

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Пономарева Светлана Викторовна
Должность: Проректор по УР и НО
Дата подписания: 18.09.2023 19:29:34
Уникальный программный ключ:
bb52f959411e64617366e12977697e8715981820



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)
АВИАЦИОННО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ**

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

по выполнению курсовой работы
по МДК 02.01 "Технология разработки программного обеспечения "
профессионального модуля ПМ 02. «Осуществление интеграции программных модулей»

для специальности 09.02.07
Информационные системы и программирование

г. Ростов-на-Дону
2022 г.

СОДЕРЖАНИЕ

	ВВЕДЕНИЕ	4
1	Курсовое проектирование	4
1.1	Цели курсового проектирования	4
1.2	Этапы курсового проектирования	5
2	Пояснительная записка	6
3	Оформление курсовой работы	11
3.1	Общие правила оформления	11
3.2	Комплектация пояснительной записки	12
3.3	Нумерация	13
3.4	Рисунки, схемы	13
3.5	Таблицы	14
3.6	Формулы и уравнения	14
3.7	Ссылки	14
3.8	Приложения	14
3.9	Подписи	15
4	Защита	15
	ПРИЛОЖЕНИЯ	
	Титульный лист	16
	Пример листа «Содержание»	17
	Стандарты описания программ	18
	Примеры тест-кейса	21

ВВЕДЕНИЕ

В профессиональном модуле ПМ.02 в МДК 02.01 «Технология разработки программного обеспечения» предусмотрено выполнение курсовой работы.

Важнейшей целью данной курсовой работы является приобретение практических навыков в создании программных продуктов реальных или почти реальных задач с использованием современного инструментария. Кроме того важной задачей является приобретение навыков в составлении текстовых документов, в частности, пояснительной записки к курсовой работе. Подобные навыки окажутся полезными как в процессе написания дипломной работы, так и в последующей трудовой деятельности будущих специалистов.

В данной методической разработке представлены основные положения, касающиеся как содержания курсовой работы, так и правил оформления курсовой работы. Следование предложенным указаниям позволят выполнить и защитить курсовую работу на достаточно высоком уровне.

Курсовое проектирование в рамках указанной дисциплины является, по сути, подготовкой и «генеральной репетицией» дипломного проектирования.

Хотя предлагается достаточно строго структурированная схема пояснительной записки, это не ограничивает творческую самостоятельность студента, это в полной мере относится, прежде всего, к созданию программного продукта. В этом - полная творческая свобода. Однако описание процесса и результата создания программного продукта должно быть формализовано и строго обосновано. Данная методическая разработка может быть рекомендована студентам в качестве пособия по выполнению курсовой работы в рамках МДК 02.01 «Технология разработки программного обеспечения», а также подготовке документов и для других учебных дисциплин.

1 КУРСОВОЕ ПРОЕКТИРОВАНИЕ

1.1 ЦЕЛИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Курсовое проектирование по МДК 02.01 «Технология разработки программного обеспечения» является завершающим и определяющим этапом данного междисциплинарного курса, позволяющим применить и закрепить полученные теоретические знания и практические умения.

Учебные, методические и воспитательные цели курсовой работы:

1. Обучение студентов профессиональным умениям; углубление, обобщение, систематизация и закрепление теоретических и практических знаний по курсу учебной дисциплины и применение этих знаний при решении конкретных научных и экономических задач;

2. Формирование и развитие у студентов умений и навыков самостоятельного умственного труда при изучении специальных дисциплин; комплексная проверка уровня знаний и умений; овладение методикой исследования и решение проблем и вопросов, разрабатываемых в курсовой работе.

3. Воспитывать у студентов познавательную деятельность, развивать творческое мышление, навыки самостоятельного исследования, научного анализа, творческого и практического подхода к полученным знаниям на примере конкретного предприятия, региона, страны исторического периода.

Время: 20 часов.

План выполнения курсового проектирования:

№ п/п	Учебные вопросы	Время (мин)
1.	АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ, ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	2ч.
2.	ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ СИСТЕМЫ В СООТВЕТСТВИИ С ПОСТАВЛЕННОЙ ЗАДАЧЕЙ	2ч.
3.	РАЗРАБОТКА ПО (КОДИРОВАНИЕ), ТЕСТИРОВАНИЕ, ОТЛАДКА СОЗДАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	12ч
4.	ОФОРМЛЕНИЕ РЕЗУЛЬТАТОВ ПРОЕКТИРОВАНИЯ В СООТВЕТСТВИИ С ПРАВИЛАМИ ОФОРМЛЕНИЯ ДАННОГО ВИДА ДОКУМЕНТОВ	2ч.
5.	ЗАКЛЮЧИТЕЛЬНАЯ ЧАСТЬ, ПОДГОТОВКА К ЗАЩИТЕ, ЗАЩИТА	2ч.

На курсовое проектирование отводится 20 часов. Целью курсовой работы является разработка программного обеспечения по решению задачи, максимально приближенной к реальным условиям, или даже являющейся работой по заказу. Разработанное программное обеспечение должно удовлетворять современным требованиям, быть надежным, удобным в эксплуатации. В процессе конструирования программного продукта необходимо следовать современным принципам программной инженерии, которые охватывают все этапы жизненного цикла программного изделия. Особое внимание необходимо уделить анализу предметной области, этапу проектирования, а также тестирования программного продукта, используя при этом методы, модели и инструментарий современной программной инженерии. Допускается групповая форма (не более 5 человек) выполнения курсовой работы; в этом случае при создании программного продукта должны использоваться такие модели конструирования, как модель RAD, XP. Предлагается широко использовать СУБД, технологию ADO, компонентные модели. Обязательным является использования языка UML для составления некоторых диаграмм, в частности, диаграмм вариантов использования при анализе требований к программному продукту.

Задачи ориентированы на имеющиеся в колледже технические средства и могут быть реализованы в любой современной системе программирования.

1.2 ЭТАПЫ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Работа над курсовым проектом может быть разбита на ряд этапов:

1. Исследование предметной области. Анализ поставленной задачи, отработка всех нюансов, уточнение исходных условий. Результатом должно стать ясное и глубокое понимание сущности поставленной задачи. Процесс анализа сопровождается созданием диаграмм UseCase на UML. Цель этапа — предоставить четкий список не дублируемых требований к системе, которые должны быть выделены из избыточных и частично дублирующихся сценариев и пользовательских историй.
2. Проектирование системы в целом, в терминах естественного языка. Именно на этом этапе должна быть продумана структура и функциональные возможности будущего программного продукта. Уже на этом этапе может быть спроектирован интерфейс, определяющий функциональные возможности ПП. Определяются цели и критерии будущего ПП. Выполняется структурирование задачи на подзадачи, моделируется схема управления между подзадачами, осуществляется декомпозиция на модули. На этом этапе необходимо: сформировать диаграмму связей (mindmap) - графическую схему взаимодействия объектов (модулей, страниц и т.д.) проектируемого ПО; произвести

прототипирование основных экранных форм (например с использованием одного из онлайн сервисов: www.ninjamock.com, www.moqups.com); выбрать архитектуру разрабатываемого ПО (автономные, двух-звенные, многозвенные, CORBA, SOA, REST и т.д.), сформировать графическую схему, описать её структуру и основные элементы; выбрать и обосновать выбор базы данных (исходя из модели данных) и СУБД, описать таблицы данных, структуру хранимых данных; исходя из обрабатываемых и хранимых данных в ПО, рассмотреть методы обеспечения информационной безопасности, особенно при хранении информации содержащей персональные данные пользователей.

3. Разработка ПО... На этом этапе происходит конкретизация определившейся на 2-м этапе структуры. Необходимо выбрать и обосновать выбор инструментальных средств разработки, средств хранения кода и т.д. Этот этап требует хороших знаний выбранного инструментария разработки. В процессе программирования (или в условиях объектно-ориентированного инструментария – при моделировании) должны быть учтены следующие моменты
 - определение приоритета целей (удобный пользовательский интерфейс или эффективность программ по времени, или использование памяти и пр.)
 - использование идей защитного программирования, блокировка и прогнозирование ошибок;
 - соблюдение хорошего стиля программирования (имена, листинги и пр.).
4. Тестирование и отладка ПП. Необходимо выбрать метод проверки качества (ручное/автоматизированное тестирование, интеграционное, нагрузочное, функциональное, А/В тестирование), обосновать выбор. Необходимо спроектировать тестовые варианты, т.е. информационное содержание всех файлов (Б.Д.), данных интерактивного ввода, значения логических переменных, и т.д. - что обеспечивает прослеживание всех логических цепочек и охватывает все возможные и даже исключительные, экстремальные ситуации. Предусмотреть как структурное тестирование («белого ящика»), так и функциональное тестирование («черного ящика»). Для функционального тестирования использовать диаграммы причин-следствий, позволяющие наиболее полно учесть все возможные функциональные особенности созданного программного продукта. Выявленные ошибки необходимо фиксировать, так как количественная информация об ошибках используется в метриках оценки созданного программного продукта.
5. Комплексная отладка, опытная эксплуатация созданного ПП.
6. Расчет функциональности созданного программного проекта на основе функциональных точек (FP) с учетом коэффициентов регулировки сложности F_i .
7. Расчет метрик программного проекта (производительность, качество) на основе рассчитанного значения FP (пункт 6).
8. Написание пояснительной записки. Пояснительная записка должна в полной мере отражать суть, процесс, средства и результаты решения задачи, должна содержать все описания, таблицы, структуры, исходные тексты программ, выходные документы; должна быть грамотно и четко составлена в соответствии со стандартами оформления текстовой информации..
9. Защита курсовой работы. Демонстрируется созданный программный продукт, проверяется пояснительная записка. Возможна организация публичной защиты курсовых работ.

2 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Курсовая работа не только должна быть содержательной и самостоятельной, но и должна быть правильно оформлена.

Примерный план пояснительной записки:

ВВЕДЕНИЕ

1. Анализ и формирование требований к ПО
2. Проектирование программного обеспечения
 - 2.1 Архитектура и структура проектируемого ПО
 - 2.2 Информационная модель
3. Разработка программного обеспечения
 - 3.1 Инструментальные средства разработки ПО
 - 3.2 Описание ПО
 - 3.3 Руководство пользователя
 - 3.4 Тестирование ПО
4. Информационная безопасность
5. Расчет метрик

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЯ

ВВЕДЕНИЕ

Рекомендуется охарактеризовать существующие технологии программирования, дать анализ возможных применений стратегий и моделей конструирования ПП, показать актуальность поставленной задачи.

1. Анализ и формирование требований к ПО содержит суть поставленной задачи, логику получения основных выходных данных, анализ предметной области. Определяются наиболее важные для заказчика требования к:

- функциональности создаваемого программного обеспечения;
- сценариям использования проектируемого ПО;
- дизайну;
- портрету пользователя.

Производится тщательное исследование предметной области. Инструментарием исследования выступает унифицированный язык визуального моделирования UML, с помощью которого строятся диаграммы UseCase. Анализ всех вариантов использования позволит достаточно хорошо исследовать предметную область. (см. пункт 1.2 «Этапы курсового проектирования»).

2. Проектирование программного обеспечения состоит из двух подразделов:

2.1 «Архитектура и структура проектируемого ПО» - необходимо выбрать и описать архитектуру разрабатываемого ПО (автономные, двух-звенные, многозвенные, CORBA, SOA, REST и т.д.), сформировать графическую схему взаимодействия объектов (модулей, страниц и т.д.) проектируемого ПО, описать её структуру и основные элементы. Показать возможности применения прототипирования основных экранных форм.

2.2 «Информационная модель» для задач обработки информации является ключевым пунктом, позволяющим понять суть задачи, полноту и эффективность ее решения. Информационная модель предполагает полный анализ входной, выходной информации.

Необходимо выбрать и обосновать выбор базы данных (исходя из модели данных) и СУБД, описать таблицы данных, структуру хранимых данных. Привести информационную модель в виде ER-модели. Исходя из обрабатываемых и хранимых данных в ПО, рассмотреть методы обеспечения информационной безопасности, особенно при хранении информации, содержащей персональные данные пользователей.

В зависимости от тематики курсовой работы этот пункт может быть расширен за счет более полного описания входной и выходной информации.

Входная информация. Необходимо представить входную информацию, включая первичные документы; описать, кто, когда и как их заполняет, какая информация из них участвует в

решении задачи. Далее необходимо представить структуры созданных файлов или баз данных, а также дать пояснения к созданным структурам

Выходная информация. Необходимо представить перечень выходных документов, их содержание, структуру, порядок выдачи. Листинги выходных документов можно привести в Приложении, на которое можно ссылаться. Надо описать запросы, т.е. информацию, получаемую в результате интерактивного диалога с пользователем. Сюда же можно отнести различную информацию справочного характера, сообщения об ошибках, информацию о ходе выполнения программы и т.п

3. Разработка программного обеспечения должна быть описана подробно. *Оформление программного текста, описания программы, описание применения программы должно выполняться в соответствии с соответствующими стандартами*

В «**Инструментальных средствах разработки ПО**» необходимо обосновать выбор инструментальных средств разработки, средств контроля версионности, средств хранения кода и т.д.. Показать, какие средства и приемы использовались для повышения надежности и качества разработки программного продукта.. Дать краткую характеристику и принципы работы используемой системы программирования, системы управления базами данных и т.д. Обосновать выбор данного инструментария.

В пункте «**Описание программного продукта**» необходимо привести внешнюю структуру программного продукта (структуру интерфейса, показывающую, как выглядит программный продукт с точки зрения пользователя), а также внутреннюю структуру (взаимодействие отдельных модулей, объектов); привести характеристику отдельных составляющих (модулей, процедур, объектов). Необходимо представить внутреннюю структуру пакета, т.е. взаимодействие отдельных модулей, объектов. Целесообразнее всего представить структуру в виде компонентной диаграммы на UML, что покажет наиболее полно все составляющие программного проекта. Учитывая инструментарий разработки, дать характеристику модульной или объектно-ориентированной технологии программирования. Показать, какие средства и приемы использовались для повышения надежности программного продукта, для защиты от несанкционированного доступа, удобства эксплуатации. Необходимо привести характеристику отдельных составляющих (модулей, процедур, объектов). Т.к сами листинги программ приведены в Приложении, этот пункт можно рассматривать как комментарий к листингам, при этом обязательны конкретные ссылки на приложения.

В пункте «**Руководство пользователя**» необходимо в понятных пользователю терминах представить инструкцию по запуску и эксплуатации программного продукта, отразить необходимые ресурсы (память, требования к технике), отметить исключительные ситуации, пояснить возможные сообщения программы, показать функциональные возможности программного продукта. Вполне вероятно, что изложение материала данного пункта окажется в большой степени связанным с интерфейсом программного продукта, который, возможно, был описан ранее, поэтому ссылки на представленные ранее схемы и рисунки вполне уместны.

В пункте **Тестирование ПО** необходимо охарактеризовать выбранный метод проверки качества (ручное / автоматизированное тестирование, интеграционное, нагрузочное, функциональное, А/В тестирование), обосновать выбор; представить разработанные тест-кейсы проверки качества ПО; представить протокол тестирования (ошибки, исправление, итерационное тестирование). Показать принципы формирования тестовых вариантов для структурного и функционального тестирования. Проектирование тестовых вариантов для функционального тестирования осуществлять с использованием метода диаграмм причинно-следственных связей. Спроектированные диаграммы, таблицы обязательно привести в тексте пояснительной записки. Выявленные ошибки свести в таблицу:

Таблица 1 – Выявленные ошибки

Ошибки проектирования	Ошибки кодирования	Синтаксические ошибки

Показать, какие методы отладки использовались (аналитические, экспериментальные). Провести тестирование всех 4-х уровней – от тестирования элементов до тестирования системного. Показать возможности восстановления созданного ПП, результаты стрессового тестирования. Если подобные возможности не реализованы, указать предполагаемые проблемы эксплуатации созданного программного обеспечения. Результаты свести в таблицу произвольного формата.

4.«Информационная безопасность» отражает требования к информационной безопасности, а также средства и способы ее реализации. Описать средства аутентификации, авторизации, хранения паролей, шифрование и пр.

5.Расчет метрик

Целесообразно использовать функционально-ориентированные метрики для оценки процесса конструирования, для чего необходимо рассчитать функциональность созданного ПП с учетом коэффициентов сложности. Для расчета необходимо составить таблицы:

Таблица 2 – Расчет общего количества FP

Модули	Информационные характеристики проекта					Кол-во методов (алгорит.)	Итого по модулю
	Вн.вводы	Выводы	Запросы	Внешн. файлы	Интерф. файлы		
					Всего:		

Таблица 3 - Исходные данные для расчёта указателя свойств

№	Характеристика	Количество	Сложность	Итого
1	Вводы	□	* 4	= □
2	Выводы	□	* 5	= □
3	Запросы	□	* 4	= □
4	Логические файлы	□	* 7	= □
5	Интерфейсные файлы	□	* 7	= □
6	Количество алгоритмов	□	* 3	= □
<i>Общее количество</i>				= □

$$FP = \text{Общее количество} * (0,65 + 0,01 * \sum_{i=1}^{14} Fi),$$

где F_i – коэффициенты регулировки сложности.

Каждый коэффициент может принимать следующие значения: 0 – нет влияния, 1 – случайное, 2 – небольшое, 3 – среднее, 4 – важное, 5 – основное.

Значения выбираются эмпирически в результате ответа на 14 вопросов, которые характеризуют системные параметры приложения (таблица 4).

Таблица 4 - Определение системных параметров приложения

№	Системный параметр	Описание
1	Передачи данных	Сколько средств связи требуется для передачи или обмена информацией с приложением или системой?
2	Распределённая обработка данных	Как обрабатываются распределённые данные и функции обработки?
3	Производительность	Нуждается ли пользователь в фиксации времени ответа или производительности?
4	Распространённость используемой конфигурации	Насколько распространена текущая аппаратная платформа, на которой будет выполняться приложение?
5	Скорость транзакций	Как часто выполняются транзакции? (каждый день, каждую неделю, каждый месяц)
6	Оперативный ввод данных	Какой процент информации надо вводить в режиме онлайн?
7	Эффективность работы конечного пользователя	Приложение проектировалось для обеспечения эффективной работы конечного пользователя?
8	Оперативное обновление	Как много внутренних файлов обновляется в онлайн-транзакции?
9	Сложность обработки	Выполняет ли приложение интенсивную математическую или логическую

		обработку?
10	Повторная используемость	Приложение разрабатывается для удовлетворения требований одного или многих пользователей?
11	Лёгкость инсталляции	Насколько трудны преобразование инсталляция приложения?
12	Лёгкость эксплуатации	Насколько эффективны и/или автоматизированы процедуры запуска, резервирования и восстановления?
13	Разнообразные условия размещения	Была ли спроектирована, разработана и поддержана возможность инсталляции приложения в разных местах для различных организаций?
14	Простота изменений	Была ли спроектирована, разработана и поддержана в приложении простота изменений?

Таблица 5 – Коэффициенты регулировки сложности

Коэф. сложн.	0	1	2	3	4	5
Системн. параметр						
1.						
2.						
...						
14.						
Всего:						

На основе найденного FP определяют производительность и качество проекта, причем трудозатраты (чел-час) тщательно учитываются во все время конструирования по всем этапам.

Производительность = Функция / Затраты [FP / чел.-мес.];

Качество = Ошибки / Функция [Единиц / FP];

Данные по трудозатратам также необходимо свести в следующую таблицу:

Этап конструирования	Трудозатраты
Всего:	

Окончательные результаты необходимо свести в следующую таблицу:

Трудозатраты	Кол-во ошибок	FP	Производительность	Качество

ЗАКЛЮЧЕНИЕ

Дать анализ проделанной работы, отметить результаты и метрики конструирования, достоинства созданного ПП, возможности его развития, практического применения. Отметить приобретенный практический опыт.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Список литературы обычно упорядочивается либо по алфавиту, либо по степени важности. Например :

Иванов И.И Программирование / Учебник –2-е изд.- М : Мир,2006.-287с.

Петров И.И. FOX-PRO.– М : Бином, 2007.– 342с.

Перед городом – ставится точка и дефис. После города перед издательством – двоеточие, после названия книги перед указанием характера книги – справочник, учебник, сборник и т.д. ставится / (слэш). В конце описания – объем в страницах.

ПРИЛОЖЕНИЯ

В приложения следует относить вспомогательный материал, который при заключении в основную часть работы загромождает текст. К вспомогательному материалу относятся образцы первичных документов, структуры баз данных, формы выходных документов, листинги

программ, иллюстрации вспомогательного характера. Если приложений больше десяти, их следует объединить по видам. Приложения обозначаются заглавными буквами русского алфавита.

3 ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

3.1 ОБЩИЕ ПРАВИЛА ОФОРМЛЕНИЯ

1. Курсовую работу рекомендуется оформлять с применением компьютера на листах белой бумаги формата А4. Для рисунков и таблиц допускается использование формата А3. Текст дипломной работы оформляется на одной стороне формата А4.
2. При работе в WordMSOffice рекомендуемый шрифт TimesNewRoman, размер шрифта № 14. Интервал полуторный.
3. Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах, применяя шрифты разной гарнитуры.
4. Текст ПЗ в рамках следует размещать, соблюдая следующие размеры согласно ГОСТ 2.104:
 - - расстояние от рамки и в конце строк не менее 3 мм;
 - - расстояние от текста до верхней и нижней рамки должно быть не менее 10 мм;
 - - абзацы в тексте начинают с отступом, равным 12 – 12,5 мм.
5. При выполнении ПЗ без рамки необходимо соблюдать следующие размеры полей: правое – не менее 10 мм, верхнее и нижнее – не менее 20 мм, левое – не менее 30 мм
6. Абзацный отступ должен быть одинаков по всему тексту и равен 12,5 мм.
7. Условные обозначения и сокращения, используемые в пояснительной записке, следует расшифровать.
8. Опечатки, описки и графические неточности допускается исправлять подчисткой или закрашиванием корректором для бумаги и нанесением на том же месте исправленного текста. Плотность вписанного текста должна быть приближенной к плотности основного текста. Если текст был напечатан на принтере, то исправления разрешено вносить только черной пастой.
9. Нумерация страниц в тексте ПЗ, включая приложения, иллюстрации и таблицы, выполненные на листах формата А4, *сквозная*. Первым листом является титульный лист ПЗ. Вторым (возможно и третьим) – задание на проектирование. Номера страниц на них не проставляются, но входят в общую нумерацию листов.
10. Полное наименование темы курсового проекта (работы) или ВКР на титульном листе, на листе задания, в основной надписи и в тексте ПЗ должно быть одинаковым. Неточности в формулировке, сокращения не допускаются.
11. Текст ПЗ должен быть кратким, четким, не допускающим различных толкований и при необходимости разделяться на разделы, подразделы, пункты и подпункты. Степень дробления материала зависит от его объема и содержания. В выпускных квалификационных работах каждый *раздел следует начинать с нового листа*, в курсовых проектах (работах) допускается последующий раздел оформлять не с нового листа, а в продолжение текста предыдущего раздела. Подразделы с нового листа не начинаются. Не допускается размещать наименования подразделов в нижней части листа, если под ними помещается менее двух строк текста
12. Разделы должны иметь порядковые номера в пределах всей ПЗ, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. Подразделы должны иметь порядковые номера в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой. Точки в конце номера подраздела не ставят. Если в подразделе имеются пункты, то нумерация пунктов должна быть в пределах подраздела. Номер пункта состоит из номеров раздела, подраздела, пункта, разделённых точками. В конце номера пункта точка не ставится.

13. Разделы и подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны чётко и кратко отражать содержание. Заголовки следует **оформлять с абзацного отступа с прописной буквы без точки в конце, не подчеркивая**. Если заголовок состоит из двух предложений, их разделяют точкой. Перенос слов в заголовках не допускается. **Точки в конце заголовка не ставятся**. Для заголовков разделов, подразделов, пунктов используется шрифт Times New Roman, размер 14 пт. Иная гарнитура шрифта не допускается (за исключением случаев оформления курсовых или дипломных работ (проектов) в рукописном виде. **Заголовки разделов допускается оформлять полужирным шрифтом**.
14. Наименования разделов основной части ПЗ, а также раздел «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ» следует располагать на отдельной строке, посередине, симметрично основному тексту с соответствующей нумерацией. Наименование разделов «СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» следует располагать на отдельной строке, посередине, симметрично основному тексту и не нумеровать.
15. Расстояние между заголовком раздела и заголовком подраздела, а также между заголовком раздела и текстом при использовании текстового редактора пропускается одна строка, интервал полуторный
16. При изложении обязательных требований в тексте должны применяться слова «должен», «следует», «необходимо», «требуется, чтобы», «разрешается только», «не допускается», «запрещается», «не следует». При изложении других положений следует применять слова «могут быть», «может быть», «как правило», «при необходимости», «в случае» и т. д. Допускается использовать повествовательную форму изложения текста ПЗ, например, «применяют», «указывают» и т. п.

3.2 КОМПЛЕКТАЦИЯ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Общий объем должен быть в пределах 25-30 страниц печатного текста (без приложений). Работа сшивается в специальной папке или переплетается. Порядок комплектации курсовой работы : 1-й лист – титульный; 2-й лист – Задание; 3-й лист - Содержание.

Начиная с 4-го листа - текст пояснительной записки (ПЗ), затем – Приложения.

Титульный лист является первым листом пояснительной записки (далее ПЗ). Оформлять его следует на белой бумаге формата А4 (210×297 мм) по ГОСТ 2.301. Для написания наименования вуза, слова «ПОЯСНИТЕЛЬНАЯ ЗАПИСКА», наименования и обозначение (шифра) проекта (работы) применяется шрифт в текстовом редакторе – Times New Roman, 14 пт, буквы прописные. Наименование вуза, слово «ПОЯСНИТЕЛЬНАЯ ЗАПИСКА» оформляются полужирным шрифтом. Для всех остальных надписей используется шрифт в текстовом редакторе – 12 пт, Times New Roman. Перенос слов на титульном листе не допускается. Точки в конце строк не ставятся.

Слово «СОДЕРЖАНИЕ» оформляется в текстовом редакторе – полужирный шрифт, 14 пт Times New Roman, буквы прописные в виде заголовка симметрично основному тексту.

Наименования разделов и подразделов, включённые в содержание, оформляются шрифтом в текстовом редакторе – 14 пт, Times New Roman строчными буквами, начиная с прописной за исключением разделов: «ВВЕДЕНИЕ», «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ», «ЗАКЛЮЧЕНИЕ», «ПРИЛОЖЕНИЯ».

Для удобства оформления раздела в текстовом редакторе можно использовать скрытую таблицу или таблицу с границами белого цвета. При использовании таблицы выравнивание в столбцах производится по левому краю. В перечне наименований разделов (пунктов) расстояние от конца строки, содержащей наименование раздела (пункта) до номера страницы, на которой начинается данный раздел (пункт), должно составлять не менее 1 см. Если наименование раздела (пункта) не умещается на одну строку, его переносят на следующие

строки, при этом перенос слов запрещён. Номер страницы проставляется напротив последней строки.

3.3 НУМЕРАЦИЯ

Страницы работы следует нумеровать арабскими цифрами соблюдая сквозную нумерацию по всему тексту, включая приложения. Титульный лист включается в общую нумерацию страниц, но номер страницы на титульном листе не проставляется. Задание является второй страницей ПЗ (возможно задание на двух листах, тогда это 2 и 3 страницы ПЗ). Номера страниц не проставляются Страница с «Содержанием» имеет номер 3 или 4 (если задание на двух листах).

Разделы, подразделы, пункты, подпункты нумеруются арабскими цифрами, например:

1. Программное обеспечение
- 2.1 Общие принципы разработки

В тексте ПЗ каждый раздел (т.е номер которого представлен цифрой без точек), а также «Введение», «Заключение», «Список литературы», «Содержание» начинаются с нового листа (это не относится к подразделам, пунктам, подпунктам и т.д.). Подразделы (имеющие сложные номера, например, 2.1. или 4.3.) пишутся в продолжении страницы; название подраздела выделяется пустыми строками. После номера точка не ставится. Внутри пунктов или подпунктов могут быть приведены перечисления. Перед каждой позицией перечисления следует поставить дефис. Более сложные перечисления могут обозначаться буквами и цифрами, например:

- а).....
- б).....
 - 1).....
 - 2).....
- в).....

В современных стандартах все чаще используется прием оформления перечислений в виде абзацев; именно этот способ рекомендуется использовать при оформлении текста пояснительной записки. Заголовки следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая, отделяя от текста одним межстрочным интервалом. Никаких фломастеров! Переносы слов в заголовках не допускаются.

3.4 РИСУНКИ, СХЕМЫ

Рисунки, иллюстрации, схемы и т.п. следует располагать сразу после текста, в котором они упоминаются впервые, или на следующей странице, если в указанном месте они не помещаются. На все иллюстрации должны быть ссылки в тексте. Иллюстрации должны иметь названия, которые помещают под иллюстрацией, их следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Допускается нумерация внутри раздела, в этом случае номер рисунка сложный. Иллюстрации обозначаются словом Рисунок.



Рисунок 1 - Документ

Если рисунок имеет наименование, оно пишется после номера рисунка через дефис и с заглавной буквы. При ссылке на рисунок пишется: «в соответствии с рисунком 1..».

3.5 ТАБЛИЦЫ

Таблицу следует располагать в работе непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На все таблицы должны быть ссылки в

тексте. Используется сквозная нумерация таблиц по всей работе. Номер таблицы размещается в левом верхнем углу таблицы после слова «Таблица». Каждая таблица должна иметь заголовок (наименование), который пишется через дефис после номера таблицы. Слово «Таблица» и заголовок начинаются с прописной буквы, точка в конце заголовка не ставится.

Таблица 1 - Наименование таблицы в миллиметрах

Заголовки граф таблицы должны начинаться с прописных букв, подзаголовки – со строчных, если последние подчиняются заголовку. Заголовки граф указываются в единственном числе.

Таблицу следует размещать так, чтобы читать ее без поворота работы. Если такое размещение невозможно, таблицу располагают так, чтобы ее можно было читать, поворачивая работу по часовой стрелке. При переносе таблицы шапку таблицы следует повторить, и над ней размещают слова «Продолжение таблицы» с указанием ее номера. Если шапка велика, можно указать только номера граф. Заголовок (т.е название) таблицы не повторяют.

Если цифровые или иные данные в какой-либо строке отсутствуют, то ставится прочерк. Если все данные выражены в одной и той же единице, то ее обозначение помещается над таблицей справа. Заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента, обозначения марок материала, нормативных материалов и т.п. не допускается. При наличии в тексте небольшого по объему материала его нецелесообразно оформлять в виде таблицы, а следует давать в виде текста, располагая цифровые данные в колонки.

3.6 ФОРМУЛЫ И УРАВНЕНИЯ

Уравнения и формулы следует выделять из текста в отдельную строку, выше и ниже которой должно быть оставлено не менее одной свободной строки.

Если уравнение не помещается на одной строке, оно должно быть перенесено с обязательным повторением математического знака (= ; +; или другие знаки) на другой строке. Формулы и уравнения нумеруются сквозной нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении напротив формулы (если только одна формула, ее можно не нумеровать). Допускается нумерация в пределах раздела, тогда используется сложный номер. Пояснения значений символов и коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Значение каждого символа следует давать с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия. После каждого пояснения ставить «,».

3.7 ССЫЛКИ

Ссылки на литературу - проставляется в квадратных скобках номер, под которым данный источник значится в библиографическом списке. Допускается использование вместо квадратных скобок косых линий /5/. В необходимых случаях (при использовании цитат, цифровых данных) указываются и страницы.

Ссылки на таблицы, рисунки, формулы, приложения берутся в круглые скобки.

3.8 ПРИЛОЖЕНИЯ

Каждое приложение должно начинаться с новой страницы и иметь заголовок с указанием сверху посередине страницы слова «Приложение» и его обозначение. Если приложений больше одного, то они обозначаются буквами русского алфавита, начиная с А, за исключением Е, З, О, Ч, Ъ, Ы, Ь. Допускается обозначение буквами латинского алфавита, за исключением I, O. В случае полного использования букв алфавитов допускается обозначение

арабскими цифрами. Если в документе одно приложение, оно обозначается «Приложение А». Текст каждого приложения может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруются в пределах каждого приложения. Располагаются приложения в порядке появления на них ссылок в тексте. Приложения помещаются в конце курсовой работы после списка литературы. Рисунки в приложении должны иметь свои номера. Эти номера могут быть в пределах каждого приложения, тогда они обозначаются следующим образом: Рисунок А.1 (для Приложения А).

3.9 ПОДПИСИ

Курсовая работа может быть представлена к защите только при наличии следующих подписей:

Титульный лист	- студент; - руководитель курсовой работы - директор колледжа
Лист задания	- председатель цикловой комиссии - руководитель курсовой работы

4 ЗАЩИТА

За неделю до официальной защиты курсовая работа сдается на проверку. По результатам проверки студент либо допускается к защите, либо нет. Защита проводится до экзамена. Студенты, не защитившие курсовую работу, к экзамену не допускаются.

Защита курсовых проектов осуществляется перед комиссией с предоставлением пояснительной записки и машинного носителя с созданным программным продуктом. Защита осуществляется с обязательной демонстрацией программного продукта на компьютере. Все замечания, полученные при проверке курсовых работ, должны быть учтены при защите. В случае «не защиты», тема курсовой работы студенту должна быть изменена. Защита переносится на следующий год или семестр.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)
Авиационно-технологический колледж**

(наименование образовательного учреждения)

Директор колледжа

(подпись)

В.А. Зибров
И.О.Ф.

«__»

_____ 2023г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по междисциплинарному курсу
МДК 02.01 «Технология разработки программного обеспечения»
(наименование учебной дисциплины (междисциплинарного курса))

профессионального модуля ПМ 02 «Осуществление интеграции программных модулей»

на тему

Автор работы

Специальность:

09.02.07 Информационные системы и программирование

код специальности

наименование специальности

Обозначение курсовой работы _____

Группа _____

Руководитель работы

Работа защищена

г. Ростов-на-Дону

202_ г.

Содержание

ВВЕДЕНИЕ	5
1. Анализ и формирование требований к ПО	6
2. Проектирование программного обеспечения	9
2.1 Архитектура и структура проектируемого ПО	9
2.2 Информационная модель	11
3. Разработка программного обеспечения	14
3.1 Инструментальные средства разработки ПО	14
3.2 Описание ПО	15
3.3 Руководство пользователя	18
3.4 Тестирование ПО	20
4. Информационная безопасность	22
5. Расчет метрик	24
ЗАКЛЮЧЕНИЕ	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27
ПРИЛОЖЕНИЯ	28

						ТРПО.NN190100.000 ПЗ			16	
<i>Изм</i>	<i>Лист</i>	<i>№ докум</i>	<i>Подп</i>	<i>Дата</i>	Разработка базы данных для ООО «МЕНС»			<i>Листы</i>	<i>Лист</i>	<i>Листов</i>
<i>Разраб</i>										
<i>Проверил</i>										

ПРИЛОЖЕНИЕ А

Стандарты оформления программного текста, описания программы, описания использования программы

Группа Т55

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

Единая система программной документации

ТЕКСТ ПРОГРАММЫ

Требования к содержанию и оформлению ГОСТ

Unified system for program documentation.

19.401-7-78

Text of program.

Requirements for contents and form of presentation

МКС 35.080

Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. №3350 дата введения установлена
01.01.80

1. Настоящий стандарт устанавливает требования к содержанию и оформлению программного документа «Текст программы», определенного ГОСТ 19.101-77.

Стандарт полностью соответствует СТ СЭВ 3746-82.

2. Структуру и оформление документа устанавливают в соответствии с ГОСТ 19.105-78.

Составление информационной части (аннотации и содержания) является необязательным. Для текста программы на исходном языке при наличии аннотации в нее включают краткое описание функций программы.

1,2. (Измененная редакция, Изм. № 1).

3. Основная часть документа должна состоять из текстов одного или нескольких разделов, которым даны наименования.

Допускается вводить наименование также и для совокупности разделов.

4. Каждый из этих разделов реализуется одним из типов символической записи, например:

символическая запись на исходном языке;

символическая запись на промежуточных языках;

символическое представление машинных кодов и т.п.

В символическую запись разделов рекомендуется включать комментарии, которые могут отражать, например функциональное назначение, структуру.

Издание официальное

Перепечатка воспрещена

Издание с Изменением № 1, утвержденным в марте 1983 г. (ИУС 7-83).

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

Единая система программной документации

Описание программы

ГОСТ
19.402-78

Unifiedsystemforprogramdocumentation
Programdescription

Мкс 35.080

Постановление Государственного комитета СССР по стандартам от 18 декабря 1978г.№3350 дата введения установлена

1. Настоящий стандарт устанавливает состав и требования к содержанию программного документа “Описание программы”, определенного Гост 19.101---77.

Стандарт полностью соответствует СТ СЭВ 2092-80
Изменившая редакция, Изм № 1).

2. Структуру и оформление документа устанавливают в соответствии с Гост 19.105-78
Составление информационной части (Аннотации и содержания) является обязательным

3. Описание программы должно содержать следующие разделы.

Общие сведения;

Функциональные назначения;

Описание логической структуры;

Используемые технические средства;

Вызов и загрузка;

Входные данные

выходные данные

В зависимости от особенностей программы допускается вводить дополнительные разделы или объединить отделены разделы

4. В разделе “Общие сведения” должны быть указаны:

Обозначение и наименование программы;

Программное обеспечение ,необходимое для функционирования программы;
языки программирования, на которых написано программа

5. В разделе “Функциональное назначение” должны быть указаны классы решаемых задач и (или) назначение программы и сведения о функциональных ограничениях на применение

6. В разделе “Описание логической структуры” должны быть указаны:

алгоритмы программы;

Используемые методы;

Структура программы с описанием функций составных частей и связи между ними;

связи программы с другими программами.

Описание логической структуры программы выполняют с учетом текста программы на исходном языке

3-6 (**Измененная редакция, Изм. №1**).

7. В разделе “Используемые технические средства” должны быть указаны типы электронных вычислительных машин и устройств, которые используются при работе программы

Издание официальное

Перепечатка воспрещена

Издание с Изменением №1, утвержденным в сентябре 1981г (ИУС 11-81)

С.2 ГОСТ 19.402-78

8. В разделе “Вызов и загрузка” должны быть указаны:

способ вызова программы с соответствующего носителя данных;

входные точки в программы

Допускается указывать адреса загрузки, сведения об использовании оперативной памяти, объем программы

9. В разделе “Входные данные” должны быть указаны:

характер, организация и предварительная подготовка входных данных.

10. В разделе “Выходные данные” должны быть указаны;

характер и организация выходных данных;
формат, описание и способ кодирования выходных данных.

11. Допускается содержание разделов иллюстрировать пояснительными примерами, таблицами, схемами, графиками

12. В приложение к описанию программы допускается включать различные материалы, которые нецелесообразно включать в разделы описания

7-12 (Введены дополнительно, Изм. № 1).

М Е Ж Г О С У Д А Р С Т В Е Н Н Ы Й С Т А Н Д А Р Т

Единая система программной документации

ОПИСАНИЕ ПРИМЕНЕНИЯ

ГОСТ

Требования к содержанию и оформлению

10 503 — 78

МКС 35.

Unified system for program documentation.

Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3350 дата введения установлена

01.01.80

1. Настоящий стандарт устанавливает состав и требования к содержанию программного документа “Описание применения”, определенного ГОСТ 19.101—77.

Стандарт полностью соответствует СТ СЭВ 2093 — 80.

(Измененная редакция, Изм. № 1).

2. Структура и оформление документа устанавливаются в соответствии с ГОСТ 19.05—78.

Составление информационной части (аннотации и содержания) является обязательным.

3. Текст документа должен состоять из следующих разделов:

назначение программы;

условия применения;

описание задачи;

входные и выходные данные;

4. В зависимости от особенностей программы допускается вводить дополнительные разделы или объединять отдельные разделы.

5. В разделе “Назначение программы” указывают назначение, возможности программы, ее основные характеристики, ограничения, накладываемые на область применения программы.

3—5. **(Измененная редакция, Изм. № 1).**

6. В разделе “Условия применения” указывают условия, необходимые для выполнения программы (требования к необходимым для данной программы техническим средствам, и другим программам, общие характеристики входной и выходной информации, а также требования и условия организационного, технического и технологического и т.п.).

7. В разделе “Описание задачи” должны быть указаны определения задачи и методы ее решения.

(Измененная редакция, Изм. № 1).

7а. В разделе “Входные и выходные задачи” должны быть указаны сведения о входных и выходных данных.

(Введен дополнительно, Изм. № 1).

8. В приложение к общему описанию могут быть включены справочные материалы (иллюстрации, таблицы, графики, примеры и т.п.).

Издание официальное

Перепечатка воспрещена

Издание с Изменением № 1, Утверждение в сентябре 1981 г. (ИУС 11—81).

Приложение Б Пример Тест- Кейса

Название:	Тест отправки сообщения	
Функция:	Контакт-Вопросы	
Действие	Ожидаемый результат	Результат теста: <ul style="list-style-type: none"> • пройден • провален • заблокирован
<i>Предусловие:</i>		
Откройте сайт Про Тестинг: http://www.protesting.ru	Сайт Про тестинг открыт и доступен	
Перейдите по ссылке " Задать вопрос " (внизу страницы)	Страница "Вопросы, пожелания и заявки" открыта и доступна	
<i>Шаги теста:</i>		
Заполните форму отправки комментария: "Тип Обращения": Консультация "Контактное лицо": Ольга "E-mail": test@test.com "Сообщение": <i>Добрый день, уважаемый коллектив "ПроТестинг"! Я еще ни разу не видела кач. примера баг-репорта, тест-кейса и прочей необходимой док-ии. Не подскажите, где я могу с ними ознакомиться? С уважением.</i>	Данные успешно введены	
Нажмите кнопку "Отправить"	Страница "Ваш запрос успешно отправлен!" открыта	
<i>Постусловие:</i>		
Кликните по ссылке " Перейти Назад на форму отправления заявок "	Страниц "Вопросы, пожелания и заявки" открыта	