовательное появление слайдов (в отличие от плакатов), что делает невозможным независимый от докладчика просмотр демонстрационных материалов, особенно во время вопросов.

Первый недостаток преодолевается тщательной настройкой оборудования, подготовкой помещения и следованием правилам оформления презентаций.

Для борьбы со вторым недостатком в качестве *обязательного* элемента презентаций присутствует раздаточный материал — листы формата A4 с распечатанным изображением наиболее важных иллюстраций. Листы должны быть подписаны и пронумерованы, например «Лист 1. Архитектурно-контекстная диаграмма».

В простейшем случае раздаточный материал должен быть получен путём распечатки всех или части слайдов средствами редактора презентаций. Если слайды используют тёмный фон, при распечатке их рекомендуется инвертировать. Допускается размещение двух или четырёх страниц презентации на одной странице раздаточного материала, если размер шрифтов и графических элементов презентации позволяет сохранить читабельность при таком масштабе.

Раздаточный материал должен быть распечатан, по крайней мере, в двух экземплярах, а желательно — по экземпляру на каждого члена комиссии. Раздаточный материал служит также запасным вариантом на случай технических проблем во время презентации.

Список литературы

1. van Heesch D. Doxygen Manual, 2003, <http://www.stack.nl/~dimitri/doxygen/manual.html>
2. ISO/IEC 7498-1:1994, Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model., 59 pp.
3. OMG Unified Modeling Language Specification. Version 2.2, (c) Object Management Group, 2009, 566 pp., <http://www.omg.org/technology/documents/formal/uml.htm>
4. Postel J., Reynolds J. Instructions for RFC authors, RFC 2223, October 1997, <ftp://ftp.isi.edu/in-notes/rfc2223.txt>
5. ГОСТ 7.1-84 Библиографическое описание документа, <http://imcs.dvfu.ru/lib/repplan/gost7p1-84.rtf>
6. Данцев А. А., Нефедова Н. В. Русский язык и культура речи для технических вузов. — Ростов н/Д: Феникс, 2001.
7. Кернинг Б., Пайк Р. Практика программирования/ Пер. с англ. — СПб.: Невский диалект, 2001.
8. Кирсанов Д. Веб-дизайн: книга Дмитрия Кирсанова. — СПб.: Символ-плюс, 2001
9. Кленин А. С. Технология программирования: программа курса, ДВГУ, 2003, <http://imcs.dvfu.ru/ru/courses/progtech>
10. Слободнюк Р. С. (ред.) Методические указания к дипломной работе для студентов специальности «Прикладная математика», Владивосток: Изд-во ДВГТУ, 1995
11. Положение об итоговой государственной аттестации выпускников ДВГУ, Владивосток, Изд-во ДВГУ, 2006

Приложение 1. Примерный текст Акта о внедрении

Акт о внедрении

Настоящим удостоверяется, что программная система «Название системы», разработанная *Фамилия Имя Отчество* введена в эксплуатацию и используется в «*название организации-заказчика*» с «*дата внедрения*». В результате использования системы была (повышена эффективность производственного процесса, решена научная задача, и т. п.).

«*Должность представителя заказчика*» «*Фамилия И. О.*»

Акт должен быть заверен подписью и печатью.