

Подписано электронной подписью:  
Вержицкий Данил Григорьевич  
Должность: Директор КГПИ КемГУ

Дата и время: 2025-04-23 00:00:00

471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Кемеровский государственный университет»  
Кузбасский гуманитарно-педагогический институт

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

**Ю.С. Гаврилова**

## **ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ**

*Методические указания к выполнению курсовой работы по дисциплине «Практикум по программированию»*

*для обучающихся по направлениям подготовки*

*01.03.02 Прикладная математика и информатика, профиль «Интеллектуальный анализ данных»*

*02.03.03 Математическое обеспечение и администрирование информационных систем, профиль «Программное и математическое обеспечение информационных технологий»*

*09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления»*

*09.03.03 Прикладная информатика, профиль «Прикладная информатика в экономике»*

Новокузнецк

2024

УДК [378.147.88:004.43](072)  
ББК 74.484(2Рос-4Кем)я73+32.973-018.1я73  
Г 12

**Гаврилова Ю.С.**

Г 12 Практикум по программированию: методические указания к выполнению курсовой работы по дисциплине для студентов факультета информатики, математики и экономики, обучающихся по направлению подготовки 01.03.02 Прикладная математика и информатика (профиль «Интеллектуальный анализ данных»), 02.03.03 Математическое обеспечение и администрирование информационных систем (профиль «Программное и математическое обеспечение информационных технологий»), 09.03.01 Информатика и вычислительная техника (профиль «Автоматизированные системы обработки информации и управления»), 09.03.03 Прикладная информатика (профиль «Прикладная информатика в экономике») / Ю.С. Гаврилова; Новокузнецкий ин-т (фил.) Кемеров. гос. ун-та. – Новокузнецк : КГПИ КемГУ, 2024 – 42 с.

В работе изложены методические указания к выполнению курсовой работы по дисциплине «Практикум по программированию»: цель, задачи и содержание курсовой работы, структура курсовой работы, требования к ее подготовке и оформлению, процедура защиты, критерии оценивания курсовой работы в балльно-рейтинговой системе, список основной и дополнительной литературы, современных профессиональных баз данных и информационных справочных систем, приложения (титульный лист, задание на курсовую работу, образец введения курсовой работы, примерные темы курсовых работ).

Рекомендовано на заседании  
кафедры математики, физики и  
математического моделирования  
Протокол № 8 от 16.03.2020

Заведующий каф. МФММ  
 /Е.В. Решетникова

Утверждено методической комиссией  
факультета информатики, математики и  
экономики  
Протокол № 1 от 12.09.2024

Председатель методической комиссии  
ФИМЭ  
 / И.А. Жибинова

УДК [378.147.88:004.43](072)  
ББК 74.484(2Рос-4Кем)я73+32.973-018.1я73  
Г 12

© Гаврилова Юлия Сергеевна  
© Федеральное государственное бюджетное  
образовательное учреждение высшего  
образования «Кемеровский государственный  
университет», Кузбасский гуманитарно-  
педагогический институт, 2024  
**Текст представлен в авторской редакции**

## ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ .....	4
ХАРАКТЕРИСТИКА СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ.....	5
СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ.....	7
ОРГАНИЗАЦИОННЫЕ ТРЕБОВАНИЯ К ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ.....	24
ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ.....	27
ОФОРМЛЕНИЕ БИБЛИОГРАФИЧЕСКОГО СПИСКА И БИБЛИОГРАФИЧЕСКИХ ССЫЛОК.....	28
ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	30
КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ .....	31
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА .....	34
СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И СПРАВОЧНЫЕ СИСТЕМЫ.....	36
ПРИЛОЖЕНИЕ 1. Титульный лист .....	37
ПРИЛОЖЕНИЕ 2. Задание на курсовую работу .....	38
ПРИЛОЖЕНИЕ 3. Образец введения курсовой работы .....	39
ПРИЛОЖЕНИЕ 4. Примерные темы курсовых работ по дисциплине «Практикум по программированию».....	41

## ПРЕДИСЛОВИЕ

Настоящие методические материалы адресованы студентам, получающим квалификацию бакалавр по направлениям подготовки: 01.03.02 Прикладная математика и информатика, 02.03.03 Математическое обеспечение и администрирование информационных систем, 09.03.01 Информатика и вычислительная техника, 09.03.03 Прикладная информатика и направлены на оказание помощи студентам в написании курсовой работы по дисциплине «Практикум по программированию».

Курсовая работа по дисциплине «Практикум по программированию» является важным компонентом профессиональной подготовки будущего специалиста в сфере информационных технологий. Согласно учебным планам основных образовательных программ направлений подготовки 01.03.02 «Прикладная математика и информатика», 02.03.03 «Математическое обеспечение и администрирование информационных систем», 09.03.01 «Информатика и вычислительная техника», 09.03.03 «Прикладная информатика» курсовая работа выполняется на втором курсе в третьем семестре.

Целью проектирования курсовой работы является формирование способности студента к самостоятельному анализу и оценке направления программирования, языков программирования и сред разработки с точки зрения их использования для создания программных продуктов, а также формирование навыков применения современных информационных технологий в процессе создания программных продуктов.

В процессе написания курсовой работы осуществляется закрепление, углубление и обобщение знаний, полученных студентами за время обучения, и применение этих знаний к решению конкретных задач.

## ХАРАКТЕРИСТИКА СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ

*Курсовая работа* представляет собой вид учебной работы обучающегося, в которой присутствуют элементы самостоятельного научного исследования.

Курсовая работа – это самостоятельный комплексный проект, при выполнении которого студент решает учебно-исследовательские задачи, носящие творческий характер. Поэтому материал для нее не копируется из интернета, а анализируется, систематизируется, приобретая в результате этого характер оригинальности.

При выполнении курсовой работы обучающийся должен научиться составлять программы высокого качества, легко модифицируемые и простые в обращении с применением современных информационных технологий.

Курсовая работа носит прикладной характер, направлена на практическое применение изученного материала к решению задачи. В рамках дисциплины «Практикум по программированию» обучающиеся изучили современные информационные технологии: технологии программирования, языки программирования и среды разработки. Некоторые из них можно применить для решения задачи, поставленной в курсовой работе. Однако, по желанию обучающийся может использовать в работе информационные технологии, изученные самостоятельно, при условии, что их применение оправдано (более эффективное решение задачи).

Во время выполнения курсовой работы обучающийся должен продемонстрировать умение составлять алгоритм решения прикладной задачи, а затем реализовывать составленный алгоритм в какой-либо среде разработки.

Алгоритм решения прикладной задачи строится на базе концептуальной модели предметной области. Концептуальная модель задачи необходима для того, чтобы конкретизировать требования к алгоритму, поскольку тема курсовой работы представлена в общем виде. Пример: тема «Разработка приложения «Библиотека» может предполагать разработку приложения для

районной библиотеки, где читатели берут бумажные книги, а может означать разработку некоторой оболочки для хранилища электронных книг. Концептуальная модель как раз и содержит описание выбранной обучающимся предметной области. Построение концептуальной модели предметной области осуществляется студентом самостоятельно, однако обязательно проверяется и корректируется руководителем курсовой работы, поскольку ошибка на данном шаге может привести к ошибкам во всей практической части работы.

Алгоритм решения прикладной задачи может быть основан на применении какого-либо известного паттерна проектирования. Обучающийся должен самостоятельно выбрать паттерн проектирования, соответствующий задаче. Если обучающийся не смог самостоятельно выполнить задание этого этапа, он может воспользоваться помощью преподавателя (с последующим выставлением соответствующего количества баллов при оценивании выполнения работы).

Решение прикладной задачи (программная реализация алгоритма) осуществляется обучающимся самостоятельно.

Оформление пояснительной записки к курсовой работе рекомендуется осуществлять параллельно с процессом решения задачи.

# СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ

Структура пояснительной записки (текста) курсовой работы должна способствовать раскрытию избранной темы. Структурные элементы: титульный лист, оглавление, введение, основная часть, заключение, библиографический список (использованные источники) и, при необходимости, приложения.

**Введение** курсовой работы в обязательном порядке должно содержать постановку цели и задач исследования, обоснование его актуальности, теоретической и практической значимости. Пример составления и оформления введения для курсовой работы на тему «Разработка программного приложения для расчета объема фигуры вращения, заданной произвольной кривой» представлен в Приложении 3.

**Актуальность исследования** – это степень важности курсовой работы в данный момент времени в конкретной стадии развития выбранной отрасли знаний. Она часто связана с неизученностью (недостаточной изученностью) темы или возможностью решения определенной задачи практики и охватывает запросы отдельной отрасли науки, региона, части общества.

В качестве **цели** исследования будет обозначен результат, который будет получен в ходе написания курсовой работы. Целью написания курсовой работы является решение конкретной проблемы, обозначенной в теме работы. Поэтому цель следует формулировать в виде достижения результата.

Все темы курсовой работы по дисциплине «Практикум по программированию» представляют собой разработку программного приложения, поэтому начать формулировку цели необходимо со слов: «разработать программное приложение» или «разработать программный продукт». Цель исследования формулируется на основе темы курсовой работы, например, если тема курсовой работы «Разработка программного приложения для расчета объема фигуры вращения, заданной произвольной

кривой», то цель данной работы можно сформулировать следующим образом: «разработать программное приложение, моделирующее фигуру вращения произвольной формы, с последующим расчетом их объема и площади фигуры под графиком».

Следует обратить внимание, что в формулировке цели курсовой работы не нужно указывать конкретный язык программирования или среду разработки, используемые в процессе выполнения работы, поскольку оценка возможности применения различных средств разработки должна быть получена во время проведения исследования, а не определена заранее (см. задание на курсовую работу). Также, при формулировании цели курсовой работы следует избегать дословного совпадения с темой работы.

Постановка **задач** исследования по выбранной теме курсовой работы осуществляется согласно поставленной цели. Задачи представляют собой ряд (как правило, четыре или пять) конкретных последовательных этапов достижения поставленной цели.

Курсовая работа должна содержать теоретическую и практическую часть, это должно отражаться и в задачах написания работы. Задачи курсовой работы взаимосвязаны друг с другом, поэтому они располагаются последовательно раскрытию теоретического и практического материала и соответствуют составленному содержанию работы.

**Теоретическая и практическая значимость** исследования формулируются на основании возможности последующего применения результатов, полученных в курсовой работе.

Основная часть пояснительной записки курсовой работы должна иметь двухчастную структуру, т.е. курсовая работа состоит из двух глав, а каждая глава, в свою очередь, из параграфов.

Рекомендуется придерживаться следующего порядка изложения текста курсовой работы (в качестве примера приведем структуру курсовой работы по теме «Разработка программного приложения для создания квестов»):

## СОДЕРЖАНИЕ

Введение.....	3
1 Проектирование программного приложения.....	4
1.1. Концепция проекта.....	4
1.2. Распределение ресурсов проекта.....	6
1.3. Сбор и анализ требований.....	8
1.4. Проект приложения.....	12
2 Разработка программного приложения.....	16
2.1. Разработка хранилища данных.....	16
2.2. Разработка приложения.....	18
2.3. Тестирование приложения.....	27
Заключение.....	30
Список литературы.....	31

**Концепция проекта** (концептуальная модель) – это краткое содержание предметной области в виде небольшого реферативного описания: термины и понятия предметной области, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности.

*Пример: если предметной областью исследования являются процессы регистрации транспортного средства в госавтоинспекции, то необходимо описать, какая информация будет храниться в приложении (регистрационный номер, цвет автомобиля, год выпуска, адрес проживания владельца, наложенные штрафы) и в каком виде; запросы, которые должны будут выполняться во время работы приложения (какая информация будет выводиться в результате работы приложения).*

**Распределение ресурсов проекта** – это разделение человеческих ресурсов по выполняемым работам (план-график работ) и финансовый план.

План-график работ должен представлять собой уточнение таблицы 1 (с конкретными исполнителями проекта). Например, для этапа разработки (или разработки прототипа) необходимо выделить отдельно разработку хранилища

данных, разработку интерфейса, фронтенд и бэкенд частей приложения и т.п.

Таблица 1 – План-график работ над проектом

Роль	Ожидаемый результат	Срок выполнения
Аналитик	Предпроектное обследование предметной области: - описание предметной области; - выбор метода сбора требований и его обоснование; - сбор и анализ требований.	<b>30.09.2024</b>
Аналитик + технический писатель	Составление спецификации требований.	<b>30.09.2024</b>
Аналитик + проектировщик	Математическое и информационное обеспечение продукта: - входные и выходные данные (структура базы данных, описание структуры файлов и т.д.); - проект приложения (структурная схема приложения в целом и отдельных частей при необходимости, диаграмма классов, диаграмма объектов, диаграмма переходов состояний интерфейса); - черновой прототип интерфейса (отрисовка экранов и переходов между ними).	<b>07.10.2024</b>
Аналитик + тестировщик	Приемочное тестирование: - качественные и количественные критерии оценки готовности будущего приложения.	<b>30.09.2024</b>
Проектировщик + разработчик	Разработка прототипа.	<b>02.12.2024</b>
Разработчик	Разработка программного продукта.	<b>30.12.2024</b>
Тестировщик	План тестирования, тестовые сценарии, результат тестирования.	<b>13.01.2025</b>
Технический писатель	Руководство пользователя.	<b>13.01.2025</b>
Аналитик + проектировщик + разработчик + технический писатель	Руководство разработчика: - первичное (структура проекта) от аналитика и проектировщика; - окончательное от разработчика.	<b>07.10.2024</b> <b>13.01.2025</b>
Проект-менеджер	- Концепция проекта. План работ, распределение ресурсов (время, люди). - Реклама.	<b>16.09.2024</b> <b>13.01.2025</b>
Аналитик + проектировщик + проект-менеджер + технический писатель	Роль контент-менеджера (формирование контента, наполнение базы данных) при необходимости.	<b>13.01.2025</b>

Планирование финансовых ресурсов проекта представляет собой расчет

стоимости разработки приложения на основе анализа стоимостной информации по каждому виду работ, выполняемых в проекте. В тексте курсовой работы приводится только итог расчета, представленный в виде таблицы. Так, например, на рисунках 1 и 2 представлены расчеты стоимости разработки веб-сайта.

Наименование работ	Результат	Сроки	Цена
Система администрирования Business	Ключ к системе администрирования UMI		19 900
<b>Разработка структуры сайта</b>	На данном этапе разрабатывается и согласовывается техническая документация, устанавливающая требования к сайту.		
Написание технического задания.	Пакет технической и организационной документации. Техническое задание. Согласование структуры и схемы дизайна сайта. Содержание Технического задания: – программное обеспечение сайта; – дополнительные соглашения.	2 дней	6 000
Составление интерактивных прототипов сайта	Прототипы уникальных страниц сайта из общей структуры	4 дня	8 000
<b>Дизайн сайта</b>	На данном этапе разрабатывается дизайн всех страниц сайта.		
Разработка эскиза «главной» страницы	Изображение в формате png размером решение вида «главной страницы» сайта. В стоимость проекта входит создание одного макета (концепций) дизайна сайта (2 версии главной страницы + типовая внутренняя страница) с учетом пожеланий заказчика и последующей корректировкой избранного варианта (два списка замечаний)	5 дней	20 000
Разработка эскиза «внутренних» страниц	Изображения в формате png размером страницы сайта. (3 различных типов страниц).	4 дня	20 000
<b>Программирование сайта</b>	На данном этапе верстаются шаблоны страниц, производится подготовка материалов для наполнения сайта, разрабатывается необходимое программное обеспечение для работы сайта и его администрирования.		
Верстка макетов сайта	HTML-страницы всех разделов сайта	5 дней	25 000
Интегрирование единой системы управления сайтом, внедрение системы в структуру сайта.		10 дней	
<b>Установка системы администрирования с первоначальной настройкой</b>			
Настройка шаблонов сайта (внедрение верстки). Интерактивный функционал сайта (JavaScript)			22 000
<b>Настройка модулей сайта (Форма обратной связи, Загрузка файлов, Галерея). Наполнение контентом основных страниц, подбор изображений.</b>			18 000
<b>Интеграция системы распознавания эмоций</b>			45000
Тестирование, отладка работы модулей	Размещение сайта в сети Интернет по адресу выбранного доменного имени	3 дня	10 000
<b>Итого:</b>			<b>193 900 руб.</b>

Рисунок 1 – Первый пример расчета стоимости разработки веб-сайта

Наименование работ	Стоимость, руб.	Результат
<b>Разработка структуры и дизайна сайта</b>		
Продумывание структуры сайта, составление технического задания на основе заполненного брифа, изучение конкурентов, SEO-анализ сайта	30000	Пакет документов. Список ключевых фраз для поисковых систем. SEO-аудит
Разработка дизайна главной страницы сайта, в двух вариантах	20000	Эскизы логотипа на бумаге для выбора. Файлы выбранного и доработанного логотипа в формате .ai и .cdr
<b>Верстка страниц сайта</b>		
Верстка макетов	22000	Графическое изображение, представляющее уникальные стилевые решения главной страницы сайта
Создание на основе дизайна работающих html-страниц (верстка).	15000	Готовая верстка основных страниц сайта
Установка, первичная настройка системы управления сайта. Создание структуры страниц.	7000	Работающий "скелет" сайта.
<b>Программирование и оптимизация</b>		
Интеграция готовой html-верстки в систему управления сайтом, программирование нестандартного функционала.	35000	Готовый сайт с пустыми разделами и страницами.
Интеграция системы распознавания эмоций	45000	Запуск системы распознавания эмоций со страниц сайта
Заполнение сайта контентом. SEO оптимизация контента.	16000	Полностью готовый сайт, со всей текстовой и графической информацией
Итоговая сумма	190000	

Рисунок 2 – Второй пример расчета стоимости разработки веб-сайта

**Сбор и анализ требований** к программному продукту должен включать анализ методов, применимых для сбора требований к данному программному продукту, обоснование выбора конкретного метода и описание результатов его применения. Если для сбора требований были использованы методы «Анкетирование» или «Интервью», то необходимо сначала указать стейкхолдеров, а затем привести список вопросов, которые были им заданы. Если выбран метод «Анализ аналогов», то необходимо указать список критериев, по которым проводился анализ, а результат анализа представить в виде таблицы. Например, в таблице 2 представлен анализ аналогов для программного приложения, предназначенного для изучения языка SQL в игровом формате.

Таблица 2 – Анализ аналогов

Информационная система	SQL Murder Mystery	SQL Island	SQL Police Department
Бесплатное использование	Да	Да	Нет
Язык интерфейса	Английский	Английский, Немецкий (по умолчанию), Португальский.	Английский
Требуемый уровень знаний SQL	Продвинутый (оперирование с множеством таблиц, использование запросов с соединением таблиц по условию, использование left join)	Начальный (оперирование с 3 таблицами, простые запросы.)	Средний (использование агрегатных функций, запросы на соединение запросов с соединением таблиц по условию)
Кроссплатформенность	Да	Нет	Да
Возможность сохранения прогресса	Нет	Нет	Да (только платная подписка)
Несколько сюжетов	Нет	Нет	Да (только платная подписка)
Время прохождения	1-2 часа	30-60 минут	Несколько сюжетов
Личная статистика	Нет	Нет	Да (только платная подписка)

Итог процесса сбора требований – сами требования – оформляются в виде списка с указанием вида требований (обычно функциональные и нефункциональные).

Пример: *«При разработке платформы для реализации квестов необходимо учитывать функциональные и нефункциональные требования. Функциональные требования определяют поведение системы и то, что она должна делать, в то время как нефункциональные, определяют, как система должна работать [3]. Рассмотрим сформулированные требования подробнее.»*

### **Функциональные требования:**

1. Платформа должна позволять пользователям создать учетную запись и зарегистрироваться на сайте, чтобы получить доступ к квестам и другим функциям системы.

2. Платформа должна предоставлять возможность создавать свои собственные квесты, включая настройку сюжета, заданий и подсказок для участников.

...

### **Нефункциональные требования:**

1. Платформа должна быть стабильной и надежной, обеспечивать сохранность данных пользователей и минимизировать риски сбоев в работе.

2. ...»

В проекте приложения могут быть представлены алгоритмы различных отдельных функций, а также структурная схема, макеты интерфейса, диаграмма переходов состояний интерфейса, диаграмма классов и диаграмма объектов приложения.

Алгоритм решения прикладной задачи может быть представлен либо в виде блок-схемы, либо в виде нумерованного списка, описывающего последовательность действий. Блок-схема алгоритма должна быть оформлена в соответствии с ГОСТ 19.701-90 (ИСО 5807-85)<sup>1</sup>.

Если в процессе решения задачи используются формулы, их необходимо также привести в тексте курсовой работы.

Пример: 1. Пользователь задает область по оси  $Ox$ , и по оси  $Oy$ , в которой находится график. Размер области построения изображения 660:480 пикселей. Исходя из этого, делается пересчет 1 пикселя для отрисовки по формулам 1-2.

$$L_x = \frac{b_x}{660}, \quad (1)$$

---

<sup>1</sup> см. раздел «Литература для оформления курсовой работы»

$$L_y = \frac{b_y}{480}. \quad (2)$$

Структурная схема разрабатываемого программного приложения позволяет определить взаимоотношения между отдельными частями программы и содержание программных сообщений. Самый простой вид программного продукта – программа – в качестве структурных компонентов может включать только подпрограммы и библиотеки ресурсов. Структурными компонентами программного продукта могут служить программы, подсистемы, библиотеки ресурсов и т. п.

Пример: «В соответствии с проведенным выше анализом задачи разрабатываемая программа должна содержать: последовательное задание исходных данных для эксперимента, которые определяют условия для решения задачи; выбор вида решения и вида результатов (в соответствии с ранее определенными функциями). В результате может быть сформирована структурная схема функционирования интерфейса, которая представлена на рисунке 3.



Рисунок 3 – Структурная схема функционирования интерфейса»

На основе структурной схемы в соответствии с нотацией UML можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д.

Более полное представление о проектируемом программном продукте с точки зрения взаимодействия его компонентов между собой и с внешней средой дает функциональная схема. Функциональная схема или схема данных – схема взаимодействия компонентов ПО с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и

устройств. Для изображения функциональных схем необходимо применять стандарт ГОСТ 19.701–90.

Если разработка программного приложения осуществлялась на основе объектно-ориентированной парадигмы программирования, необходимо в соответствии с нотацией UML представить следующие диаграммы: диаграммы взаимодействия, иерархии классов и т.д.

Диаграмма взаимодействия – это диаграмма, на которой представлено взаимодействие, состоящее из множества объектов и отношений между ними, включая и сообщения, которыми они обмениваются. Этот термин применяется к видам диаграмм с акцентом на взаимодействии объектов. Пример диаграммы взаимодействия для предметной области «Библиотека» представлен на рисунке 4.



Рисунок 4 – UML-диаграмма взаимодействия

Диаграмма иерархии классов – это набор статических, декларативных элементов модели. Информация с диаграммы классов напрямую отображается в исходный код приложения – в большинстве существующих инструментов UML-моделирования возможна кодогенерация для определенного языка программирования. Таким образом, диаграмма классов – конечный результат проектирования и отправная точка процесса разработки. Пример диаграммы классов для предметной области «Доставка товара» представлен на рисунке 5.

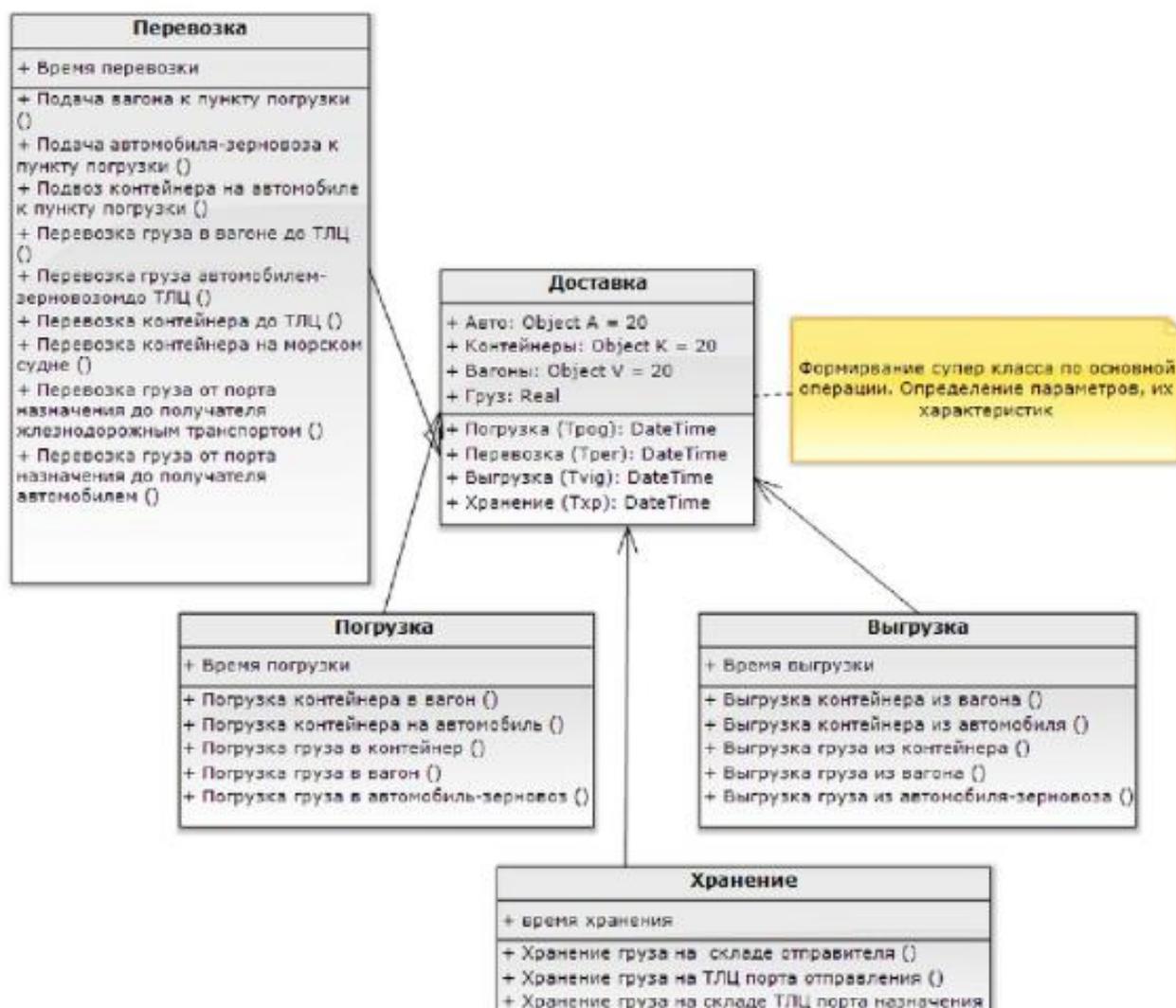


Рисунок 5 – UML-диаграмма иерархии классов приложения

**Разработка хранилища данных** – это универсальное название параграфа для любого проекта. Если в проекте используется база данных, то можно сформулировать название параграфа «**Разработка базы данных**». В параграфе должна быть представлена структура хранилища данных: иерархическая схема для файлового хранилища, структура полей файла, инфологическая модель базы данных и т.п.

Пример: «Схема базы данных (рисунок б) состоит из пяти таблиц:

– **Таблица «Пользователи»** хранит такие данные о зарегистрированных пользователях, как логин, пароль, ник, секретный вопрос и ответ на него.

– **Таблица «Квест»** хранит всю информацию о созданных квестах

(название квеста, описание, автор квеста).

– **Таблица «Вопросы»** хранит информацию о вопросах для каждого квеста, такую как текст вопроса, подсказка, хеш верного ответа, и другую служебную информацию, связанную с каждым вопросом.

– **Таблица «Комната»** хранит информацию о том, какое количество вопросов находится в комнате, и к какому квесту эта комната относится.

– **Таблица «Прохождение»** хранит информацию о том, какие квесты были пройдены каждым пользователем.

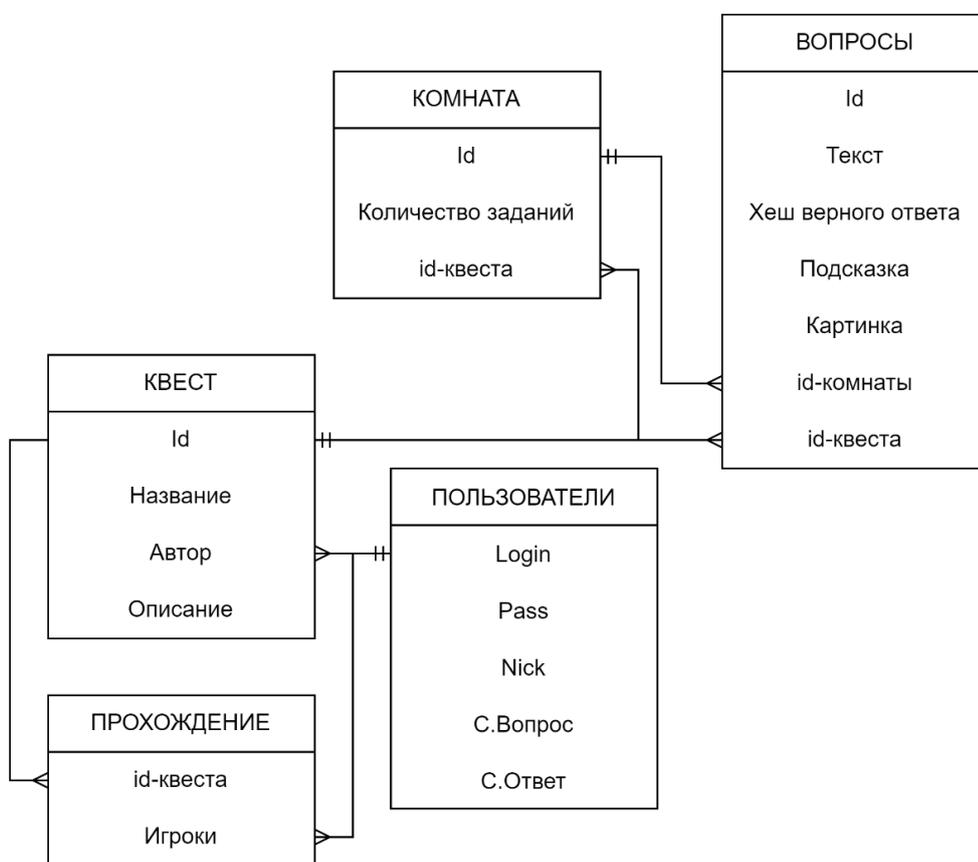


Рисунок 6 – Инфологическая модель базы данных»

Пример: «Хранилище данных для программного приложения, моделирующего Солнечную систему, представляет собой JSON-файл, который содержит подробное описание каждого объекта в модели Солнечной системы, а также его название и основные характеристики. Структура JSON-файла представлена на рисунке 7.

```
data.json
1 [
2   {
3     "id": "Sun",
4     "name": "Солнце",
5     "info": "Средний диаметр\n1,392*10^9 м\nЭкваториальный радиус\n6,9551*10^8 м\nПлощадь поверхности\n6,07877*10^
6   },
7   {
8     "id": "Merk",
9     "name": "Меркурий",
10    "info": "Экваториальный радиус\n2739,7 км\nПлощадь поверхности\n7,48*10^7 км2\nОбъём\n6,083*10^10 км3\nМасса\n
11  },
12  {
13    "id": "Venera",
14    "name": "Венера",
15    "info": "Экваториальный радиус\n6051,8 км\nПлощадь поверхности\n4,60*10^8 км2\nОбъём\n9,38*10^11 км3\nМасса\n4
16  },
17  {
18    "id": "Earth",
19    "name": "Земля",
20    "info": "Экваториальный радиус\n6378,71 км\nПлощадь поверхности\n51*10^7 км2\nОбъём\n1,083*10^12 км3\nМасса\n5
21  },
22  {
23    "id": "Mars",
24    "name": "Марс",
25    "info": "Экваториальный радиус\n3396,2 км\nПлощадь поверхности\n1,44*10^7 км2\nОбъём\n1,631*10^10 км3\nМасса\n

```

Рисунок 7 – JSON файл»

При написании параграфа «**Разработка приложения**» обучающийся может приводить фрагменты кода, если они представляют какую-либо научную или практическую значимость, или если ему необходимо сделать акцент на этой части разработки.

При оформлении фрагментов следует использовать шрифт Courier New, размер – 12 пт, межстрочный интервал – одинарный. Рекомендуется отделять смысловые блоки пустыми строками, а также визуально обозначать вложенные конструкции с помощью отступов.

Ключевые слова и комментарии рекомендуется выделять с помощью различных начертаний шрифта. Таким же образом в основном тексте работы должны обозначаться имена библиотек, подпрограмм, констант, переменных, структур данных, классов, их поля и методы. Следует заметить, что код должен сопровождаться подробным описанием.

Пример: *«Чтобы фон не был однотонным, необходимо создать динамичный фон с изменяющимся рисунком. Для этого был написан скрипт, принцип которого заключается в создании объекта particles, который инициализирует Particles с более сложными настройками, включая цвета, соединение частиц и адаптивные настройки для разных размеров экрана.*

```

window.onload = function () {
  Particles.init({
    selector: ".background"
  });
};
const particles = Particles.init({
  selector: ".background",
  color: ["#03dac6", "#ff0266", "#000000"],
  connectParticles: true,
  responsive: [
    {
      breakpoint: 768,
      options: {
        color: ["#faebd7", "#03dac6", "#ff0266"],
        maxParticles: 43,
        connectParticles: false
      }
    }
  ]
});

```

*Таким образом, этот скрипт отвечает за создание и настройку анимации частиц в элементе с классом `.background`, изменяя их поведение и внешний вид в зависимости от ширины экрана.»*

Необходимо учесть, что стоит рационально подойти к размещению в тексте пояснительной записки фрагментов кода, копировать все строки программы в работу не требуется.

Результат разработки оформляется в виде снимков экрана, сделанных во время функционирования приложения. Выбор конкретных модулей, которые следует показать, осуществляется обучающимся самостоятельно, однако, стоит соблюсти 2 условия:

- наиболее полно продемонстрировать функциональные возможности приложения;
- не допустить избыточности информации.

При необходимости обучающийся может проконсультироваться с преподавателем по поводу представления графического материала (какие модули и окна программного приложения следует поместить в текст курсовой работы).

Чаще всего необходимо представить следующие результаты: главное окно приложения, окно ввода данных и результат, полученный после запуска

программы.

Пример таких изображений представлен на рисунках 8-10.



Рисунок 8 – Главное окно приложения

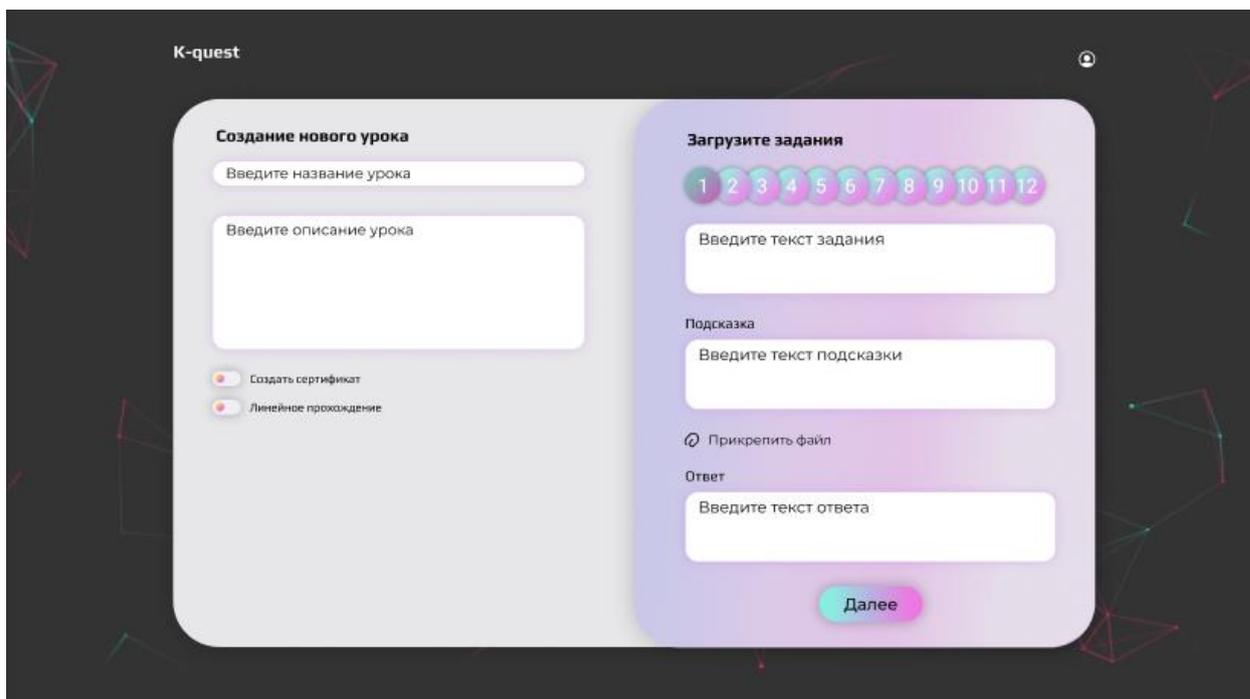


Рисунок 9 – Окно ввода данных для создания квеста



Рисунок 10 – Окно лобби с 4 комнатами

В случае, если программное приложение предназначено для обработки информации и предполагает вывод данных не только на форму приложения, но и в файл (или любое другое хранилище данных), то результат также необходимо представить в тексте курсовой работы (в виде снимка экрана с открытым файлом).

**Тестирование программного приложения** должно быть представлено чек-листом тестирования и баг-репортами. В чек-листе указываются функции программного приложения, тестирование которых было проведено в ходе выполнения курсовой работы. Баг-репорт оформляется отдельно для каждой найденной в процессе тестирования ошибки.

В **заключении** обобщаются наблюдения, сделанные во время работы и формулируются основные выводы, отражающие наиболее значимые результаты работы, предлагаются рекомендации относительно возможностей использования материалов и результатов работы. Обычно обучающийся формулирует выводы в порядке, соответствующем порядку задач курсовой работы (из введения). Кроме того, обучающийся может указать направление дальнейших исследований в рамках развития данной задачи.

При составлении текста пояснительной записки следует помнить, что:

- необходимо избегать логических ошибок, например, нельзя давать одинаковое название курсовой работе и одному из ее параграфов;
- содержание работы необходимо иллюстрировать наглядными материалами: таблицами, рисунками, схемами, графиками и т.п.

# ОРГАНИЗАЦИОННЫЕ ТРЕБОВАНИЯ К ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ

1. Для направлений подготовки 01.03.02 Прикладная математика и информатика (профиль «Интеллектуальный анализ данных»), 02.03.03 Математическое обеспечение и администрирование информационных систем (профиль «Программное и математическое обеспечение информационных технологий») в начале третьего семестра кафедра предлагает студентам второго курса темы курсовых работ по дисциплине «Практикум по программированию». Студент может предложить и свою тему, но ее необходимо обосновать, согласовать с руководителем и зарегистрировать на кафедре. Общее руководство и контроль за ходом выполнения курсовой работы осуществляет руководитель курсовой работы.

2. После выбора темы студенту следует встретиться с руководителем курсовой работы для обсуждения задачи. Руководитель выясняет степень подготовленности студента к работе над темой, рекомендует необходимую литературу и дает консультации о порядке выполнения работы. Для самоорганизации студента и своевременного контроля за ходом работы можно составить график ее выполнения.

*График выполнения курсовой работы по дисциплине «Практикум по программированию»*

студента (ки) группы \_\_\_\_\_

шифр группы

ФИО

№ п/п	Этапы работы	Планируемая дата выполнения	Фактическая дата	Отметка научного руководителя
1	2	3	4	5
1.	Выбор темы, ее регистрация на кафедре, обсуждение целей и задач работы, графика ее выполнения с руководителем курсовой работы. Подбор библиографических источников по теме, согласование их с руководителем.			

2.	Подготовка чернового варианта теоретической части работы и представление его руководителю работы. Работа над замечаниями руководителя курсовой работы, исправление недочетов и повторное представление.			
3.	Выполнение практической части работы: разработка алгоритма решения задачи. Представление руководителю разработанного алгоритма. Исправление недочетов и повторное представление руководителю.			
4.	Выполнение практической части работы: программная реализация разработанного алгоритма. Представление руководителю завершенной разработки. Исправление недочетов и повторное представление руководителю.			
5.	Подготовка чернового варианта практической части работы и представление его руководителю. Работа над замечаниями руководителя. Написание введения и заключения, оформление библиографического списка, титульного листа работы			
6.	Представление руководителю чернового варианта всей работы. Проверка на наличие в тексте работы некорректных заимствований. Исправление недочетов. Подготовка чистового варианта работы и представление его руководителю.			
7.	Защита курсовой работы			

Руководитель курсовой работы: \_\_\_\_\_

Дата « \_\_\_\_\_ » \_\_\_\_\_<sup>ФИО</sup> 2024

3. После беседы с руководителем студент приступает к подбору литературы, ее анализу и составлению плана работы. Кроме рекомендуемой руководителем литературы, желательно начать поиск дополнительной литературы по данной теме. Изучая литературу, полезно делать необходимые выписки (конспектировать), закладки, заметки.

4. Желательно, чтобы в процессе написания курсовой работы студент периодически встречался с руководителем, показывал ему законченные части работы, обсуждал свои идеи и наработки. Это поможет детализировать процесс написания курсовой работы, предостережет от ложных направлений, даст дополнительный импульс к творческому поиску. Руководитель свои рекомендации по выполнению курсовой работы дает лишь после того, как он убедился, что студент достаточно изучил проблему и понял ее суть.

5. Опыт показывает, что даже при самом добросовестном отношении студента, первый вариант курсовой работы потребует существенных исправлений и доработки. Перечитав и выправив черновик, студент передает его руководителю курсовой работы, который принимает решение либо о допуске работы к защите, либо о доработке и устранении имеющихся недочетов и ошибок.

6. Чтобы студент мог успешно защитить курсовую работу, он должен хорошо владеть ее содержанием, уметь доказывать и обосновывать полученные результаты.

## ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

1. Объем курсовой работы зависит от темы. Студенту рекомендуется быть максимально кратким, обходиться без повторов, опускать совершенно очевидные выкладки, расчеты и пояснения. Опыт показывает, что объем курсовой работы колеблется в пределах 25-35 страниц печатного текста.

2. Структурные элементы курсовой работы описаны ранее.

3. Текст курсовой работы оформляется в соответствии с Правилами оформления учебных работ<sup>2</sup>.

4. Текст курсовой работы печатается на одной стороне листа бумаги стандартного размера А4 (210×297 мм). Каждый лист вкладывается в отдельный файл (мультифору), затем все файлы помещаются в скоросшиватель.

5. Курсовая работа должна содержать самостоятельные выводы и авторский текст не менее 70%. Проверка на наличие некорректных зависимостей осуществляется руководителем курсовой работы с помощью системы [antiplagiat.ru](http://antiplagiat.ru) во время предоставления обучающимся на проверку черновой работы. Обучающемуся рекомендуется самостоятельно заранее проверять свою работу по мере написания текста.

---

<sup>2</sup> Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный

## **ОФОРМЛЕНИЕ БИБЛИОГРАФИЧЕСКОГО СПИСКА И БИБЛИОГРАФИЧЕСКИХ ССЫЛОК**

Особое место среди методов исследования занимают методы изучения литературы по проблеме данной работы. Изучение литературы служит средством изучения истории и причин возникновения проблемы, ее современного состояния. Работа с литературой включает несколько методов.

Укажем наиболее важные из них.

1. Составление библиографии, то есть списка литературы, использованной автором при написании курсовой работы.

2. Реферирование – сжатое изложение основного содержания одной или нескольких работ по общей теме.

3. Конспектирование – детальное изложение главных положений и концептуальных идей работы.

4. Аннотирование – краткое, предельно сжатое изложение основного содержания литературных источников.

5. Цитирование – дословная запись высказываний, выражений автора, а также приведение в тексте курсовой работы фактических и статистических данных, содержащихся в литературных источниках.

Еще одно правило работы с литературой – использование библиографического списка в тексте работы: источник, внесенный в список, хотя бы один раз должен быть назван в тексте. И, наоборот, любой источник, на который автор ссылается в тексте работы, должен быть внесен в библиографический список.

При написании курсовой работы наиболее приемлемым является алфавитный способ группировки материалов, включенных в список использованных источников. Использованные источники при этом располагаются в общем алфавитном порядке фамилий авторов или заглавий книг, статей (если автор не указан).

Обращаясь к цитированию, необходимо соблюдать следующие правила:

нельзя вырывать фразы из текста; искажать его произвольными сокращениями; цитату необходимо брать в кавычки и т.п. Следует также обратить внимание на точное указание источников цитат.

Ссылки на использованную литературу в тексте указываются в квадратных скобках порядковым номером по списку источников, например, [23]. Если автор делает ссылку сразу на несколько источников, то в скобках через запятую указываются их номера в списке, например, [23, 25, 38]. Если автор приводит цитату из первоисточника, то в скобках после номера источника обязательно указывается страница, например, [34. С. 216].

Список литературы оформляется в соответствии с требованиями **ГОСТ Р 7.0.100-2018**.

Курсовая работа с неправильно или небрежно оформленным библиографическим списком к защите не допускается.

## ЗАЩИТА КУРСОВОЙ РАБОТЫ

1. Курсовая работа должна быть защищена не позднее начала экзаменационной сессии.

2. Руководитель курсовой работы, ознакомившись с окончательным вариантом курсовой работы, принимает решение о допуске к защите курсовой работы.

3. Во время защиты обучающийся кратко (5 - 7 минут) докладывает основные результаты своей работы, сопровождая результат наглядным материалом (презентация), а затем отвечает на вопросы преподавателя.

В докладе обучающийся озвучивает цель и задачи исследования; используемые при разработке алгоритмические и программные решения; основные результаты разработки; выводы по работе.

4. Преподаватель оценивает, насколько успешно достигнута цель и решены задачи, поставленные в курсовой работе; качество доклада и ответов на вопросы; степень самостоятельности студента при выполнении исследования (анализе технологий программирования, языков программирования и средств разработки); полнота проведенной работы (разработка алгоритма решения задачи, программная реализация алгоритма). Отметка за курсовую работу выставляется согласно приведенной в методических указаниях балльно-рейтинговой системе и таблице перевода баллов в буквенный эквивалент.

5. После этого руководитель курсовой работы заносит отметку в зачетную книжку студента и в соответствующую ведомость.

Курсовая работа хранится на кафедре вплоть до завершения студентом учебы в университете.

## КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ

Баллы по курсовой работе выставляются обучающемуся за два вида деятельности:

- выполнение исследования и оформление пояснительной записки (80 баллов) (таблица 3);
- защита курсовой работы (20 баллов) (таблица 4).

Таблица 3 – Распределение баллов по этапам выполнения исследования

Этапы выполнения исследования	Оценка в аттестации	Баллы
<p>Этап 1. Проектирование программного приложения. 1.1 Концепция. 1.2 Распределение ресурсов проекта. 1.3 Сбор и анализ требований. 1.4 Проект приложения</p>	<p style="text-align: center;"><b>Баллы за этап 1.1:</b></p> <p><b>1 балл</b> – концепция представлена, но в ней имеются серьезные ошибки;  <b>2 балла</b> – в целом концепция представлена, но есть некоторые недочеты;  <b>3 балла</b> – представлено полное описание предметной области и самого приложения, выявлены предполагаемые пользователи.</p> <p style="text-align: center;"><b>Баллы за этап 1.2:</b></p> <p><b>1 балл</b> – план-график работы над проектом и распределение финансовых ресурсов проекта представлены, но в них есть существенные недостатки;  <b>2 балла</b> – план-график работы над проектом и распределение финансовых ресурсов проекта представлены, но есть некоторые недочеты;  <b>3 балла</b> – представлены в полном объеме план-график работы над проектом и распределение финансовых ресурсов проекта.</p> <p style="text-align: center;"><b>Баллы за этап 1.3:</b></p> <p><b>3 балла</b> – требования собраны, но не в полном объеме (или 1 методом);  <b>4 балла</b> – в целом требования собраны, но наблюдаются недочеты в виде неправильной их классификации;  <b>6 баллов</b> – в полном объеме собраны требования к</p>	10-20

	<p>программному продукту (не менее чем 2 методами, один из которых устный или письменный опрос).</p> <p><b>Баллы за этап 1.4:</b></p> <p><b>5 баллов</b> – проект представлен, но в схемах или алгоритме есть серьезные ошибки;</p> <p><b>6 баллов</b> – в целом проект представлен, однако имеются некоторые неточности, или недочеты;</p> <p><b>8 баллов</b> – в полном объеме представлен проект программного приложения.</p>	
<p>Этап 2.</p> <p>Разработка приложения</p> <p>2.1 Разработка хранилища данных</p> <p>2.2 Программная реализация</p> <p>2.3 Тестирование приложения</p>	<p><b>Баллы за этап 2.1:</b></p> <p><b>4 баллов</b> – хранилище данных спроектировано и реализовано, но наблюдаются существенные ошибки в реализации;</p> <p><b>8 баллов</b> – хранилище данных спроектировано и реализовано, но наблюдаются некоторые неточности или недочеты;</p> <p><b>10 баллов</b> – хранилище данных спроектировано и реализовано самостоятельно и в полном объеме, в тексте курсовой работы представлена его структура и способы записи/извлечения данных.</p> <p><b>Баллы за этап 2.2:</b></p> <p><b>30 баллов</b> – программное приложение реализовано не в полном объеме;</p> <p><b>35 баллов</b> – программное приложение реализовано, но наблюдаются некоторые неточности или недочеты;</p> <p><b>40 баллов</b> – программное приложение реализовано в полном объеме.</p> <p><b>Баллы за этап 2.3:</b></p> <p><b>7 баллов</b> – тестирование проведено небрежно и/или не в полном объеме;</p> <p><b>9 баллов</b> – тестирование проведено в полном объеме, но наблюдаются некоторые неточности или недочеты;</p> <p><b>10 баллов</b> – тестирование проведено в полном объеме, представлен чек-лист тестирования и баг-репорты.</p>	41-60
<b>Итого за выполнение курсовой работы</b>		<b>51-80</b>

Таблица 4 – Распределение баллов за защиту курсовой работы

Виды и результаты учебной работы	Оценка в аттестации	Баллы
Устное выступление об основных результатах, полученных во время выполнения курсовой работы (5-7 минут)	<b>6 баллов</b> – пороговое значение; <b>10 баллов</b> – максимальное значение.	6 - 10
Сопровождение устного выступления наглядным материалом (презентация)	<b>2 балла</b> – пороговое значение; <b>5 баллов</b> – максимальное значение.	2 - 5
Ответы на вопросы по теме курсовой работы	<b>2 балла</b> – пороговое значение; <b>5 баллов</b> – максимальное значение.	2 - 5
<b>Итого по промежуточной аттестации (защита курсовой работы)</b>		<b>10-20</b>

Набранные обучающимся баллы переводятся в оценку и буквенный эквивалент согласно Положению о балльно-рейтинговой системе оценки деятельности студентов КемГУ от 30.12.2016г. (таблица 5).

Таблица 5 – Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент

Сумма баллов для дисциплины	Оценка	Буквенный эквивалент
86 - 100	5	отлично
66 - 85	4	хорошо
51 - 65	3	удовлетворительно
0 - 50	2	неудовлетворительно

## РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

### Основная учебная литература

1. Трофимов, В.В. Алгоритмизация и программирование : учебник / В.В. Трофимов, Т.А. Павловская. – Москва : Издательство Юрайт, 2019. – 137 с. – ISBN 978-5-534-07834-3. – URL: <https://www.biblio-online.ru/viewer/algoritmizaciya-i-programmirovanie-423824>.
2. Зыков, С. В. Программирование : учебник и практикум / С. В. Зыков. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2023. – 285 с. – (Бакалавр. Академический курс). – ISBN 978-5-534-16031-4. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/530294> (дата обращения: 22.08.2024).

### Дополнительная литература

1. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. – 5-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2023. – 227 с. – (Высшее образование). – ISBN 978-5-534-17323-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/532868> (дата обращения: 22.08.2024).
2. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. – 2-е изд. – Москва : Издательство Юрайт, 2024. – 150 с. – (Высшее образование). – ISBN 978-5-534-16942-3. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/537721> (дата обращения: 22.08.2024).
3. Маркин, А. В. Программирование на SQL : учебник и практикум для вузов / А. В. Маркин. – 3-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2024. – 805 с. – (Высшее образование). – ISBN 978-5-534-18371-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/534873> (дата обращения: 07.09.2024).
4. Огнева, М. В. Программирование на языке C++: практический

курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. – Москва : Издательство Юрайт, 2023. – 335 с. – (Высшее образование). – ISBN 978-5-534-05123-0. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/515142>.

5. Python : сайт. – 2001 - . - URL: <https://www.python.org/> (дата обращения: 22.03.2020). – Текст : электронный.

6. Предварительная спецификация C# 6.0 : сайт. – 2020 - . - URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/language-specification/introduction> (дата обращения: 22.03.2020). – Текст : электронный.

7. Руководство по языку F# : сайт. – 2020 - . - URL: <https://docs.microsoft.com/ru-ru/dotnet/fsharp/language-reference/> (дата обращения: 22.03.2020). – Текст : электронный.

8. Visual Studio : сайт. – 2024 - . - URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 22.08.2024). – Текст : электронный.

9. Android Studio : сайт. – 2024 - . - URL: <https://developer.android.com/studio/intro> (дата обращения: 22.08.2024). – Текст : электронный.

10. IntelliJ IDEA : сайт. – 2000 - . - URL: <https://www.jetbrains.com/idea/> (дата обращения: 22.08.2024). – Текст : электронный.

### **Литература для оформления курсовой работы**

1. Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный.

2. ГОСТ 19.701-90 (ИСО 5807-85) Единая система программной документации (ЕСПД). Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения : межгосударственный стандарт : издание официальное : введен впервые : дата введения 1992-01-01 / Москва

Стандартинформ, 2010 – 158 с. – Текст: непосредственный.

## **СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И СПРАВОЧНЫЕ СИСТЕМЫ**

CITForum.ru : on-line библиотека свободно доступных материалов по информационным технологиям на русском языке : сайт. – 2001 – URL: <http://citforum.ru> (дата обращения: 22.08.2024). – Текст: электронный.

eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . – URL: <http://www.elibrary.ru> (дата обращения: 22.08.2024). – Режим доступа: для зарегистрир. пользвателей. – Текст: электронный.

Единое окно доступа к образовательным ресурсам : сайт. – Москва, 2005 - . – URL: <http://window.edu.ru/> (дата обращения: 22.08.2024). –Текст: электронный.

# ПРИЛОЖЕНИЕ 1. Титульный лист

## МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Кузбасский гуманитарно-педагогический институт  
федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Кемеровский государственный университет»

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

**Иванов Иван Иванович**  
гр. ПМИБ-23-1

### РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА

Курсовая работа  
по дисциплине «Языки и методы программирования»

по направлению подготовки 01.03.02 Прикладная математика и информатика  
направленность (профиль) подготовки «ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ»

Проверил:  
старший преподаватель  
Ю.С. Гаврилова

Общий балл: \_\_\_\_\_

Оценка: \_\_\_\_\_

\_\_\_\_\_ подпись

« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Новокузнецк 20 \_\_\_\_\_

## ПРИЛОЖЕНИЕ 2. Задание на курсовую работу

Кузбасский гуманитарно-педагогический институт  
федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Кемеровский государственный университет»

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

УТВЕРЖДАЮ

заведующий кафедрой

\_\_\_\_\_ Е.В. Решетникова

« \_\_\_\_ » \_\_\_\_\_

### ЗАДАНИЕ

на курсовую работу

студенту гр. ПМИб-23-1 **Иванову Ивану Ивановичу**

**1. Тема курсовой работы** «**Разработка графического редактора**» утверждена распоряжением декана \_\_\_\_\_.

**2. Исходные данные:** печатные и электронные учебные издания, ресурсы сети интернет.

**Содержание работы (перечень подлежащих разработке вопросов):**

Подобрать и проанализировать научную литературу по теме «**Разработка графического редактора**»; исследовать технологии программирования, языки программирования и среды разработки, оценить возможность их применения для решения поставленной задачи; разработать программное приложение с использованием выбранных информационных технологий.

**3. Задание выдано:** « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**4. Срок сдачи курсовой работы:** « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

Руководитель старший преподаватель \_\_\_\_\_ Ю.С. Гаврилова

Студент гр. ПМИб-23-1 \_\_\_\_\_ **И.И. Иванов**

## **ПРИЛОЖЕНИЕ 3. Образец введения курсовой работы**

**Тема: Разработка программного приложения для расчета объема фигуры вращения, заданной произвольной кривой**

### **Введение**

#### **Актуальность исследования**

Сегодня в мире информационных технологий широко распространены задачи 3D-моделирования объектов. Трехмерная графика считается незаменимым средством для демонстрации разного рода сложных технических узлов, многоступенчатых производств, архитектурных сооружений. Трехмерные модели наглядно показывают все особенности строения объекта, его мельчайшие элементы, которые скрыты от глаз наблюдателя. Также достаточно важной является задача расчета различных характеристик построенных моделей и дальнейшее их проецирование на оригинал. На основании вышеизложенного можно считать актуальной задачу разработки приложения, которое позволит осуществлять моделирование фигур вращения, заданных произвольной кривой, расчет их объема и площади фигуры под графиком.

**Цель исследования:** разработать программный продукт, моделирующего фигуры вращения произвольной формы, с последующим расчетом их объема и площади фигуры под графиком.

#### **Задачи исследования:**

1. анализ предметной области решения задачи;
2. выбор оптимальной технологии программирования, языка программирования и среды разработки для решения задачи;
3. построение алгоритма решения задачи;
4. разработка программного приложения;
5. тестирование разработанного программного приложения.

**Теоретическая значимость исследования** заключается в систематизации методов, применяемых при решении задачи расчета объема фигуры вращения, заданной произвольной кривой.

**Практическая значимость исследования** заключается в том, что выводы и результаты курсовой работы (формулы, алгоритм и программное приложение) могут быть использованы на практике при решении аналогичных задач.

#### **ПРИЛОЖЕНИЕ 4. Примерные темы курсовых работ по дисциплине «Практикум по программированию»**

1. Разработка графического редактора.
2. Разработка редактора UML-диаграмм.
3. Разработка приложения моделирования и визуализации физических законов.
4. Разработка приложения «Автовокзал».
5. Разработка приложения моделирования движения транспортных средств на перекрестке.
6. Разработка приложения автоматизированного раскроя материала.
7. Разработка приложения «Библиотека».
8. Разработка приложения заказа такси.
9. Разработка приложения «Гостиница».
10. Разработка приложения для автостоянки.
11. Разработка приложения регистрации участников соревнования.
12. Разработка приложения для сервисного центра.
13. Разработка приложения «Планировщик».
14. Разработка приложения для автосервиса.
15. Разработка файлового менеджера.
16. Разработка приложения, моделирующего функционирование замкнутой биологической системы.
17. Разработка приложения для автобусного парка.
18. Разработка приложения моделирования планетарной системы.
19. Разработка приложения для вычисления площади произвольного многоугольника.
20. Разработка приложения моделирования муравейника.
21. Разработка компьютерной версии игры «Монополия».
22. Разработка приложения проведения on-line аукционов.
23. Разработка приложения заказа товаров.
24. Разработка приложения «Портфолио студента».

25. Разработка приложения «Туристическое агентство».
26. Разработка приложения моделирования химических процессов.
27. Разработка приложения формирования и отображения 3D-объектов.
28. Разработка приложения «Музей».
29. Разработка приложения «Зоопарк».
30. Разработка приложения моделирования улья.