

Московский государственный технический университет  
имени Н.Э. Баумана

---

Факультет «Информатика и системы управления»

Кафедра «Компьютерные системы и сети»

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

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

Электронное учебное издание

Москва

© 2019 МГТУ им. Н.Э. Баумана

УДК 004.42

*Рецензент:* доц., к.т.н., Наталия Владимировна Новик

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

Методические указания по выполнению курсовой работы по дисциплине «Технология разработки программных систем». Электронное учебное издание. - М.: МГТУ имени Н.Э. Баумана, 2019, 42 с.

Настоящее электронное учебное издание содержит методические указания и рекомендации по выполнению и оформлению курсовой работы по дисциплине «Технология разработки программных систем». В издании описываются требования к темам курсовых работ, этапы ее выполнения, порядок ее защиты, а также требования к документам, выносимым на защиту, и правила их оформления. Особое внимание обращено на содержание и правила составления текстовых документов: технического задания и расчетно-пояснительной записки. В приложении приводятся примеры оформления реферата на расчетно-пояснительную записку.

Пособие предназначено для студентов, обучающихся на третьем курсе бакалавриата кафедры «Компьютерные системы и сети».

*Рекомендовано учебно-методической комиссией факультета  
«Информатика и системы управления» МГТУ им. Н.Э. Баумана*

*Электронное учебное издание*

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

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

© 2019 МГТУ имени Н.Э. Баумана

## Оглавление

<b><u>ВВЕДЕНИЕ</u></b> .....	<b>5</b>
<b><u>1 ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ</u></b> .....	<b>6</b>
<u>1.1 Понятие проекта и управления проектами</u> .....	6
<u>1.2 Жизненный цикл разработки ПО</u> .....	6
<b><u>2 ОРГАНИЗАЦИЯ ПРОВЕДЕНИЯ КУРСОВОЙ РАБОТЫ</u></b> .....	<b>8</b>
<u>2.1 Темы курсовых работ</u> .....	8
<u>2.2 Порядок выполнения курсовой работы</u> .....	9
<u>2.3 Сроки выполнения отдельных этапов</u> .....	10
<u>2.4 Порядок защиты курсовой работы</u> .....	11
<u>2.5 Примеры вопросов, задаваемых при защите курсовой работы</u> .....	12
<b><u>3 ПРИНЦИПЫ ФОРМИРОВАНИЯ ТЕХНИЧЕСКОГО ЗАДАНИЯ</u></b> .....	<b>13</b>
<u>3.1 Назначение технического задания</u> .....	13
<u>3.2 Содержание разделов технического задания</u> .....	13
<b><u>4 РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА</u></b> .....	<b>16</b>
<u>4.1 Содержание расчетно-пояснительной записки</u> .....	16
<u>4.2 Реферат</u> .....	18
<u>4.3 Введение</u> .....	18
<u>4.4 Анализ требований и уточнение спецификаций</u> .....	18
<u>4.5 Проектирование структуры и компонентов программного продукта</u> .....	21
<u>4.6 Выбор стратегии тестирования и разработка тестов</u> .....	27
<u>4.7 Заключение</u> .....	28
<b><u>5 ПРАВИЛА ОФОРМЛЕНИЯ ДОКУМЕНТАЦИИ</u></b> .....	<b>29</b>
<u>5.1 Стандарты оформления документов</u> .....	29
<u>5.2 Оформление текстового и графического материала</u> .....	30
<u>5.3 Оформление рисунков, схем алгоритмов, таблиц и формул</u> .....	31
<u>5.4 Оформление текстов программ</u> .....	33
<u>5.5 Оформление приложений</u> .....	34
<u>5.6 Оформление списка литературы</u> .....	35
<b><u>6 ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ</u></b> .....	<b>37</b>
<u>6.1 Современные инструменты автоматизации разработки ПО</u> .....	37
<u>6.2 Регистрация и работа с репозиторием кода</u> .....	38
<u>6.3 Полезные ссылки</u> .....	38

<b><u>7 ЛИТЕРАТУРА</u></b> .....	<b>39</b>
<u>7.1 Основная литература по дисциплине</u> .....	39
<u>7.2 Дополнительные учебные материалы</u> .....	39
<b><u>ПРИЛОЖЕНИЕ А ПРИМЕР РЕФЕРАТА НА РАСЧЕТНО-ПОЯСНИТЕЛЬНУЮ ЗАПИСКУ</u></b> .....	<b>40</b>
<b><u>ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СОКРАЩЕНИЙ И ТЕРМИНОВ</u></b> .....	<b>41</b>

## Введение

Курсовая работа по дисциплине «Технология разработки программных систем» посвящена выработке навыков разработки программного продукта среднего уровня сложности. Для этого студенту нужно создать программу размером не менее 1000 операторов универсального языка программирования, с развитым пользовательским интерфейсом, с применением современных инструментов разработки, которые используются для реализации средних и крупных программных проектов.

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

- понимание и умение уточнять требования к программным системам;
- разработка программных продуктов (разработка структуры программного обеспечения, структур данных, алгоритмов и реализующих их программ, разработка тестов и т.п.);
- тестирование и отладка интерактивных систем программного обеспечения;
- составление пояснительной записки, содержащей обоснование принятых проектных решений;
- применение нормативных документов, регламентирующих состав, содержание и форму технической документации на разработанный программный продукт;
- применение современных инструментов контроля версий, документирования кода и других инструментов поддержки разработки крупных проектов.

Методические указания основаны на следующих учебниках:

- Иванова Г.С. Технология программирования [1],
- Иванова Г.С. Ничушкина Т.Н., Объектно-ориентированное программирование [2],
- Иванова Г.С. Программирование [3].

Также учитывались требования ГОСТ 7.32-2017 [4] и других регламентирующих документов.

# 1 Основы разработки программного обеспечения

## 1.1 Понятие проекта и управления проектами

*Проект* – совокупность мероприятий для разработки нового продукта или улучшения существующего продукта<sup>1</sup>.

Любой проект по разработке программного обеспечения (ПО), даже небольшой, испытывает влияние следующих ключевых объективных факторов, которые называют «ограничениями проекта»:

- ограничение срока выполнения проекта;
- ограничение бюджета проекта;
- ограничение на содержание продукта проекта;
- ограничение на качество продукта проекта.

Можно заметить, что изменение одного из этих ограничений, как правило, противоречит другим. Так, уменьшение срока выполнения проекта, скорее всего повлечёт сокращение содержания продукта проекта и/или ухудшение качества продукта. При расширении содержания продукта проекта нужно учитывать увеличение сроков и бюджета проекта. И так далее.

*Управление проектами* является практической деятельностью с применением разнообразных инструментов, технологий и методик, которые дают возможность организовать работу с учётом указанных ограничений, сбалансировать их и контролировать изменение соотношений. Чем больше проект, тем актуальнее применение инструментов, технологий и методик управления проектом для контроля над этими ограничениями.

Предлагаемый для использования в курсовой работе подход являются одними из наиболее успешных и проработанных на практике способов управления ограничениями проекта.

## 1.2 Жизненный цикл разработки ПО

Стандарт ГОСТ Р ИСО/МЭК 12207-2010. «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» даёт следующее определение жизненного цикла:

---

<sup>1</sup> ISO/IEC 26514 Systems and software engineering — Requirements for designers and developers of user documentation

*Жизненный цикл* (life cycle): Развитие системы, продукта, услуги, проекта или других изготовленных человеком объектов, начиная со стадии разработки концепции и заканчивая прекращением применения.

Указанный стандарт предлагает построение этапов жизненного цикла на основе процессов. Приводятся следующие группы процессов:

- основные процессы (приобретение, поставка, разработка, эксплуатация, сопровождение);
- вспомогательные процессы; обеспечивают выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит);
- организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение жизненного цикла ПО, обучение).

Указанные процессы можно условно сгруппировать, выделив следующие основные этапы (в скобках указаны соответствующие стадии разработки по ГОСТ 19.102–77 «Стадии разработки»):

- постановка задачи (стадия «Техническое задание»);
- анализ требований и разработка спецификаций (стадия «Эскизный проект»);
- проектирование (стадия «Технический проект»);
- реализация (стадия «Технический проект»);
- сопровождение.

В курсовой работе необходимо выделять эти этапы.

## 2 Организация проведения курсовой работы

### 2.1 Темы курсовых работ

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

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

Примерные темы курсовых работ:

- «Графический редактор для изображения схем алгоритмов»;
- «Система представления табличной информации в графическом виде»;
- «Специализированный текстовый редактор»;
- «Система расчета простейших электрических схем»;
- «Информационно-поисковая система «Астроном»;
- «Система учета успеваемости студентов»;
- «Автоматизированная система складского учета»;
- «Консультирующая медицинская экспертная система»;
- Web – приложение:
  - «Корпоративный сайт»,
  - «Информационный портал»,
  - «Интернет-магазин»,
  - «Система управления содержимым сайтов»,
  - «Система управления документооборотом»,
  - «Система управления продажами»,
  - «Система бронирования билетов» и др.



## 2.2 Порядок выполнения курсовой работы

Порядок выполнения курсовой работы зависит от класса разрабатываемого программного продукта.

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

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

При разработке веб-приложений студенты должны:

- разработать развернутое техническое задание на веб-приложение;
- провести анализ предметной области и существующих способов реализации задачи;
- выполнить проектирование и разработку дизайна интерфейса пользователя (проектирование логической структуры страниц сайта и логики перехода между ними; выбор способов управления сайтом – меню, всплывающие закладки и пр.; дизайн – подбор расположения элементов, цветовых схем, рисунков и др.);
- выполнить проектирование логики обработки данных;
- выбрать средства для реализации сервера;
- выполнить проектирование серверных компонентов и выбрать методы их тестирования (например, с использованием фреймворка RubyOnRails: выделение основных сущностей – объектный принцип проектирования; проектирование контроллеров, представлений, маршрутов; оценка возможности использования стандартных генераторов Rails; проектирование моделей, ограничений и связей; наполнение тестов контроллеров др.);
- выполнить проектирование БД;

- выбрать средства для реализации интерфейса пользователя (статический, динамический, с использованием библиотек jQuery, Dojo и пр.);
- провести проектирование компонентов интерфейса пользователя;
- реализовать серверные и клиентские компоненты;
- разработать технологию и выполнить тестирование веб-приложения (тестирование компонентов; тестирование моделей данных; написание функциональных тестов, тестирование со стороны интерфейса пользователя при помощи Selenium Webdriver и др.);
- разработать необходимую документацию, указанную в техническом задании.

При разработке экспертной системы студенты должны:

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

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

## 2.3 Сроки выполнения отдельных этапов

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

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

Таблица 1 – Этапы выполнения курсовой работы

Этап	Содержание этапа	Сроки и объем	Представляемые результаты
1.	Утверждение задания на выполнение работы	2 неделя	Утвержденное руководителем курсовой работы задание (лист задания в двух экземплярах)
2.	Утверждение технического задания	4 неделя (25%)	Утвержденное техническое задание, готовность части работы (интерфейс, схемы, часть программы)
3.	Реализация основных функций программы	7 неделя (50 %)	Готовность основных функций программы демонстрируются руководителю
4.	Завершение разработки программы	11 неделя (75 %)	Готовая программа, которую руководитель оценивает в баллах
5.	Подготовка к защите и защита курсовой работы	14 неделя (100 %)	Подписанные РПЗ и другие документы. Доклад (3-5 минут)

## 2.4 Порядок защиты курсовой работы

На защиту студент предоставляет:

1. Техническое задание (в качестве приложения А к РПЗ).
2. Программный продукт (подписанный CD/DVD-диск с проектными документами, исходными кодами и программным продуктом, либо те же проектные артефакты в вузовском репозитории кода – <https://bmstu.codes> [5] – и ссылку на проект в нем в расчетно-пояснительной записке).
3. Расчетно-пояснительную записку на 25-30 страницах в распечатанном виде (шрифт 12 через два интервала), содержащую описание разработки и соответствующие иллюстрации.
4. Программную документацию, указанную в разделе «Требования к программной документации» технического задания.

Программный продукт студент предварительно демонстрирует и сдает руководителю курсовой работы, который оценивает работу (максимум – 50 баллов, зачетный минимум – 30 баллов) и допускает студента к защите.

В процессе демонстрации программного продукта проверяется:

- соответствие программы техническому заданию;
- работоспособность в различных режимах;
- стиль оформления кода программы;
- использование современных инструментов разработки ПО.

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

Распечатанные и подписанные студентом и руководителем курсовой работы документы необходимо поместить в папку или файл.

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

Оценка за курсовую работу выставляется с учетом:

- качества выполненного программного продукта – 30–50 баллов;
- правильности оформления расчетно-пояснительной записки – 18–30 баллов;
- результатов защиты – доклад и ответы на вопросы – 12–20 баллов.

## **2.5 Примеры вопросов, задаваемых при защите курсовой работы**

1) Уточните способ декомпозиции, который вы использовали при проектировании программного продукта. Почему вы выбрали именно этот способ декомпозиции?

2) Какие технологии вы использовали при разработке вашего программного продукта?

3) К какому типу относится интерфейс вашего программного продукта? К какому виду?

4) Какие еще варианты интерфейса вы рассматривали?

5) Что показывает схема структурная программного обеспечения?

6) Какие средства вы выбрали для реализации вашего программного продукта и почему?

7) В какой последовательности велась разработка вашего продукта? Реализация продукта?

8) Как тестировался ваш программный продукт?

9) Какие методы оценочного тестирования вы использовали и почему?

## 3 Принципы формирования технического задания

### 3.1 Назначение технического задания

Процесс создания нового ПО начинают с постановки задачи, в процессе которой определяют требования к программному продукту. Это один из наиболее ответственных этапов в процессе создания ПО. От того насколько полно определены функции будущего программного продукта и другие требования к нему, во многом зависит стоимость разработки и ее качество.

Техническое задание представляет собой документ, в котором формулируют основные цели разработки, требования к программному продукту, определяют сроки и этапы разработки и регламентируют процесс приемно-сдаточных испытаний. В формулировании технического задания участвуют как представители заказчика, так и представители исполнителя. В основе этого документа лежат исходные требования заказчика, анализ передовых достижений техники, результаты выполнения научно-исследовательских работ, предпроектных исследований, научного прогнозирования и т.п.

На техническое задание существует стандарт (ГОСТ 19.201–78). В соответствии с этим стандартом техническое задание должно содержать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе или программному изделию;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

При необходимости допускается в техническое задание включать приложения.

### 3.2 Содержание разделов технического задания

*Введение* должно включать наименование и краткую характеристику области применения программы или программного продукта. Основное назначение введения – продемонстрировать актуальность данной разработки и показать, какое место эта разработка занимает в ряду подобных.

Раздел *Основания для разработки* должен содержать наименование документа, на основании которого ведется разработка, организации, утвердившей данный документ, и наименование или условное обозначение темы разработки. Таким документом может служить план, приказ, договор и т.п.

Раздел *Назначение разработки* должен содержать описание функционального и эксплуатационного назначения программного продукта с указанием категорий пользователей.

Раздел Требования к программе или программному изделию должен включать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

Наиболее важным из перечисленных выше является подраздел *Требований к функциональным характеристикам*. В этом разделе должны быть перечислены выполняемые функции и описаны требования к составу, характеристикам и формам представления исходных данных и результатов. В этом же разделе при необходимости указывают критерии эффективности: максимально допустимое время ответа системы, максимальный объем используемой оперативной и/или внешней памяти и др.

**Примечание** – В подразделе *Требования к функциональным характеристикам* особенно важно приводить проверяемые формулировки, достаточно четкие и подробные, чтобы не возникало многозначных толкований. Это важно с точки зрения управления *проектными ограничениями* (см. подраздел 1.1), а именно:

- должна быть возможность каждое функциональное требование продемонстрировать в готовом программном продукте и доказать его выполнение при сдаче проекта (курсовой работы);
- у заказчика (приемной комиссии) не должно возникнуть иного представления о реализации функциональности, кроме того, что описано в требованиях ТЗ и реализовано в готовом программном продукте.

В подразделе *Требования к надежности* указывают уровень надежности, который должен быть обеспечен разрабатываемой системой и время восстановления системы после сбоя. Для систем с обычными требованиями к надежности в этом разделе иногда регламентируют действия разрабатываемого продукта по увеличению надежности результатов (контроль входной и выходной информации, создание резервных копий промежуточных результатов и т. п.).

В подразделе *Условия эксплуатации*, указывают особые требования к условиям эксплуатации: температуре окружающей среды, относительной влажности воздуха и т.п. Как правило, подобные требования формулируют, если разрабатываемая система будет эксплуатироваться в нестандартных условиях или использует специальные внешние устройства, например, для хранения информации. Здесь же указывают вид обслуживания, необходимое количество и квалификация персонала.

В подразделе *Требования к составу и параметрам технических средств* указывают необходимый состав технических средств с указанием их основных технических характеристик: тип микропроцессора, объем памяти, наличие внешних устройств и т.п. При этом часто указывают два варианта конфигурации: минимальный и рекомендуемый.

В подразделе *Требования к информационной и программной совместимости* при необходимости можно задать методы решения, определить язык или среду программирования для разработки, а также используемую операционную систему и другие системные и пользовательские программные средства, с которыми должно взаимодействовать разрабатываемое ПО. В этом же разделе при необходимости указывают, какую степень защиты информации необходимо предусмотреть.

В разделе *Требования к программной документации* указывают необходимость наличия руководства программиста, руководства пользователя, руководства системного программиста, пояснительной записки и т.п. На все эти типы документов также существуют ГОСТы.

В разделе *Технико-экономические показатели* рекомендуется указывать ориентировочную экономическую эффективность и экономические преимущества по сравнению с существующими аналогами.

В разделе *Стадии и этапы разработки* указывают стадии разработки, этапы и содержание работ с указанием сроков разработки и исполнителей.

В разделе *Порядок контроля и приемки* указывают виды испытаний и общие требования к приемке работы.

В приложениях при необходимости приводят: перечень научно-исследовательских работ, обосновывающих разработку; схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке.

В зависимости от особенностей разрабатываемого продукта разрешается уточнять содержание разделов, т.е. использовать подразделы, вводить новые разделы или объединять их.

## 4 Расчетно-пояснительная записка

### 4.1 Содержание РПЗ

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

Пример содержания расчетно-пояснительной записки приведен в таблице 2.

Таблица 2 – Пример содержания расчетно-пояснительной записки (при выполнении разработки по технологии ООП)

Наименование раздела	Номер страницы от начала
Титульный лист	1 (не ставится)
Задание на курсовую работу	2
Реферат (в содержании не указывается.)	3
Содержание (в содержании не указывается)	4
Введение	5
1. Анализ требований и уточнение спецификаций	6
1.1. Анализ задания и выбор технологии, языка и среды разработки	6
1.2. Разработка диаграммы вариантов использования	8
1.3. Разработка концептуальной модели предметной области	10
1.4. Выбор методов решения задачи и разработка основных алгоритмов предметной области	12



Наименование раздела	Номер страницы от начала
2. Проектирование структуры и компонентов программного продукта	14
2.1. Разработка интерфейса пользователя	14
2.1.1. Посторонние графа (диаграммы) состояний интерфейса	14
2.1.2. Разработка форм ввода-вывода информации	16
2.3. Разработка структурной схемы программного продукта	19
2.4. Проектирование классов для реализации интерфейса и предметной области	21
3. Выбор стратегии тестирования и разработка тестов	22
Заключение	25
Литература	26
Приложение А. Техническое задание.	
Приложение Б. Руководство пользователя.	

При структурном подходе к разработке или исходя из потребностей можно дополнять РПЗ следующими темами (параграфами):

- «1.2. Анализ процесса обработки информации и построение функциональных диаграмм»;
- «1.3. Анализ хранимой информации и выбор структур данных для ее представления»;
- «1.3. Выбор методов и алгоритмов решения задачи»;
- «2.2. Разработка алгоритма основной программы и структурной схемы программного продукта»;
- «2.3. Разработка основных алгоритмов программного продукта».

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

## 4.2 Реферат

Реферат должен содержать:

- сведения об объеме, количество иллюстраций, таблиц, приложений, количество частей, количество используемых источников;
- перечень ключевых слов, записываемы прописными буквами;
- текст реферата.

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

Текст реферата должен отражать:

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

Пример реферата приведен в приложении А.

## 4.3 Введение

В этом разделе обосновывается актуальность разработки, по возможности, анализируются существующие аналоги, и проводится их сравнение с разрабатываемой системой, указываются преимущества разрабатываемого продукта над аналогами.

## 4.4 Анализ требований и уточнение спецификаций

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

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

Далее определяются спецификации разрабатываемого программного обеспечения в виде одной или нескольких моделей:

- сценариев использования и диаграммы вариантов использования (рисунок 1);

- концептуальной диаграммы классов (рисунок 2);
- математических моделей;
- функциональных диаграмм (рисунок 3);
- моделей и структур представления данных и т.п.

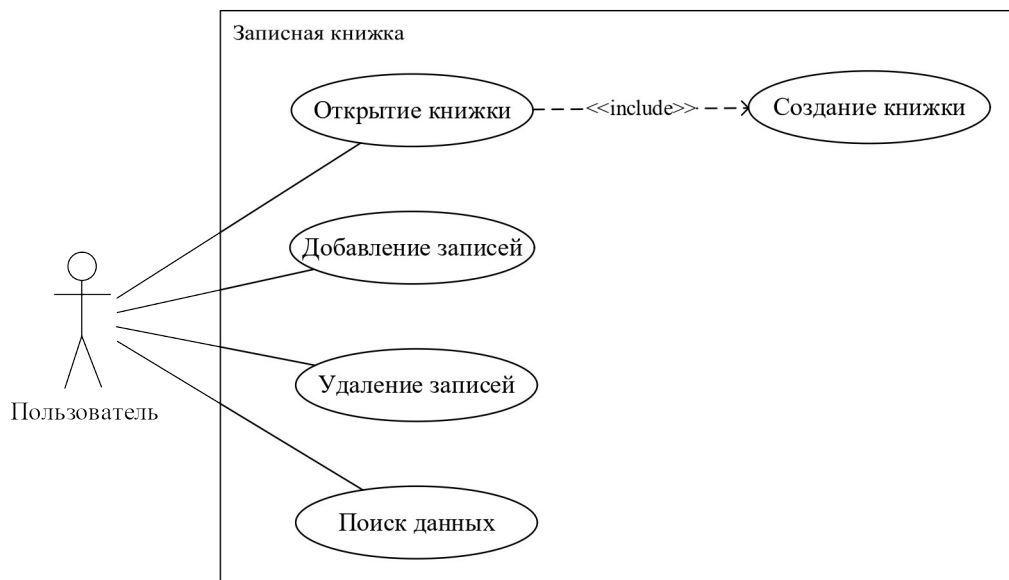


Рисунок 1 – Диаграмма вариантов использования

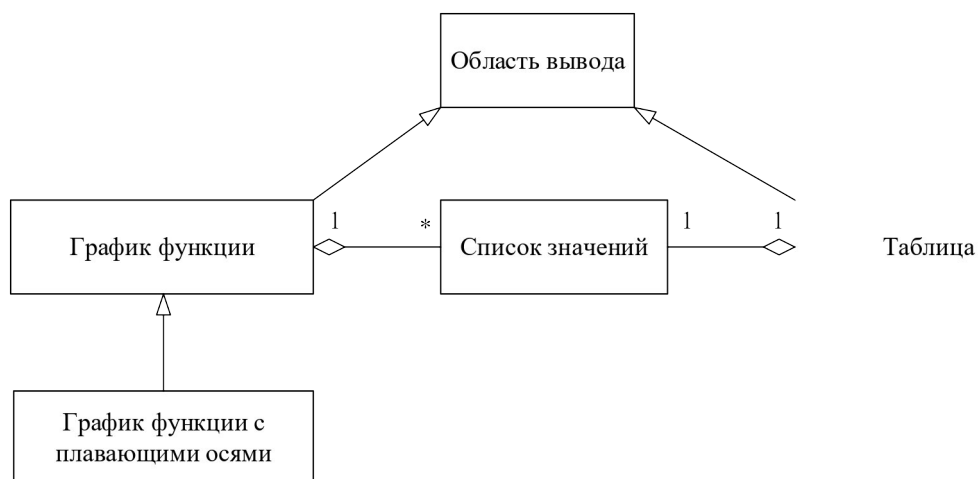


Рисунок 2 – Концептуальная диаграмма классов

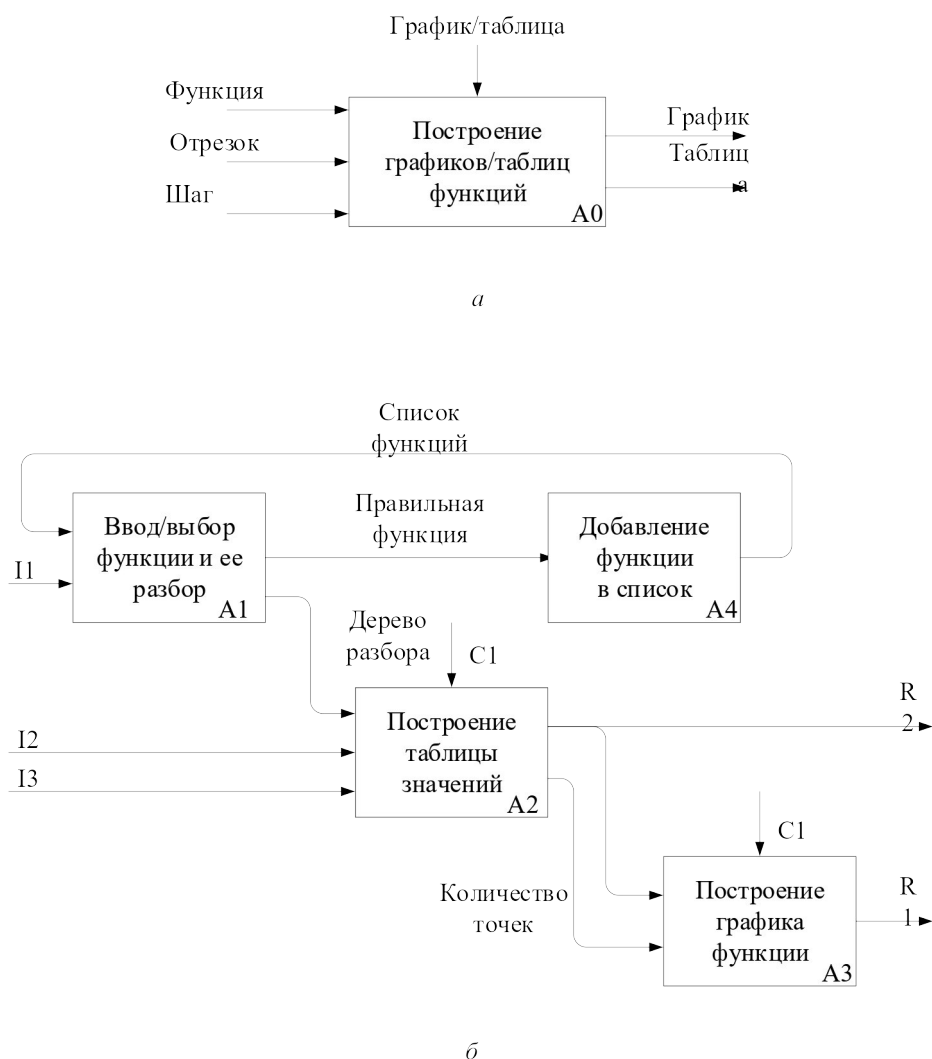


Рисунок 3 – Функциональные диаграммы: *а* – общая часть и *б* – детализация

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

При наличии сложно организованной информации необходим выбор структур для представления данных, который осуществляется на основе анализа основных процессов обработки данных (статические или динамические, массивы или другие структуры). При необходимости создаются новые структуры данных или модифицируются уже известные. Обычно при выборе структур учитываются следующие параметры: объем и типы данных, а также основные операции над данными (хранение, поиск, сортировка) и частота обращения к ним в процессе выполнения программы. Если возможны варианты, то производится их оценка по объему требуемой памяти и вычислительной сложности выполнения основных операций.

## 4.5 Проектирование структуры и компонентов программного продукта

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

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



Рисунок 4 – Экранная форма

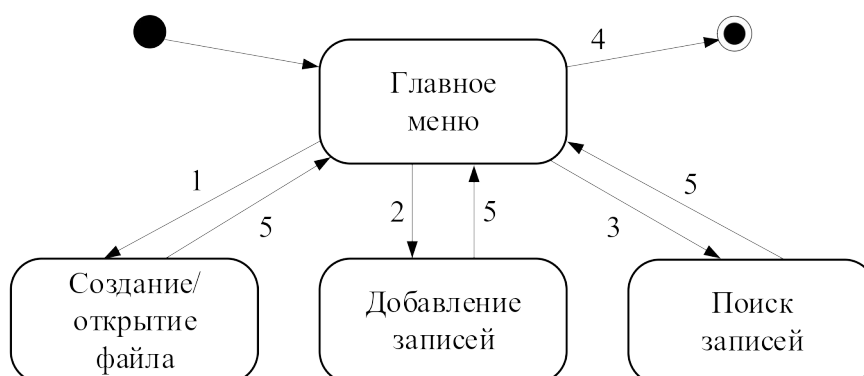


Рисунок 5 – Диаграмма состояний интерфейса

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

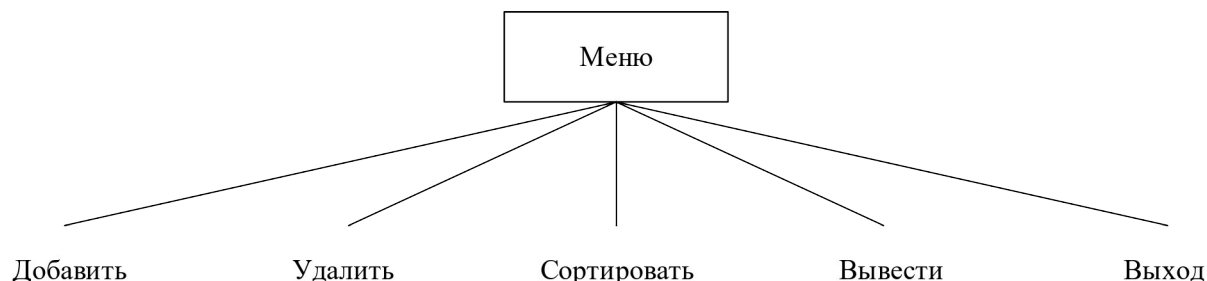


Рисунок 6 – Иерархии меню

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

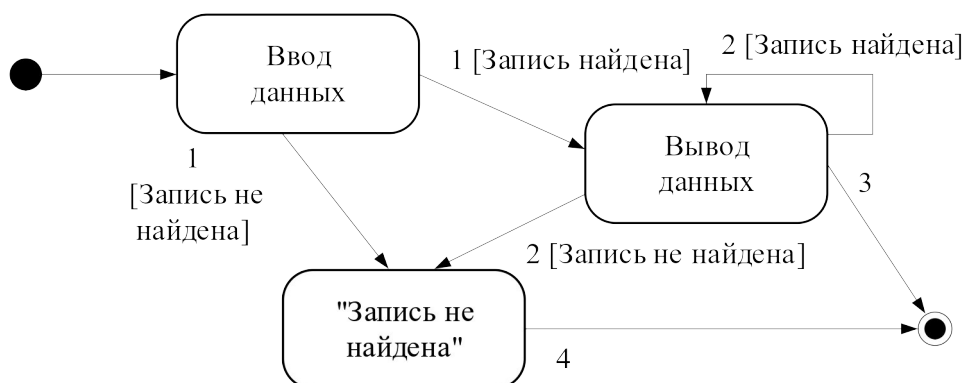


Рисунок 7 – Диаграмма состояний формы интерфейса (при событийном программировании)

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

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

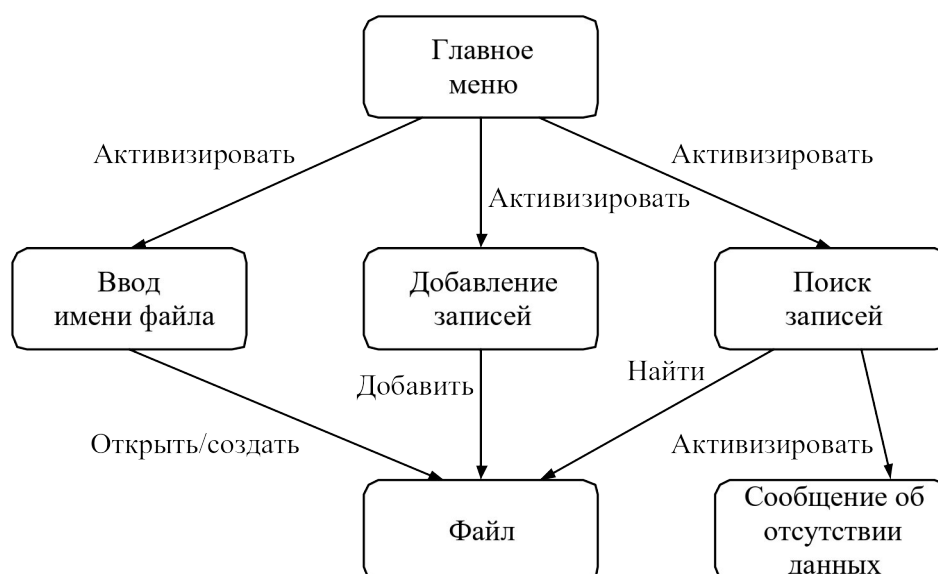


Рисунок 8 – Структурная схема программного продукта (объектная декомпозиция)

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

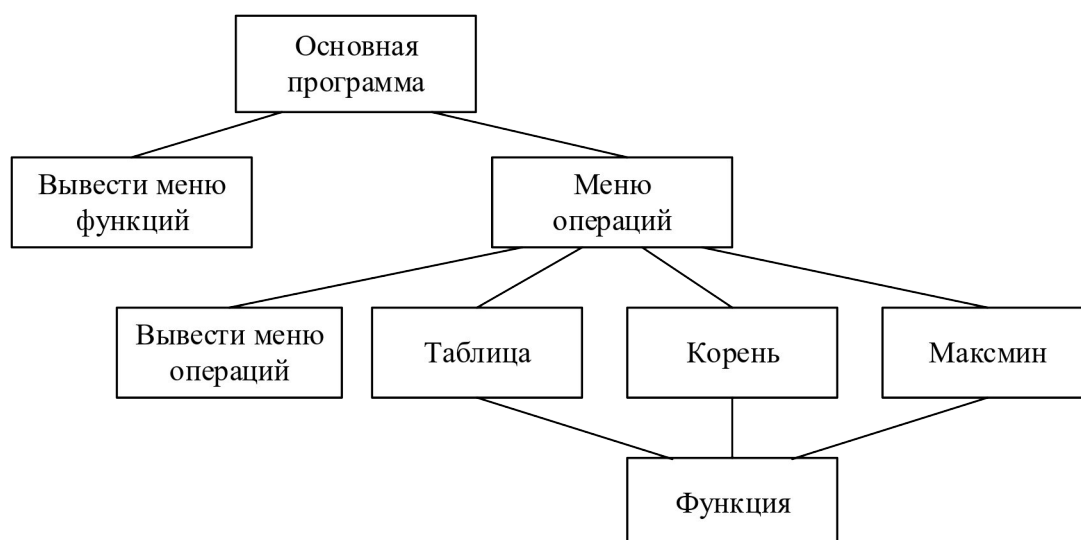


Рисунок 9 – Структурная схема программного продукта (процедурная декомпозиция)

Далее описывается проектирование компонентов в соответствии с выбранной технологией. Для программы, использующей структурный подход к программированию, в данном разделе приводятся обобщенные алгоритмы, например, алгоритм основной программы (рисунок 10) и описывается межпрограммный интерфейс подпрограмм.

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

**Примечание** – Каждый алгоритм должен быть представлен:

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

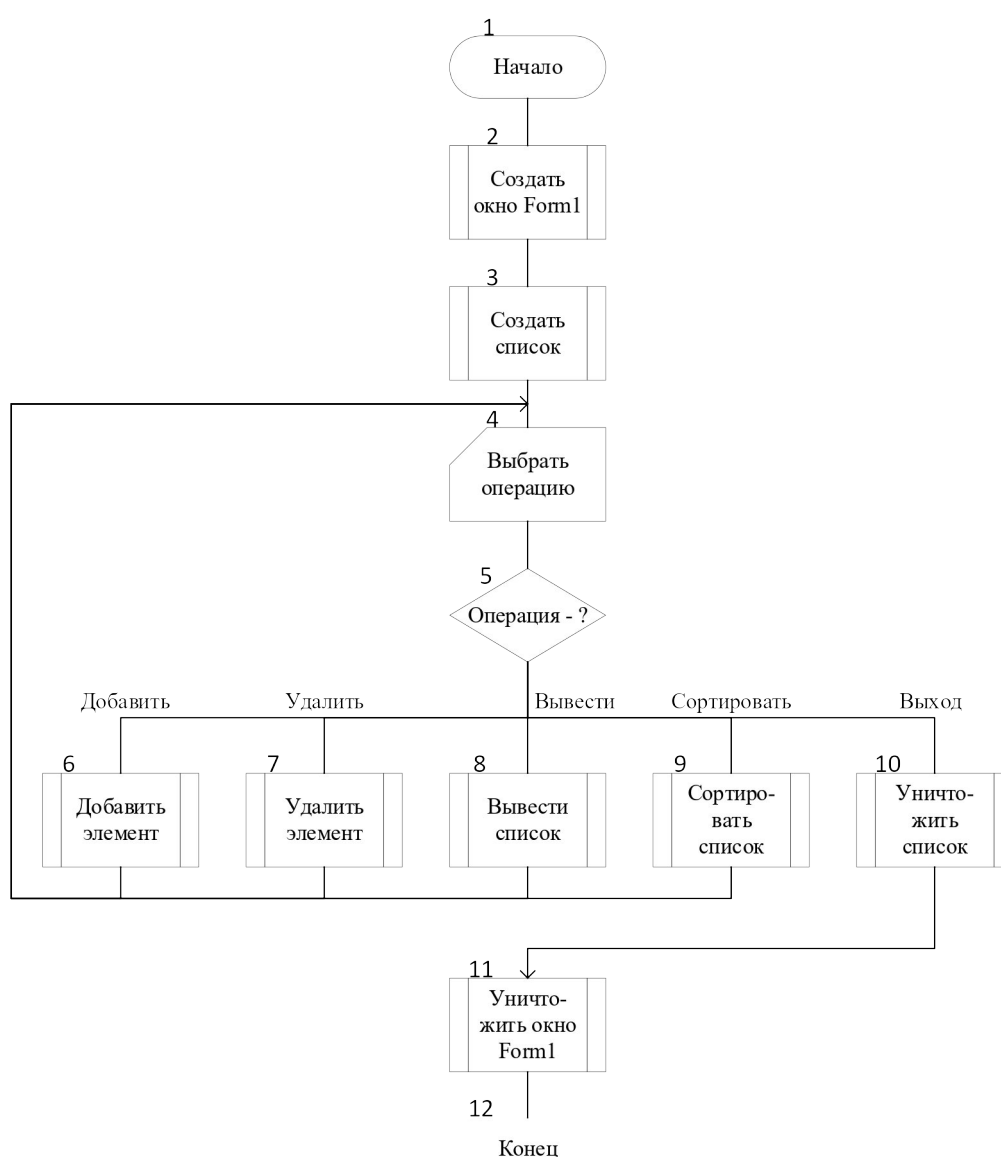


Рисунок 10 – Схема алгоритма основной программы (процедурное программирование)

Описание каждого алгоритма должно включать:

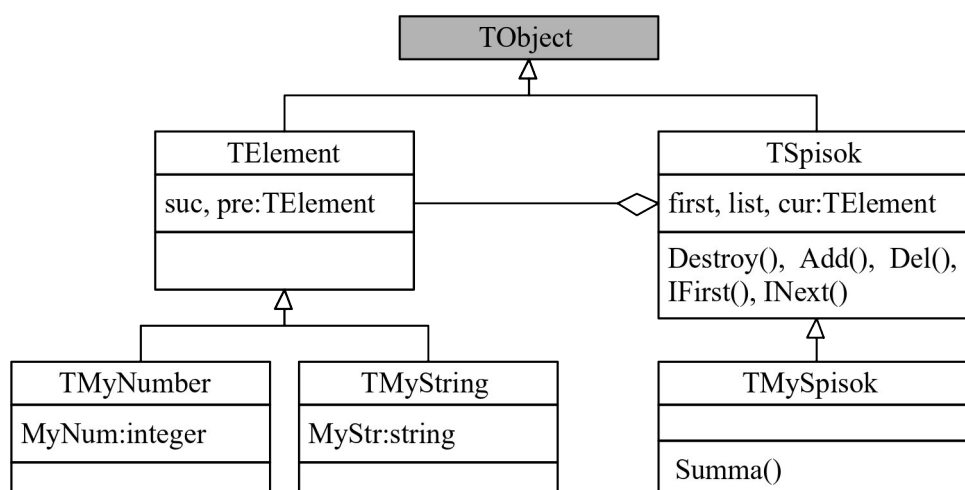


- функциональное назначение алгоритма;
- входные и выходные данные (результаты выполнения);
- список формальных параметров и их назначение;
- пример вызова модуля или подпрограммы;
- используемые технические средства;
- ссылку на таблицу переменных алгоритма;
- ссылку на рисунок со схемой алгоритма;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

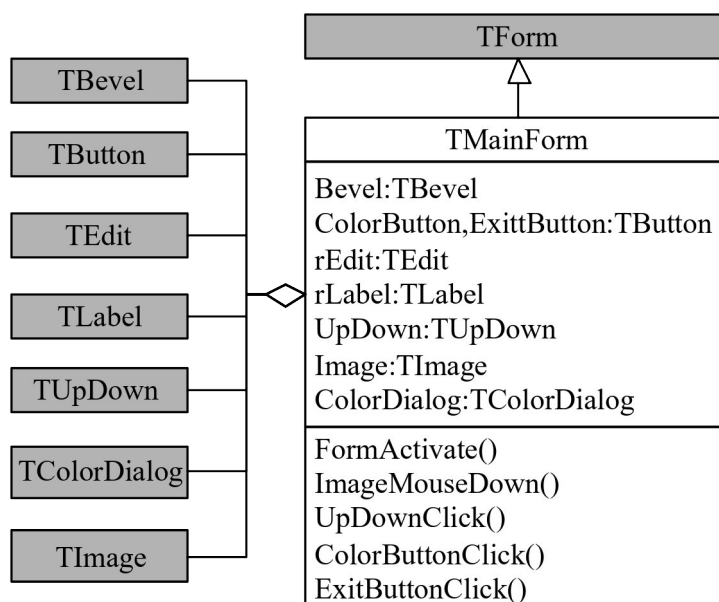
При описании процесса обработки данных в соответствии со схемой алгоритма необходимо пояснить все циклы, каждую альтернативу ветвления, принятое решение по результатам анализа альтернатив и последующие действия.

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

Для программы, при разработке которой использовалась объектно-ориентированная технология, обязательно должна описываться иерархия или диаграмма классов (рисунок 11). Для каждого класса желательно указать дополнительные поля и методы, соответственно обосновывая их назначение и функции. При необходимости здесь же можно привести алгоритмы некоторых методов.



a



б

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

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

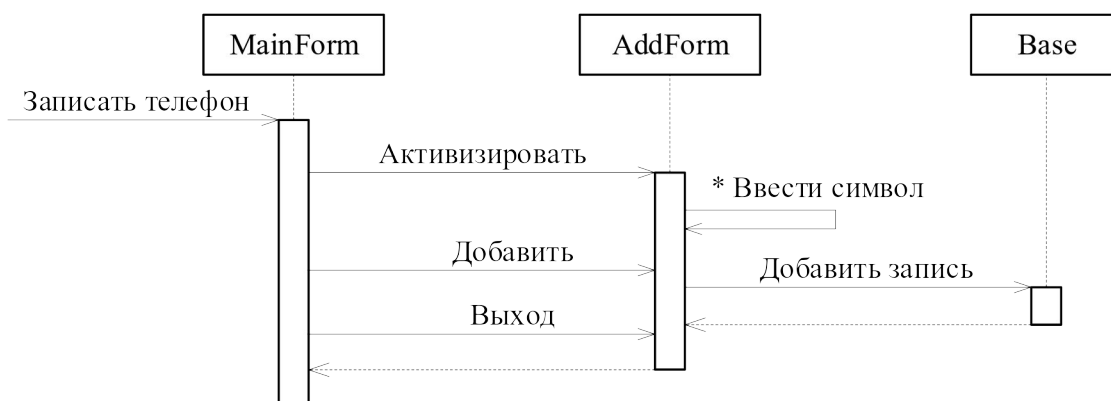


Рисунок 12 – Диаграмма последовательности действий

В завершении раздела описывается декомпозиция разрабатываемой программы на модули и приводится диаграмма пакетов программного продукта (рисунок 13). Также может быть представлена диаграмма развертывания.

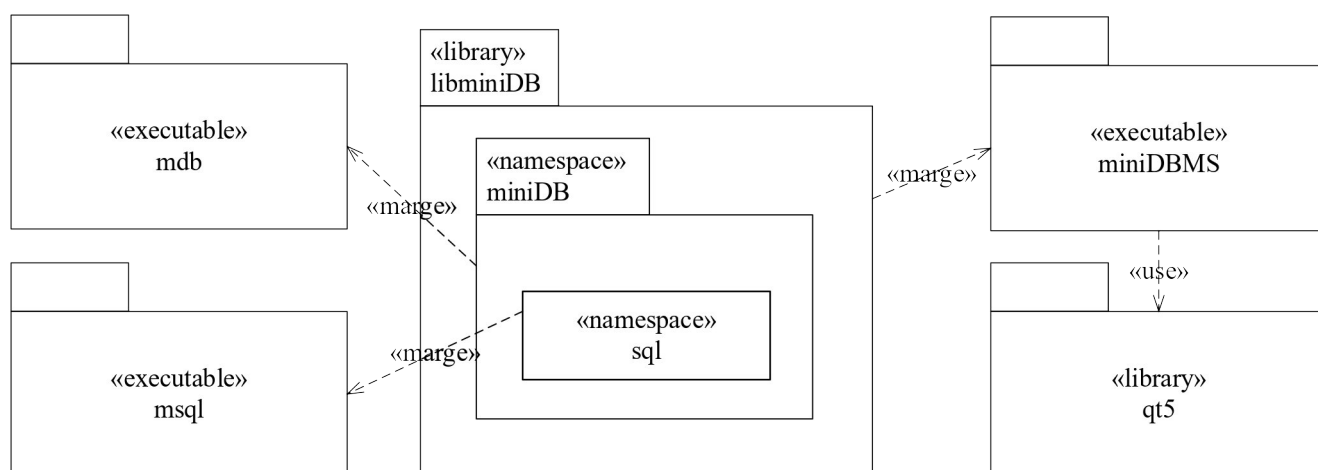


Рисунок 13 – Диаграмма пакетов программного продукта

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

## 4.6 Выбор стратегии тестирования и разработка тестов

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

## **4.7 Заключение**

Раздел содержит выводы по разработанному продукту. Приводятся характеристики продукта и результаты их сравнения с аналогами. Также даются рекомендации по использованию полученного программного продукта и указываются возможные направления его дальнейшего совершенствования.

## 5 Правила оформления документации

### 5.1 Стандарты оформления документов

Основные правила оформления программной документации представлены в стандарте ЕСПД – Единой системе программной документации.

В стандартах ЕСПД устанавливают требования, регламентирующие разработку, сопровождение, изготовление и эксплуатацию программ, что обеспечивает возможность:

- унификации программных изделий для взаимного обмена программами и применения ранее разработанных программ в новых разработках;
- снижения трудоемкости и повышения эффективности разработки, сопровождения, изготовления и эксплуатации программных изделий;
- автоматизации изготовления и хранения программной документации.

Стандарты ЕСПД носят рекомендательный характер, поэтому не обязательны для исполнения и применяются по взаимной договоренности сторон.

ЕСПД включает большое число стандартов, среди которых чаще всего используются следующие:

- ГОСТ 19.102-77. ЕСПД. Стадии разработки;
- ГОСТ 19.105-78. ЕСПД. Общие требования к программным документам;
- ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом;
- ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению;
- ГОСТ 19.202-78. ЕСПД. Спецификация. Требования к содержанию и оформлению;
- ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению;
- ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.

При оформлении расчетно-пояснительной записки следует руководствоваться правилами ГОСТ 7.32-2017 «Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления».

## 5.2 Оформление текстового и графического материала

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

Нумерация страниц – сквозная. Номер проставляется внизу по центру арабской цифрой. Страницами являются листы с текстами, рисунками и текстами приложения.

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

Вторая страница – задание на специальном бланке.

Третья страница – реферат.

Четвертая страница – содержание, отражающее состав изложенного материала. Ни реферат, ни само содержание в нем не упоминаются.

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

Записка завершается списком литературы.

При необходимости после содержания может идти раздел «термины и определения», содержащий перечень сокращений и обозначений.

Далее должны следовать приложения, содержащие материал, не вошедший в записку по причине ее ограниченного размера, но интересный для более глубокого понимания назначения и возможностей разработки. Расчетно-пояснительная записка может содержать одно и более приложений.

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

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

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

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

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

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

### **5.3 Оформление рисунков, схем алгоритмов, таблиц и формул**

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

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

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

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

Например:

Рисунок 12 – Форма окна основного меню

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

Слово «рисунок» и его наименование располагается посередине строки.

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

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

Рисунок 12 – Продолжение

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

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

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

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

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

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

Оформление таблиц должно соответствовать ГОСТ 1.5 и ГОСТ 2.105.

Пример оформления таблицы приведен на рисунке 14.

Таблица \_\_\_\_\_ - \_\_\_\_\_

Головка {	номер		наименование таблицы			} Заголовки граф
						} Подзаголовки граф
						Строки } (горизонтальные ряды)
Боковик (графа для заголовков)		Графы (колонки)				

Рисунок 14 – Пример оформления таблицы



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

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

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

$$z := \sin(x) + \ln(y); \quad (12)$$

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

Оформление примечаний.

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

Пример одного примечания:

Примечание – \_\_\_\_\_

Пример нескольких примечаний:

Примечания

1 \_\_\_\_\_

2 \_\_\_\_\_

3 \_\_\_\_\_

## 5.4 Оформление текстов программ

Тексты программ должны оформляться в соответствии с «хорошим стилем» программирования, т.е. должны быть легко читаемы и хорошо документированы. Рекомендуется использовать комментарии в формате генераторов документации `doxygen`, `javadoc`, `rdoc` и пр. В текстах должны быть комментарии:

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

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

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

## 5.5 Оформление приложений

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

Например:

ПРИЛОЖЕНИЕ В

Титульный лист расчетно-пояснительной записки

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

- рисунок А.12      – 12-й рисунок приложения А;
- рисунок В.2      – 2-й рисунок приложения В;
- таблица Г.3      – 3-я таблица приложения Г.

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

Рисунок А.3 – Файл **mod1.pas** – исходный текст библиотеки процедур обработки;

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

## 5.6 Оформление списка литературы

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

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

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

Далее приводятся примеры оформления библиографических описаний различных источников.

### **Статья в периодических изданиях и сборниках статей:**

1 Гуреев В.Н., Мазов Н.А. Использование библиометрии для оценки значимости журналов в научных библиотеках (обзор)//Научно-техническая информация. Сер. 1. - 2015. - N 2. - С. 8 - 19.

2 Колкова Н.И., Скипор И.Л. Терминосистема предметной области "электронные информационные ресурсы": взгляд с позиций теории и практики//Научн. и техн. б-ки. - 2016. - N 7. - С. 24 - 41.

### **Книги, монографии:**

1 Земсков А.И., Шрайберг Я.Л. Электронные библиотеки: учебник для вузов. - М: Либерия, 2003. - 351 с.

2 Костюк К.Н. Книга в новой медицинской среде. - М.: Директ-Медиа, 2015. - 430 с.

### **Тезисы докладов, материалы конференций:**

1 Леготин Е.Ю. Организация метаданных в хранилище данных//Научный поиск. Технические науки: Материалы 3-й науч. конф. аспирантов и докторантов/отв. за вып. С.Д. Ваулин; Юж.-Урал. гос. ун-т. Т. 2. - Челябинск: Издательский центр ЮУрГУ, 2011. - С. 128 - 132.

2 Антопольский А.Б. Система метаданных в электронных библиотеках//Библиотеки и ассоциации в меняющемся мире: Новые технологии и новые формы сотрудничества: Тр. 8-й Междунар. конф. "Крым-2001"/г. Судак, (июнь 2001 г.). - Т. 1. - М., 2001, - С. 287 - 298.

3 Парфенова С.Л., Гришакина Е.Г., Золотарев Д.В. 4-я Международная научно-практическая конференция "Научное издание международного уровня - 2015: современные тенденции в мировой практике редактирования, издания и оценки научных публикаций"//Наука. Инновации. Образование. - 2015. - N 17. - С. 241 - 252.

#### **Патентная документация согласно стандарту ВОИС:**

1 ВУ (код страны) 18875 (N патентного документа) С1 (код вида документа), 2010 (дата публикации).

#### **Электронные ресурсы:**

1 Статистические показатели российского книгоиздания в 2006 г.: цифры и рейтинги [Электронный ресурс]. - 2006. - URL: [http://bookhamber.ru/stat\\_2006.htm](http://bookhamber.ru/stat_2006.htm) (дата обращения 12.03.2009).

2 Прогноз научно-технологического развития Российской Федерации на период до 2030 года. - URL: <http://government.ru/media/files/41d4b737638891da2184/pdf> (дата обращения 15.11.2016).

3 Web of Science. - URL: <http://apps.webofknowledge.com/>(дата обращения 15.11.2016).

#### **Нормативные документы:**

1 ГОСТ 7.0.96-2016 Система стандартов по информации, библиотечному и издательскому делу. Электронные библиотеки. Основные виды. Структура. Технология формирования. - М.: Стандартинформ, 2016. - 16 с.

2 Приказ Минобрнауки РФ от 19 декабря 2013 г. N 1367 "Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры". - URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_159671/](http://www.consultant.ru/document/cons_doc_LAW_159671/)(дата обращения: 04.08.2016).

3 ISO 25964-1:2011. Information and documentation - Thesauri and interoperability with other vocabularies - Part 1: Thesauri for information retrieval. - URL: [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber53657](http://www.iso.org/iso/catalogue_detail.htm?csnumber53657) (дата обращения: 20.10.2016).

## 6 Дополнительная информация

### 6.1 Современные инструменты автоматизации разработки ПО

Для того, чтобы уверенно чувствовать себя на рынке, производить надежное ПО, оптимизировать проектные ограничения, необходимо понимать и применять передовые методологии, техники и инструменты автоматизации производства ПО. При выполнении курсовой работы актуальным является использование современных инструментов автоматизации. Далее приведены некоторые из них, сгруппированные по выполняемым функциям:

- системы автоматизации при коллективной работе и системы контроля версий, позволяющие минимизировать риски, связанные с потерей управления и рассогласования кода при коллективной разработке:

- 1) системы контроля версий (Git, Bazaar и др.),
- 2) внешние репозитории для хранения кода и ведения проекта (Github, Launchpad, GitLab и др.),
- 3) средства автоматической сборки проекта, интеграции и развёртывания CI/CD (Travis CI, Circle CI, Jenkins, GitLab и др.),
- 4) системы командной работы для отслеживания ошибок, доработок, управления изменениями, обсуждения (Jira, GitLab и др.);

- сдвиг всех проверок как можно ближе к началу проекта или этапа (спринга) с целью минимизации стоимости ошибки. К инструментам этой группы можно отнести:

- 1) статические методы тестирования кода (Clang-Tidy, CppCheck и др.),
- 2) методы формальной верификации,
- 3) автоматические методы тестирования, юнит-тесты;

- минимизация затрат на сопровождение кода (самодокументирование):

- 1) описание проекта в коде проекта (Doxigen и др.),
- 2) средства реверс-инжиниринга с целью построения схем описания взаимодействия модулей, классов и других частей проекта (Umbrello, MagicDraw, Graphviz и др.).

Использование приведённых или подобных инструментов при выполнении курсовой работы будет поощряться начислением дополнительных баллов при сдаче программного продукта по усмотрению руководителя курсовой работы.

## 6.2 Регистрация и работа с репозиторием кода

В МГТУ им. Н.Э. Баумана развёрнута система управления репозиториями кода GitLab по адресу: <https://bmstu.codes> [5]. Для размещения своего проекта на этой площадке нужно выполнить регистрацию и создать проект с ограниченным уровнем доступа (доступ должен быть возможен любому зарегистрированному пользователю). Ссылку на свой проект в репозитории нужно передать своему руководителю курсовой работы.

## 6.3 Полезные ссылки

### Титулы и шаблоны документов курсовой работы

Актуальные методические указания, титулы и шаблоны документов данной курсовой работы можно найти на следующих ресурсах:

- вузовском репозитории кода (<https://bmstu.codes/ru6/docs/tdss>) [6],
- кафедральном портале электронных документов (<http://e-learning.bmstu.ru/moodle>) [7].

Документы могут быть представлены в следующих форматах:

- TeX,
- OpenDocument Format (расширение «odt», каталог «odf»)
- Microsoft Word (расширение «docx», каталог «doc»).

### Примеры курсовых работ за прошлые годы

Может быть полезным посмотреть оформление и реализацию некоторых курсовых работ за прошлые годы, которые размещены на вузовском репозитории кода по адресу <https://bmstu.codes/ru6/courseworks/3tdss> [8].

### Работа с Doxygen

Хорошее вводное описание работы с Doxygen на русском языке приведено в статье «[Документируем код эффективно при помощи Doxygen](#)» [9].

Пример оформления кода с применением данной техники можно посмотреть в следующем проекте: <https://bmstu.codes/lxx/elective/minidb> [10].

## 7 Литература

### 7.1 Основная литература по дисциплине

- 1 Иванова Г.С. Технология программирования: Учеб. для вузов. – М.: «Кнорус», 2013.
- 2 Иванова Г.С. Ничушкина Т.Н., Объектно-ориентированное программирование. Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014.
- 3 Иванова Г.С. Программирование: Учеб. для вузов. – М.: «Кнорус», 2014.
- 4 ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – [введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст]

### 7.2 Дополнительные учебные материалы

- 5 Система управления репозиториями кода МГТУ им. Н.Э.Баумана. – URL: <https://bmstu.codes>
- 6 Актуальные методические указания, титулы и шаблоны документов курсовой работы по дисциплине «Технология разработки программных систем» в системе управления репозиториями кода МГТУ им. Н.Э.Баумана. – URL: <https://bmstu.codes/iu6/docs/tdss>
- 7 Актуальные методические указания, титулы и шаблоны документов курсовой работы по дисциплине «Технология разработки программных систем» на кафедральном портале электронных документов. – URL: <http://e-learning.bmstu.ru/moodle>
- 8 Примеры курсовых работ за прошлые годы в системе управления репозиториями кода МГТУ им. Н.Э.Баумана. – URL: <https://bmstu.codes/iu6/courseworks/3tdss>
- 9 Документируем код эффективно при помощи Doxygen/Norserium (никнейм автора). – URL: <https://habr.com/ru/post/252101/>
- 10 Пример оформления кода с использованием технологии Doxygen. – URL: <https://bmstu.codes/lxx/elective/minidb>

## **Приложение А**

### **Пример реферата на расчетно-пояснительную записку**

Реферат

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

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

Объектом разработки является автоматизированная система исследования функций одной переменной.

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

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

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

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



### **Перечень условных обозначений, сокращений и терминов**

БД	– База данных
ГОСТ	– Межгосударственный стандарт
ЕСПД	– Единая система программной документации
ИУ6	– Кафедра «Компьютерные системы и сети» факультета «Информатика и системы управления» МГТУ им Н.Э. Баумана
ООП	– Объектно-ориентированное проектирование
ПО	– Программное обеспечение
РПЗ	– Расчетно-пояснительная записка
ТЗ	– Техническое задание