

Министерство науки и высшего образования
Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Кузбасский государственный технический университет
имени Т. Ф. Горбачева»

Институт профессионального образования
Кафедра информатики и информационных систем

Составитель О. С. Семенова

УЧЕБНАЯ ПРАКТИКА

Методические материалы

Рекомендовано цикловой методической комиссией
Информационных систем и программирования
в качестве электронного издания
для использования в образовательном процессе

Кемерово 2026

Рецензенты: Асанов С. А., старший преподаватель кафедры информационных и автоматизированных производственных систем ФГБОУ ВО «Кузбасский государственный технический университет имени Т. Ф. Горбачева»
Семенов Ю. Н., доцент кафедры эксплуатации автомобилей ФГБОУ ВО «Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Семенова Ольга Сергеевна

Учебная практика : методические материалы для обучающихся специальности СПО 09.02.11 «Разработка и управление программным обеспечением» / Кузбасский государственный технический университет имени Т. Ф. Горбачева, Институт профессионального образования, Кафедра информатики и информационных систем ; составитель О. С. Семенова. – Кемерово : КузГТУ, 2026. – 1 файл (528 КБ). – Текст : электронный.

Методические указания по учебной практике описывают содержание, задания, перечень работ и вопросов на защиту следующих практик: УП.01.01 Учебная практика (Разработка, администрирование и защита баз данных), УП.02.01 Учебная практика (Разработка и интеграция модулей программного обеспечения), УП.03.01 Учебная практика (Проектирование и разработка информационных систем).

© Кузбасский государственный
технический университет
имени Т. Ф. Горбачева, 2026
© Семенова О. С.,
составление, 2026

ОГЛАВЛЕНИЕ

Введение	3
Общие положения.....	3
УП.01.01 Учебная практика (Разработка, администрирование и защита баз данных).....	6
УП.02.01 Учебная практика (Разработка и интеграция модулей программного обеспечения)	17
УП.03.01 Учебная практика (Проектирование и разработка информационных систем).....	30
Литература.....	35
Приложения.....	36

Введение

Первоначальные профессиональные навыки обучающиеся по основным профессиональным образовательным программам получают во время прохождения учебных и производственных практик. Практика имеет целью комплексное освоение обучающимися всех видов профессиональной деятельности по специальности (профессии) среднего профессионального образования (СПО), формирование общих и профессиональных компетенций, а также приобретение необходимых умений и опыта практической работы по специальности (профессии).

Практика по профилю специальности направлена на формирование у обучающегося общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках профессиональных модулей ОП СПО по каждому из видов профессиональной деятельности, предусмотренных ФГОС СПО по специальности.

Общие положения

Учебная практика проводится в КузГТУ в специально оборудованных аудиториях. Организацию и руководство практикой по профилю специальности (профессии) осуществляют руководители практики из числа профессорско-преподавательского состава КузГТУ.

Руководитель практики из числа лиц, относящихся к профессорско-преподавательскому составу университета:

- разрабатывает содержание и планируемые результаты практики;
- осуществляет руководство практикой;
- контролирует реализацию программы практики и условия проведения практики, в том числе требования охраны труда, безопасности жизнедеятельности и пожарной безопасности в соответствии с правилами и нормами, в том числе отраслевыми;
- формирует группы в случае применения групповых форм проведения практики;

- определяет процедуру оценки общих и профессиональных компетенций обучающегося, освоенных им в ходе прохождения практики;

- разрабатывает формы отчетности и оценочный материал прохождения практики.

Обучающиеся в период прохождения практики обязаны:

- выполнять задания, предусмотренные программами практики;

- соблюдать действующие в КузГТУ правила внутреннего трудового распорядка;

- соблюдать требования охраны труда и пожарной безопасности.

Результаты практики определяются программами практики, разрабатываемыми руководителями практики из числа профессорско-преподавательского состава КузГТУ. По результатам практики руководителем практики формируется аттестационный лист, содержащий сведения об уровне освоения обучающимся профессиональных компетенций, а также характеристика на обучающегося по освоению профессиональных компетенций в период прохождения практики.

В период прохождения практики обучающимся ведется дневник практики. По результатам практики обучающимся составляется отчет. В качестве приложения к дневнику практики обучающийся оформляет графические, аудио-, фото-, видеоматериалы, подтверждающие практический опыт, полученный на практике.

Аттестация по итогам учебной практики проводится с учетом (или на основании) результатов ее прохождения, подтверждаемых аттестационным листом.

Отчет по практике является основным документом, характеризующим работу обучающегося во время практики. Отчет составляется в соответствии с программой практики и содержит следующие разделы:

1. Титульный лист
2. Задание на учебную практику
3. Введение
4. Теоретические основы в соответствии с темами практики

5. Реализация поставленной задачи

6. Выводы

7. Список литературы

Комплект документов, оформляемых при прохождении учебной практики, а также титульный лист отчета по учебной практике приведены в Приложении А–И.

УП.01.01 Учебная практика (Разработка, администрирование и защита баз данных)

Целью учебной практики является комплексное освоение обучающимися основного вида профессиональной деятельности «Разработка, администрирование и защита баз данных», закрепление теоретических знаний, полученных при изучении профессионального модуля ПМ.01 «Разработка, администрирование и защита баз данных».

Задачи, решаемые в процессе прохождения учебной практики: проектирование структуры учебной базы данных; обработка данных с использованием языка запросов; оптимизация запросов для улучшения производительности; работа с транзакциями; написание хранимых процедур, функций и триггеров; установка и настройка системы управления базами данных; управление пользователями и правами доступа; настройка резервного копирования и восстановления базы данных; мониторинг производительности и настройка параметров производительности; обновление и документирование.

УП.01.01 Учебная практика (Разработка, администрирование и защита баз данных) включает следующие темы:

Тема 1. Работа с SQL и NoSQL базами данных.

Тема 2. Администрирование баз данных.

Задание на учебную практику

Заданием на учебную практику учебного модуля «Разработка, администрирование и защита баз данных» является рассмотрение вопросов, связанных с разработкой, администрированием и защитой баз данных.

Типовые задания на учебную практику:

Тема 1.1. Проектирование структуры учебной базы данных

Задание 1 (для реляционных систем управления базами данных (СУБД)). Выберите одну из предложенных предметных областей (или предложите свою, согласовав с преподавателем):

- онлайн-библиотека;
- интернет-магазин;

- система учета студентов и экзаменов;
- сервис бронирования отелей.

Проведите анализ выбранной области: выделите основные сущности (не менее 4–5), для каждой сущности определите атрибуты (включая потенциальные ключи), идентифицируйте связи между сущностями (укажите тип связи: 1:1, 1:N, N:M) и их семантику (например, «студент сдает экзамен», «заказ включает товары»).

Разработайте ER-диаграмму базы данных, приведите полученные отношения как минимум к 3-й нормальной форме.

Задание 2 (для реляционных СУБД). Создайте схему базы данных в выбранной СУБД. (SQLite, PostgreSQL). Определите таблицы, их поля и связи между таблицами. Выберите корректные типы данных для каждого поля, задайте первичные и внешние ключи, выберите стратегию обеспечения ссылочной целостности для каждой связи.

Задание 3 (для реляционных СУБД). Используя инструменты генерации данных (например, ChatGPT, Hugging Face, Moscaroo, Faker библиотеку для Python/JavaScript), сгенерируйте реалистичные тестовые данные для всех таблиц вашей БД.

Количество записей в родительских таблицах должно быть не менее 100 штук, в дочерних таблицах – не менее 500.

Заполните таблицы сгенерированными данными. Для заполнения можно использовать SQL-запросы `INSERT INTO ... VALUES ...` или механизмы массового импорта (например, `COPY` в PostgreSQL). Убедитесь, что данные соответствуют типам полей и ограничениям.

Задание 4 (для нереляционных СУБД). Выберите одну из предметных областей (или согласуйте свою):

- блог-платформа;
- каталог товаров;
- система учета задач;
- социальная сеть.

Проведите анализ предметной области: выделите основные сущности (не менее 4), определите атрибуты каждой сущности, выявите типовые сценарии использования (например: «показать

пост с комментариями», «найти товары по категории и цене», «отобразить ленту пользователя»).

Спроектируйте документоориентированную модель для нереляционной СУБД (MongoDB): определите и обоснуйте, какие данные встраивать в документ, а какие хранить отдельно (ссылки) с учетом требований к производительности. Создайте примеры документов для каждой коллекции (покажите структуру документа с полями и вложенными объектами/массивами, продемонстрируйте связи между документами).

Задание 5 (для нереляционных СУБД). Создайте базу данных и коллекции в MongoDB. Для каждой коллекции определите структуру документов (поля, типы, вложенные объекты/массивы), задайте валидацию схем (используйте validator при создании коллекции).

Задание 6. Сгенерируйте реалистичные тестовые данные для всех коллекций. Заполните коллекции используя insertMany() для массовой вставки данных.

Тема 1.2. Обработка данных с использованием языка запросов

Задание 1 (для реляционных СУБД). Изучите диалект языка запросов для выбранной СУБД. Изучите отличия языка запросов для выбранной СУБД от стандарта SQL:2023:

- Изучите основные операторы SELECT, WHERE, ORDER BY для извлечения данных из одной таблицы.
- Изучите возможность использования операции соединения таблиц с помощью операторов INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN для получения данных из нескольких таблиц.
- Изучите агрегатные функции, такие как SUM, COUNT, AVG, MIN, MAX для вычисления соответствующего значения для группы записей, изучите оператор HAVING для фильтрации агрегированных данных.
- Рассмотрите возможность использования подзапросов для создания более сложных и гибких запросов.

Задание 2 (для реляционных СУБД). Определите, какие запросы к данным БД необходимы. Создайте соответствующие за-

просы для выборки, обновления, удаления и добавления данных в таблицы.

Задание 3 (для реляционных СУБД). Создайте покрывающие запросы индексы для ускорения поиска и обработки данных.

Задание 4 (для реляционных СУБД). Создайте отчеты на основе выполненных запросов.

Задание 5 (для нереляционных СУБД). Изучите базовый синтаксис языка запросов для СУБД MongoDB. Изучите основные операторы для выборки данных, освоите фильтрацию и сортировку, изучите агрегационный фреймворк (aggregate), рассмотрите встроенные механизмы замены подзапросов.

Задание 6 (для нереляционных СУБД). Изучите синтаксис для формирования запросов для всех основных операций с данными.

- Создайте запросы на выборку (find), например, получите все документы из коллекции, выберите документы по простому условию, сформулируйте сложный фильтр.

- Напишите запросы на обновление (updateOne, updateMany), например, измените одно поле в одном документе, обновите несколько документов по условию, добавьте элемент в массив (\$push), удалите его (\$pull), обновите по условию (\$set с arrayFilters).

- Сформируйте запросы на удаление (deleteOne, deleteMany), например, удалите один документ по _id, удалите несколько документов по условию.

- Реализуйте операцию добавления данных (insertOne, insertMany), например, добавьте один новый документ, массово вставьте несколько документов.

- Создайте агрегационные запросы для аналитических задач, например, подсчитайте количество документов по группам, найдите средние, минимальные, максимальные значения, соедините данные из двух коллекций с использованием \$lookup.

Тема 1.3. Оптимизация запросов для улучшения производительности

Задание 1 (для реляционных СУБД). Проведите анализ плана выполнения запроса. Для этого выберите 3–4 типовых запроса к вашей БД (например, выборка записей по условию, соединение

2–3 таблиц, агрегация с GROUP BY и HAVING, подзапрос в WHERE или SELECT). Получите план выполнения для каждого запроса, используя EXPLAIN ANALYZE в PostgreSQL <запрос> или EXPLAIN QUERY PLAN <запрос> в SQLite. Найдите операции Sequential Scan, Index Scan, Nested Loop, Hash Join, Sort. Оцените стоимость запроса, количество обрабатываемых строк, время выполнения. Определите, где происходят дорогие операции (например, полное сканирование таблицы). Сделайте выводы, ответив на вопросы: «какой запрос самый медленный и почему?», «какие операции доминируют в плане?».

Задание 2 (для реляционных СУБД). Создайте и протестируйте индексы. На основе анализа, выполненного в предыдущем задании выберите 2–3 поля, которые часто используются в условиях WHERE, соединениях JOIN, сортировке ORDER BY, группировке GROUP BY. Создайте некластеризованные индексы с помощью инструкции CREATE INDEX idx_name ON table(column).

Перезапустите планы выполнения для запросов из Задания 1 и сравните как изменился план (например, появились инструкции Index Scan вместо Sequential Scan), насколько сократилось время выполнения, уменьшилось ли количество просматриваемых строк.

Проверьте избыточность индексов. Для этого используйте запросы к системным таблицам (например, pg_stat_user_indexes в PostgreSQL) для поиска неиспользуемых индексов.

Задание 3 (для реляционных СУБД). Проведите оптимизацию запросов с JOIN и с подзапросами. Проверьте, есть ли индексы на полях соединения в запросе с JOIN. Сравните, есть ли разница в стоимости плана выполнения запросов с INNER JOIN и LEFT JOIN. Перепишите подзапросы по примеру:

- было: WHERE id IN (SELECT product_id FROM order_items WHERE quantity > 10);
- стало: JOIN order_items ON products.id = order_items.product_id WHERE order_items.quantity > 10;
- или: WHERE EXISTS (SELECT 1 FROM order_items WHERE product_id = products.id AND quantity > 10).

Сравните планы и время выполнения для всех вариантов.

Создайте материализованное представление VIEW для сложного запроса. Сравните скорость выборки из представления и исходного запроса.

Задание 4 (для реляционных СУБД). Проведите оптимизацию запросов с агрегацией и сортировкой. Проанализируйте план выполнения запроса: определите, какая операция используется для группировки (Hash Aggregate, Group Aggregate), и сколько строк обрабатывается в операциях запроса. Создайте и протестируйте покрывающий запрос индекс.

Задание 5 (для нереляционных СУБД). Проведите анализ плана выполнения запроса в СУБД MongoDB с помощью инструкции explain(). Для этого выберите 3–4 типовых запроса к БД, например, осуществите выборку документов по простому условию (find({ status: "active" })), выполните запрос с сортировкой (sort({ createdAt: -1 })) и запрос на соединение коллекций с помощью инструкции \$lookup ().

Получите план выполнения для каждого запроса с помощью explain("executionStats"). Проанализируйте план. Для этого найдите поле winningPlan (выбранный оптимизатором план), проверьте план выполнения каждого шага (COLLSCAN – полное сканирование, IXSCAN – сканирование индекса, FETCH – извлечение документов, SORT – сортировка в памяти/на диске). Оцените следующие метрики:

- executionTimeMillis – общее время выполнения (мс);
- totalKeysExamined – сколько ключей индекса было просмотрено;
- totalDocsExamined – сколько документов было просмотрено;
- nReturned – сколько документов возвращено.

Сделайте выводы, какой запрос самый медленный и почему, на каком шаге происходит полное сканирование и можно ли его заменить на сканирование индекса, насколько эффективно используется индекс (соотношение totalKeysExamined / nReturned).

Задание 5 (для нереляционных СУБД). Создайте и протестируйте индексы на основе анализа, выполненного в предыдущем задании. Для этого выберите поля, которые часто использу-

ются в условиях `find()`, сортировке `sort()`, агрегационных операциях `$match`, `$group`, `$sort`.

Создайте индексы с помощью инструкции `createIndex()`. Перезапустите `explain("executionStats")` для тех же запросов и сравните, появилась ли операция сканирования индекса вместо операции полного сканирования, как изменилось `executionTimeMillis`, уменьшилось ли `totalDocsExamined`.

Тема 1.4. Работа с транзакциями

Задание 1 (для реляционных СУБД). Создайте скрипт или хранимую процедуру, инкапсулирующую бизнес-транзакцию. Транзакция должна включать несколько последовательных операций `INSERT`, `UPDATE` и/или `DELETE` над разными таблицами. Продемонстрируйте два сценария выполнения:

а) Успешное выполнение с фиксацией изменений (`COMMIT`). Проверить согласованность данных после завершения.

б) Откат транзакции (`ROLLBACK`) при имитации сбоя или нарушении бизнес-правила. Убедиться, что изменения не применены.

Задание 2 (для реляционных СУБД). Исследуйте уровни изоляции и конкурентного доступа. Создайте сценарий, моделирующий параллельный доступ к одним и тем же данным из двух независимых сессий. Продемонстрируйте и объясните возможные аномалии параллельного доступа (грязное, неповторяющееся, фантомное чтение) в зависимости от установленного уровня изоляции.

Задание 3 (для реляционных СУБД). Проведите обработку ошибок внутри транзакции для обеспечения ее отката при возникновении исключительных ситуаций. Изучите и примените на практике механизм точек сохранения (`SAVEPOINT`) для реализации частичного отката внутри сложной транзакции. Смоделируйте и разрешите ситуацию взаимной блокировки (`deadlock`). Определите, как СУБД обнаруживает и устраняет такие ситуации.

Задание 4 (для реляционных СУБД). Используйте системные представления или команды СУБД для мониторинга текущих транзакций, их статуса и блокировок. Настройте параметры СУБД, связанные с завершением долгих или «зависших» тран-

закций. Объясните роль журнала транзакций в обеспечении свойства устойчивости (Durability).

Задание 5 (для нереляционных СУБД). Спроектируйте коллекцию orders так, чтобы весь заказ и все его позиции (items) хранились в одном документе (денормализация). Реализуйте операцию «добавление товара в заказ». Используйте одну атомарную операцию updateOne() с \$inc (для увеличения счетчика) и \$push (для добавления элемента в массив items), чтобы гарантировать, что обе части операции выполняются как единое целое. Продемонстрируйте два сценария:

а) Успешное выполнение. Проверьте согласованность документа.

б) Имитируйте сбой (например, нарушение схемы валидации). Убедитесь, что документ не был изменен.

Для версий MongoDB 4.0+: Повторите задание, но с использованием много-документных транзакций (multi-document ACID transactions), если заказ и товары хранятся в разных коллекциях. Объясните затраты на реализацию и ограничения этого подхода.

Тема 1.5. Написание хранимых процедур, функций и триггеров

Задание 1 (для реляционных СУБД). Создайте хранимую процедуру или функцию, реализующую сложную бизнес-логику, требующую выполнения последовательных операций с данными. Процедура должна включать обработку входных параметров, условные конструкции (IF/ELSE, CASE), циклы и возвращать результат.

Задание 2 (для реляционных СУБД). Создайте набор триггеров разного типа (BEFORE/AFTER, INSERT/UPDATE/DELETE), обеспечивающих автоматическое поддержание целостности данных и аудит изменений:

а) Реализуйте триггер для проверки бизнес-правила при изменении данных (например, запрет уменьшения цены товара более чем на 20 %);

б) Создайте триггер для ведения журнала аудита, фиксирующего все изменения указанной таблицы с указанием пользователя, времени и типа операции;

в) Реализуйте каскадное обновление связанных данных через триггеры (например, пересчет общей суммы заказа при изменении состава позиций).

Задание 3 (для нереляционных СУБД). Создайте JavaScript-функцию, реализующую сложную бизнес-логику обработки заказа клиента с последовательным выполнением операций. Функция должна принимать входные параметры, содержать условные конструкции и циклы, а также возвращать структурированный результат.

Логика функции должна включать: 1) проверку существования клиента по идентификатору клиента `customerId` в коллекции `customers`; 2) для каждого элемента в `items` – поиск товара в коллекции `products`, проверку наличия на складе (`stock >= quantity`) и расчет стоимости с учетом возможной скидки по `promoCode`; 3) применение общей скидки по промокоду (если он валиден и не истек), но с ограничением: итоговая цена не может быть ниже 30 % от исходной суммы; 4) формирование документа заказа в коллекции `orders` со статусом `pending` и всеми рассчитанными суммами; 5) обновление остатков на складе в коллекции `products` (уменьшение `stock` на заказанное количество); 6) начисление бонусных баллов клиенту (1 балл за каждые потраченные 100 рублей). Функция должна возвращать объект следующего формата: { `success: boolean`, `orderId: string | null`, `totalAmount: number`, `discountAmount: number`, `bonusPointsEarned: number`, `errors: string[]` }. Сохраните функцию в системе MongoDB через коллекцию `system.js` с идентификатором `processOnlineOrder`, затем вызовите ее с тестовыми данными через `db.eval()` или агрегационный конвейер с оператором `$function`. Убедитесь, что все операции выполняются атомарно в рамках транзакции (используйте `db.getMongo().startSession()` и `withTransaction`), а при ошибке транзакция откатывается.

Тема 2.1. Установка и настройка системы управления базами данных

Задание 1. Изучите основные типы серверов баз данных и их применение.

Задание 2. Подготовьте сервер для установки базы данных: выберите аппаратное и программное обеспечение, установите необходимое программное обеспечение (ПО).

Задание 3. Установите и настройте выбранный сервер баз данных (например, PostgreSQL, MongoDB).

Тема 2.2. Управление пользователями и правами доступа

Задание 1 (для реляционных СУБД). Создайте новых пользователей базы данных (БД). Сопоставьте пользователя БД с именем входа «ФИО». Создайте новую роль БД «РольФИО». Включите созданного пользователя БД в новую роль БД «РольФИО». Предоставьте созданной роли БД разрешения на объекты БД (например, в SQLServer на одну из таблиц БД предоставьте разрешения SELECT, на другую таблицу – INSERT, UPDATE, DELETE).

Задание 2 (для нереляционных СУБД). Создайте в БД TestDB пользователя user_FIO с паролем. Создайте в этой же БД роль role_FIO. Назначьте роли role_FIO права: find на коллекцию products; insert, update, delete на коллекцию orders. Назначьте пользователю user_FIO роль role_FIO.

Тема 2.3. Настройка резервного копирования и восстановления базы данных

Задание 1. Создайте и настройте задания для автоматического обслуживания базы данных: настройте автоматический запуск процедуры резервного копирования базы данных, перестроение индексов и т. д. Проведите тестирование процедуры восстановления данных из резервной копии.

Задание 2. Проведите импорт данных из внешних источников и экспорт данных из учебной базы данных.

Тема 2.4. Мониторинг производительности и настройка параметров производительности

Задание 1. Проведите оптимизацию запросов к учебной базе данных. Для этого идентифицируйте медленные запросы и создайте покрывающие запрос индексы.

Тема 2.5. Обновление и документирование

Задание 1. Требуется составить описание структуры каждой таблицы/коллекции: список полей с типами данных; первичные и внешние ключи (если есть); индексы; примеры документов/строк.

Типовые вопросы на зачет

1. Особенности реализации поставленной задачи.
2. Что является целью выполнения задания по практике?

3. Поясните схему реализованной базы данных.
4. Особенности, выявленные при анализе предметной области.
5. Сформулируйте требования к базе данных.
6. Концептуальная модель базы данных.
7. Инфологическая модель базы данных.
8. Дatalogическое проектирование базы данных.
9. Определение информационных единиц и связей между ними.
10. Преобразование исходной инфологической модели в модель данных, поддерживаемую конкретной СУБД.
11. Проверка адекватности полученной дatalogической модели предметной области.
12. Сценарии работы с данными.
13. Архитектуры системы.
14. Технические требования к серверу баз данных.
15. Требования к корпоративной сети.
16. Серверные компоненты базы данных.
17. Клиентские компоненты базы данных.
18. Установка и настройка сервера баз данных.
19. Создание механизмов сервера для обслуживания базы данных.
20. Установка и развертывание системы.
21. Работа с журналом аудита базы данных.
22. Мониторинг нагрузки сервера.
23. Настройка политики безопасности для базы данных.
24. Создание резервных копий базы данных.
25. Восстановление базы данных.
26. Мониторинг активности портов.
27. Оформление требований и разработка технического задания по сертификации информационной системы (базы данных).
28. Выбор сертификатов.
29. Сроки действия сертификатов.

УП.02.01 Учебная практика (Разработка и интеграция модулей программного обеспечения)

Целью учебной практики является комплексное освоение обучающимися основного вида профессиональной деятельности «Разработка и интеграция модулей программного обеспечения», закрепление теоретических знаний, полученных при изучении профессионального модуля ПМ.02 «Разработка и интеграция модулей программного обеспечения».

Исходя из целей, перед обучающимися ставятся следующие задачи: формирование общих и профессиональных компетенций; приобретение опыта профессиональной деятельности и самостоятельной работы.

УП.02.01 Учебная практика (Разработка и интеграция модулей программного обеспечения) включает следующие темы:

Тема 1. Проектирование программных модулей.

Тема 2. Создание и интеграция программных продуктов в целевую информационную систему.

Тема 3. Отладка программных продуктов.

Тема 4. Создание технической документации.

Задание на учебную практику

Заданием на учебную практику является проектирование, моделирование, тестирование и внедрение модулей прикладного программного обеспечения. При выполнении задания необходимо определить этапы разработки программного обеспечения; выбрать технологию разработки исходного модуля исходя из его назначения; разработать программный модуль; произвести интеграцию программного модуля в информационную систему; отладить программный модуль, составить набор тестов и провести его тестирование.

Примеры типовых заданий на практику:

Тема 1.1. Проектирование модулей программного обеспечения с учетом технического задания

Задание 1. Выберите существующий проект информационной системы (например, из репозиторий GitHub, GitLab или аналогичных платформ) либо используйте собственный ранее разра-

ботанный проект. Изучите его архитектурную документацию: описание общей архитектуры (схемы, диаграммы); спецификации интерфейсов и API; описания модулей и их взаимодействия; требования к производительности и масштабируемости.

Задание 2. Составьте краткий обзор (1–2 страницы) с указанием: назначения системы; ключевых технологических решений (языки, фреймворки, СУБД); основных архитектурных паттернов (монолит, микросервисы, клиент-сервер и т. п.).

Задание 3. На основе изученной документации выделите 2–3 существующих модуля системы, между которыми требуется улучшить интеграцию (например, модуль аутентификации и модуль обработки заказов). Для каждого модуля опишите его функциональное назначение и интерфейсы; укажите текущие точки интеграции (API-эндпоинты, очереди сообщений, базы данных); перечислите проблемы или ограничения в текущем взаимодействии (например, задержки, дублирование данных, отсутствие транзакционности). Сформулируйте требования к новому интеграционному механизму (например, использование брокера сообщений, синхронный/асинхронный API, кэширование), обосновав выбор.

Тема 1.2. Визуализации и описания архитектурных решений

Задание 1. Разработайте и визуализируйте архитектурное решение для улучшения интеграции между выбранными модулями. Для этого создайте UML-диаграмму компонентов, отражающую новые и существующие модули, их интерфейсы и связи; постройте диаграмму последовательности для ключевого сценария взаимодействия (например, «создание заказа с проверкой прав пользователя»); опишите протокол обмена данными (формат сообщений, методы передачи, механизмы обработки ошибок); укажите, как решение обеспечивает отказоустойчивость и масштабируемость.

Оформите результат в виде раздела архитектурной документации (2–3 страницы), включающего цель и область применения решения; схемы и диаграммы; краткое описание алгоритмов взаимодействия; список используемых технологий и их обоснование.

Тема 1.3. Определение интерфейсов и взаимодействия модулей в системе

Задание 1. В рассматриваемой информационной системе (ИС) определите список основных программных модулей/сервисов; способы их взаимодействия (API, сообщения, общие базы данных); форматы данных, передаваемых между модулями.

Задание 2. Составьте таблицу «Модули и их интерфейсы» со столбцами: название модуля; назначение модуля; входные данные (формат, источник); выходные данные (формат, получатель); протокол взаимодействия (HTTP, MQTT, RPC и т. п.); примеры эндпоинтов/точек интеграции (URL, очереди, таблицы).

В конце добавьте краткий вывод (3–4 предложения) о том, насколько прозрачна и документирована интеграция между модулями в выбранной ИС.

Задание 3. Спроектируйте интерфейсы для двух новых модулей, которые должны быть интегрированы в систему – модуля импорта данных из внешних CSV-файлов и модуля отправки уведомлений по электронной почте. Для каждого модуля определите входные и выходные данные (с указанием типов и ограничений); опишите протокол взаимодействия с существующими модулями (например, REST API, очередь сообщений); приведите примеры запросов/ответов (в формате JSON или XML) для ключевых операций; укажите, как модуль обрабатывает ошибки.

Оформите результат в виде раздела «Спецификация интерфейсов» (1–2 страницы), включающего описание целей и области применения каждого модуля, таблицы с параметрами интерфейсов, примеры сообщений, схему взаимодействия модулей (упрощенную UML-диаграмму компонентов).

Тема 2.1. Создание модулей программного обеспечения

Задание 1. Разработайте базовый функционал модуля. Для этого выберите один из спроектированных ранее модулей (например, модуль импорта данных). Реализуйте основной функционал модуля согласно спецификации. Обеспечьте корректную обработку входных данных, реализуйте валидацию данных, добавьте логирование основных операций.

Задание 2. Реализуйте систему обработки исключений, добавьте механизмы восстановления после сбоя, внедрите уведом-

ления об ошибках. Проведите документирование всех возможных ошибок и способов их обработки.

Тема 2.2. Работа с API и веб-сервисами для взаимодействия между модулями

Задание 1. Разработайте спецификацию REST API для выбранного модуля, определите HTTP методы для каждой операции (GET, POST, PUT, DELETE), спроектируйте структуру URL-адресов, определите форматы запросов и ответов в JSON, опишите коды состояния HTTP и сценарии обработки ошибок, составьте документацию по API в формате OpenAPI/Swagger.

Задание 2. Спроектируйте API-шлюз для ИС: определите правила маршрутизации запросов, реализуйте механизмы аутентификации и авторизации, внедрите ограничение по частоте запросов, добавьте кэширование часто используемых данных, реализуйте агрегацию данных из нескольких сервисов

Задание 3. Спроектируйте систему обмена сообщениями между модулями: выберите технологию message broker (RabbitMQ, Kafka и т. п.), определите типы сообщений и их структуру, реализуйте producer-компонент и consumer-компонент, обеспечьте гарантированную доставку сообщений, внедрите обработку ошибок и повторных попыток.

Задание 4. Осуществите интеграцию с внешними сервисами: выберите реальный внешний API (например, КузГТУ), проанализируйте документацию внешнего API, спроектируйте адаптер для интеграции, реализуйте механизм аутентификации с внешним сервисом, обеспечьте обработку различных сценариев ответа, внедрите мониторинг доступности внешнего сервиса

Задание 5. Проведите тестирование API. Для этого разработайте набор тестовых сценариев для API, реализуйте автоматизированные тесты, проверьте корректность обработки различных типов запросов, протестируйте обработку ошибок и граничных случаев, оцените производительность API при нагрузке и составьте отчет о результатах тестирования.

Задание 6. Подготовьте техническую документацию по API: опишите все доступные эндпоинты и их параметры, приведите примеры запросов и ответов, задокументируйте механизмы аутентификации, опишите обработку ошибок и коды состояния,

подготовьте руководство для разработчиков по использованию API.

Тема 2.3. Работа с интеграционными платформами и инструментами

Задание 1. Проведите детальный анализ популярных интеграционных платформ (MuleSoft, Apache Camel, IBM Integration Bus). Требуется сравнить их функциональные возможности, определить оптимальные сценарии применения каждой платформы. Необходимо составить таблицу сравнения по следующим параметрам: поддержка протоколов, масштабируемость, инструменты разработки, стоимость владения, уровень поддержки сообщества.

Задание 2. Выполните установку и настройку выбранного брокера сообщений (Apache Kafka или RabbitMQ). Создайте необходимые топики и очереди для различных сценариев работы. Настройте параметры репликации и отказоустойчивости. Реализуйте систему мониторинга состояния брокера. Подготовьте подробную документацию процесса установки и настройки.

Задание 3. Выберите реальный бизнес-сценарий для интеграции (например, синхронизация данных между системами). Спроектируйте архитектуру решения. Реализуйте интеграционный поток, включающий определение источников данных, преобразование форматов, обработку ошибок, логирование. Проведите тестирование решения на тестовых данных.

Задание 4. Изучите принципы работы Enterprise Service Bus. Спроектируйте ESB-решение для существующей системы. Реализуйте основные компоненты: маршрутизацию сообщений, трансформацию данных, обработку исключений. Настройте системы мониторинга и логирования для обеспечения стабильной работы.

Задание 5. Установите и настройте платформу API Management. Создайте API-шлюз. Настройте механизмы аутентификации и авторизации, ограничения доступа, кэширования, мониторинга производительности.

Тема 3.1. Оценка объема тестирования программного обеспечения с целью определения необходимых ресурсов для его выполнения

Задание 1. Определите необходимое количество тестировщиков с учетом сложности проекта и объема тестируемых функций.

Задание 2. Используя методы декомпозиции задач (например, Work Breakdown Structure) и оценки трудозатрат (например, метод трехзначной оценки по PERT), рассчитайте общее время, необходимое для проведения всех видов тестирования (модульное, интеграционное, системное, приемочное) в рамках проекта. Исходные данные: спецификация требований и список тестовых сценариев. Проанализируйте, как полученная оценка влияет на сроки релиза и формирование календарного плана (графика тестирования).

Задание 3. На основе технической архитектуры проекта определите необходимый набор тестовых сред, оборудования, лицензий на ПО и инструментов для тестирования (системы управления тестами, инструменты автоматизации, средства виртуализации и т. д.). Составьте смету на развертывание и поддержание этой инфраструктуры. Оцените, какие элементы можно использовать из облачных сервисов (SaaS/PaaS/IaaS) для оптимизации затрат.

Тема 3.2. Подготовка тестовых платформ

Задание 1. На основе технического задания составьте матрицу необходимых тестовых платформ. Включите параметры: ОС, версии ОС, типы устройств (десктоп, планшет, смартфон), версии браузеров/мобильных ОС, разрешения экранов, наличие/отсутствие специфического оборудования (например, сканер отпечатков). Обоснуйте выбор наиболее критичных комбинаций для тестирования.

Задание 2. Для проекта из предыдущего задания сравните три стратегии:

- 1) Закупка и содержание собственного «железного» парка.
- 2) Использование облачных сервисов виртуализации (AWS, Azure, GCP).

3) Использование специализированных облачных платформ для мобильного и кроссплатформенного тестирования (BrowserStack, Sauce Labs, AWS Device Farm).

Составьте сравнительную таблицу по критериям: начальные затраты, операционные расходы, скорость развертывания, гибкость (масштабируемость), безопасность, доступность редких конфигураций.

Задание 3. Используя инструмент виртуализации (VirtualBox/VMware) или облачный образ (например, в GCP/Azure), разверните чистую виртуальную машину с ОС (например, Ubuntu Server LTS или Windows 10). С помощью скриптов автоматизируйте установку и базовую настройку: обновление системы, установку необходимого ПО (JDK, Node.js, Python, Git, Docker), настройку сети, создание стандартного пользователя для тестов. Создайте шаблон этой виртуальной машины. Опишите процесс и приложите скрипты.

Тема 3.3. Отладка программного обеспечения на уровне программных модулей

Задание 1. Изучите основные методы и инструменты отладки программного обеспечения. Освойте работу с отладчиком в выбранной среде разработки. Научитесь устанавливать точки останова и отслеживать выполнение кода. Изучите техники пошаговой отладки.

Задание 2. Проанализируйте различные типы ошибок в программном обеспечении, определите основные причины возникновения ошибок.

Задание 3. Освойте использование различных инструментов отладки – встроенные отладчики IDE, профилировщики кода, системы логирования, мониторы производительности.

Задание 4. Изучите особенности отладки многопоточных и асинхронных приложений. Освойте техники синхронизации потоков.

Задание 5. Проведите профилирование программного модуля. Определите узкие места в коде. Проанализируйте использование памяти и CPU. Оптимизируйте критические участки кода.

Задание 6. Научитесь диагностировать и исправлять ошибки в производственной среде. Освойте техники удаленного доступа к логам. Изучите методы сбора дампов памяти.

Тема 3.4. Формирование тестовых сценариев

Задание 1. Проведите детальный анализ функциональных и нефункциональных требований к системе, определите приоритетные области тестирования, составьте матрицу трассировки требований, идентифицируйте критические бизнес-процессы, требующие особого внимания при тестировании.

Задание 2. Сформулируйте стратегию тестирования с учетом особенностей проекта. Определите подходы к тестированию (ручное, автоматизированное) и выберите технику тестирования. Разработайте план тестирования с учетом временных и ресурсных ограничений.

Задание 3. Проанализируйте функциональные требования и разработайте тестовые сценарии.

Задание 4. Проанализируйте возможности автоматизации тестирования. Определите подходящие инструменты автоматизации. Рассчитайте ROI (возврат инвестиций) от автоматизации. Составьте план внедрения автоматизированного тестирования.

Задание 5. Составьте детальный график тестирования с учетом подготовки тестовых данных, выполнения тест-кейсов, исправления дефектов, повторного тестирования.

Тема 3.5. Формирование и представление отчетности о подготовке к выполнению задания на тестирование программного обеспечения в соответствии с установленными регламентами

Задание 1. Создайте универсальный чек-лист для подтверждения готовности к началу тестирования функционального блока. Чек-лист должен включать разделы: документация, артефакты, среда, ресурсы. Для каждого пункта укажите ответственного за подтверждение и статус. Представьте заполненный пример для гипотетического проекта.

Задание 2. Составьте отчет о приемке тестовой среды. Проведите аудит по чек-листу (версии ОС, СУБД, промежуточного ПО, открытые порты, доступ к логам, данные миграции). Зафиксируйте найденные расхождения с ожидаемой конфигурацией. Сформируйте официальный отчет о приемке, который включает:

- 1) Цель отчета.
- 2) Описание конфигурации (фактическая / ожидаемая).
- 3) Результаты smoke-теста.

4) Выявленные риски и ограничения.

5) Подпись о принятии ответственности за дальнейшую эксплуатацию.

Сымитируйте процесс обмена этим отчетом между командами.

Задание 3. На основе шаблона стандарта IEEE 829 или внутреннего регламента компании заполните ключевые разделы плана тестирования для небольшого проекта (например, «Добавление формы оплаты через SberPay»). Сосредоточьтесь на разделах, описывающих подготовку: «Объекты тестирования», «Функции для тестирования», «Подход/стратегия», «Критерии входа/выхода», «Окружение/потребности», «Расписание», «Риски и ограничения». Особое внимание уделите критериям входа (Entry Criteria) как формальному обоснованию готовности.

Тема 3.6. Выполнение тестовых процедур на тестовых данных

Задание 1. Разработайте детальное описание тестовых сценариев для каждого функционального блока программы. Опишите шаги, ожидаемые результаты и оценку прохождения теста для каждого сценария.

Например, тестовый сценарий для функционала «Регистрация нового пользователя» может выглядеть следующим образом:

Шаг 1: Открыть веб-приложение.

- Открыть веб-браузер.
- Ввести URL адрес веб-приложения.
- Нажать Enter.

Шаг 2: Перейти на страницу регистрации.

- Найти и нажать на кнопку «Регистрация».
- Дождаться загрузки страницы.

Шаг 3: Ввести данные для регистрации.

- Ввести уникальное имя пользователя в поле «Имя пользователя».
- Ввести пароль в поле «Пароль».
- Повторно ввести пароль для подтверждения в поле «Повторите пароль».
- Ввести адрес электронной почты в поле «Email».
- Нажать на кнопку «Зарегистрироваться».

Шаг 4: Проверить результат регистрации.

- Дождаться загрузки страницы с подтверждением регистрации.

- Проверить, что на странице отображается сообщение «Регистрация успешно завершена».

- Проверить, что новый пользователь появился в списке зарегистрированных пользователей в базе данных.

Шаг 5: Выйти из приложения.

- Найти и нажать на кнопку «Выход».

- Дождаться завершения выхода из приложения.

Задание 2. Подготовьте необходимые тестовые данные для проведения тестовых сценариев. Убедитесь, что тестовые данные покрывают различные сценарии использования и позволяют достичь полного покрытия тестирования.

Задание 3. Запустите разработанные тестовые сценарии для проверки работы программного обеспечения. Зафиксируйте результаты тестирования и проанализируйте полученные данные.

Тема 3.7. Тестирование программного обеспечения

Задание 1. Изучите основные инструменты отладки (отладчики, профилировщики).

Задание 2. Подготовьте входные параметры для проверки работы модуля в процессе отладки. Проведите отладку программного модуля пошагово, следя за выполнением кода и результатами выполнения каждой команды. Установите точки останова в нужных местах программы для контролируемой остановки исполнения, используйте отладчик для просмотра значений переменных, состояния памяти и анализа стека вызовов в процессе выполнения программы. Используйте полученную информацию из отладки для исправления ошибок в программном модуле.

Тема 4.1. Создание технической документации для модулей

Задание 1. Определите, какая документация должна сопровождать программный модуль, и создайте ее полную структуру (оглавление). В структуре должны присутствовать следующие разделы и документы:

1. Верхнеуровневый обзор.

2. Архитектурная документация (диаграммы, решения).
3. Руководство пользователя/интегратора.
4. Справочная документация.
5. Операционная документация.
6. Внутренняя документация (для разработчиков модуля).

Для каждого документа укажите целевую аудиторию, примерное содержание и формат (Markdown, PDF, Wiki-страница).

Задание 2. Проанализируйте жизненный цикл модуля: от идеи и проектирования до разработки, тестирования, выпуска и поддержки. Создайте диаграмму (например, схему процесса), на которой для каждого этапа жизненного цикла укажите:

- Какие документы создаются или изменяются.
- Кто является автором и ответственным за актуальность.
- С какими другими документами существует связь (зависимость, ссылка).
- В каком инструменте (системе) документ хранится.

Задание 3. Обменяйтесь документацией по модулю с другой командой. Проведите аудит полученной документации по чек-листу, оценив полноту, актуальность, ясность, доступность, единообразие стиля. На основе аудита создайте план улучшения.

Тема 4.2. Документирование кода, API и интерфейсов

Задание 1. Создайте внутренний стандарт компании для комментирования кода и описания API. Стандарт должен включать:

- для кода: правила оформления комментариев для модулей, классов, публичных методов, сложных приватных функций. Укажите обязательные и опциональные теги.
- для API: шаблон описания каждого endpoint (REST) или метода (GraphQL). Включите поля: краткое описание, параметры (входные/выходные), возможные ошибки, пример запроса/ответа.
- примеры и антипаттерны: приведите по 2–3 примера хорошего и плохого документирования для каждого случая.

Задание 2. Проведите рефакторинг и документирование «немного» интерфейса. Вам дан файл с кодом, реализующим некий программный интерфейс (например, класс с методами, файл с функциями). Код не содержит комментариев, а имена сущностей неочевидны. Ваша задача:

1. Проанализировать код и понять его логику.

2. Переименовать сущности (классы, методы, параметры) в соответствии с их фактической целью.

3. Добавить исчерпывающие комментарии в формате, выбранном в предыдущем задании.

4. Выделить и описать в отдельном документе ключевые паттерны использования этого интерфейса.

Задание 3. Для предоставленного API спецификации (например, в формате OpenAPI 3.0) разверните и настройте инструмент для создания интерактивной документации. Настройте его так, чтобы:

- была возможность отправлять реальные тестовые запросы к mock-серверу или указанному URL;
- документация была доступна онлайн по определенному адресу;
- примеры запросов были заполнены реалистичными данными;
- была реализована простая аутентификация для защищенных эндпоинтов (если требуется).

Тема 4.3. Работа со специализированным программным обеспечением по документированию программного кода

Задание 1. Настройте автоматическую генерацию документации в CI/CD пайплайне.

Задание 2. Проведите сравнительный анализ двух инструментов для проектирования и документирования API на выбор (например, Stoplight Studio и Apicurio Studio, или SwaggerHub и Postman). Создайте одну и ту же простую спецификацию API (3–4 эндпоинта) в обоих инструментах. Сравните их по критериям:

- удобство интерфейса и обучения;
- возможности совместной работы;
- поддержка различных стандартов (OpenAPI, AsyncAPI);
- интеграция с другими инструментами (IDE, CI/CD, тестирование);
- стоимость (включая бесплатный тариф).

Оформите вывод в виде таблицы с рекомендацией по использованию в разных сценариях.

Задание 3. Организуйте процесс сквозного документирования для проекта. Для этого:

1. Выберите и настройте статический генератор сайтов для технической документации (например, MkDocs, Docusaurus, VuePress).

2. Создайте структуру проекта: разделы для пользователей, разработчиков, API.

3. Настройте интеграцию: документация хранится в том же репозитории, что и код; изменения вносятся через Pull Request; собранный сайт развертывается автоматически.

4. Настройте плагин для автоматического включения сгенерированной API-документации в общий сайт. Предоставьте структуру репозитория, ключевые конфигурационные файлы и ссылку (или скриншоты) итогового сайта.

Типовые вопросы на зачет

1. Модели процесса разработки программного обеспечения.
2. Основные принципы процесса разработки программного обеспечения.
3. Основные подходы к интегрированию программных модулей.
4. Основы верификации и аттестации программного обеспечения.
5. Методы и способы идентификации сбоев и ошибок при интеграции приложений.
6. Кто занимается выявлением требований к ИС?
7. Перечислите этапы формулировки потребностей.
8. Что такое «требование к информационной системе»?
9. Кто занимается выявлением требований к ИС?
10. Перечислите этапы формулировки потребностей.
11. Что такое техническое задание?
12. Приведите основные команды git.
13. Как создать новую ветку в git?
14. Технологии моделирования предметной области.
15. Средства модульного тестирования.
16. Последовательность модульного тестирования.
17. Разработка группы тестов.
18. Приемы отладки ПО.
19. Способы нагрузочного тестирования.
20. Создание тестов в среде Microsoft Visual Studio.

УП.03.01 Учебная практика (Проектирование и разработка информационных систем)

Целью учебной практики является комплексное освоение обучающимися основного вида профессиональной деятельности «Проектирование и разработка информационных систем», закрепление теоретических знаний, полученных при изучении профессионального модуля ПМ.03 «Проектирование и разработка информационных систем».

Задачи: формирование у студента общих и профессиональных компетенций; приобретение практического опыта, реализуемого в рамках ОП СПО по основным видам профессиональной деятельности для последующего освоения ими общих и профессиональных компетенций по специальности 09.02.11 «Разработка и управление программным обеспечением»; проверка знаний, полученных при изучении модуля «Проектирование и разработка информационных систем».

УП.03.01 Учебная практика (Проектирование и разработка информационных систем) включает следующие темы:

Тема 1. Проектирование информационной системы для заданной предметной области.

Тема 2. Разработка информационной системы.

Тема 3. Тестирование информационной системы.

Тема 4. Разработка технической документации.

Тема 5. Оценка возможности модернизации информационной системы.

Задание на учебную практику

Задание на учебную практику включает в себя проектирование и разработку информационной системы для заданной предметной области в рамках выбранной технологии разработки ПО.

Типовые задания на учебную практику:

Тема 1.1. Изучение предметной области

Задание 1. Выберите предметную область для изучения. Изучите текущие бизнес-процессы, которые можно автоматизировать или оптимизировать в заданной предметной области.

Тема 1.2. Сбор исходных данных для разработки проектной документации

Задание 1. Выберите бизнес-процесс для автоматизации. Идентифицируйте основные потоки данных и взаимодействия между участниками выбранного бизнес-процесса. Для этого используйте моделирование потоков данных DFD и метод функционального моделирования SADT/IDEF0.

Задание 2. Проведите сбор функциональных и нефункциональных требований к ИС, отобразите их с помощью соответствующей диаграммы UML.

Тема 1.3. Разработка проектной документации

Задание 1. Разработайте общую структуру проектной документации, включая разделы, подразделы и их содержание.

Задание 2. Составьте техническое задание на разработку информационной системы.

Тема 2.1. Разработка модулей информационной системы

Задание 1. Опираясь на техническое задание, разработайте прототип интерфейса каждого модуля.

Задание 2. Опираясь на техническое задание, разработайте архитектурное решение для каждого модуля, определите его структуру, интерфейсы и взаимодействие с другими модулями.

Задание 3. Напишите код для каждого модуля.

Задание 4. Соедините разработанные модули в единую информационную систему, обеспечив совместимость и согласованность их работы.

Тема 2.2. Разработка подсистемы безопасности информационной системы

Задание 1. Ознакомьтесь с основными понятиями безопасности информационной системы. Найдите определения конфиденциальности, целостности, доступности, аутентификации и авторизации.

Задание 2. Исследуйте основные угрозы информационной безопасности, такие как взлом, вирусы, DoS-атаки.

Задание 3. Составьте план действий для создания подсистемы безопасности информационной системы. Рассмотрите различные методы и инструменты для обеспечения безопасности информационной системы, такие как аутентификация по паролю,

двухфакторная аутентификация, биометрическая аутентификация и т. д. Проанализируйте их преимущества и недостатки.

Задание 4. Проведите тестирование безопасности для проверки эффективности указанных мер безопасности.

Тема 3.1. Осуществление тестирования разработанной информационной системы

Задание 1. Определите цели тестирования ИС.

Задание 2. Разработайте сценарии тестирования, которые охватывают различные функциональные возможности информационной системы.

Задание 3. Проведите тестирование информационной системы. Составьте отчет о проведенном тестировании и исправлении ошибок для последующего анализа.

Тема 4.1. Разработка технической документации на эксплуатацию информационной системы

Задание 1. Выберите целевую аудиторию, которая будет использовать данную документацию (администраторы, конечные пользователи, технические специалисты и т. д.). Уточните цели создания документации: обучение пользователей, обеспечение поддержки, предоставление инструкций по эксплуатации и т. д.

Задание 2. Определите структуру документации. Обычно техническая документация включает в себя описание системы, инструкции по установке, настройке, использованию, администрированию, обслуживанию, справочную информацию и т. д.

Задание 3. Напишите текст документации, используя четкий и понятный язык. Включите в документацию подробные инструкции, примеры использования, скриншоты, диаграммы и другие визуальные элементы для улучшения понимания информации.

Задание 4. Проведите тестирование документации, выполняя описанные в ней процедуры и инструкции. Обратите внимание на возможные проблемы, несоответствия или недочеты в документации.

Тема 5.1. Оценка информационной системы для выявления возможности ее модернизации

Задание 1. Парно с одноклассниками обменяйтесь проектами разработанных информационных систем. Изучите основные

характеристики (функциональность, производительность, безопасность и удобство использования) исследуемой ИС.

Задание 2. Поставьте себя на место пользователя/администратора/технического специалиста исследуемой ИС. Выявите текущие проблемы и недостатки в работе информационной системы, которые могут потребовать модернизации.

Задание 3. Проведите оценку технического состояния исследуемой ИС. Изучите используемые технологии, аппаратное и программное обеспечение, а также возможности интеграции с другими системами.

Задание 4. Разработайте бизнес-план модернизации информационной системы, включающий оценку затрат и ожидаемые результаты. Подготовьте презентацию.

Типовые вопросы на зачет

1. Какие основные этапы жизненного цикла разработки информационных систем?

2. Какие этапы включает процесс проектирования и разработки ИС?

3. Какие методологии разработки информационных систем вы знаете? Опишите основные принципы каждой из них.

4. Что такое архитектура информационной системы, и какие принципы лежат в ее основе?

5. Какова роль баз данных в информационных системах?

6. Какие типы баз данных существуют, и как выбрать подходящую для конкретного проекта?

7. Какие факторы следует учитывать при выборе технологических платформ и инструментов для разработки информационных систем?

8. Какие этапы включает процесс тестирования информационных систем?

9. Какие методы тестирования вы используете?

10. Какие методы защиты данных вы знаете?

11. Каковы основные принципы управления проектами в области информационных систем? Какие инструменты используются для оценки и управления рисками?

12. Какие требования следует учитывать при разработке пользовательского интерфейса информационных систем, чтобы

обеспечить удобство использования и эффективность работы пользователей?

Литература

1. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г. Н. Федорова. – Москва : КУРС : ИНФРА-М, 2024. – 336 с.

2. Базовые принципы разработки программного обеспечения : учебное пособие / В. И. Шипков, Т. Р. Захаренкова, А. А. Нечаев, А. С. Грицай. – Омск : Омский государственный технический университет, 2023. – 116 с.

3. Гагарина, Л. Г. Введение в архитектуру программного обеспечения : учебное пособие / Л. Г. Гагарина, А. Р. Федоров, П. А. Федоров. – Москва : ФОРУМ : ИНФРА-М, 2023. – 320 с.

4. Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения : учебное пособие / А. В. Проскуряков. – Ростов-на-Дону ; Таганрог : Изд-во Южного федерального университета, 2022. – 197 с.

5. Роцин, П. Г. Командная разработка программного обеспечения с помощью системы контроля версий GIT: конспект лекций : учебное пособие / П. Г. Роцин. – Москва : Национальный исследовательский ядерный университет «МИФИ», 2022. – 106 с.

6. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для СПО / С. А. Чернышев. – Москва : Юрайт, 2025. – 176 с.

7. Ананьева, Т. Н. Стандартизация, сертификация и управление качеством программного обеспечения : учебное пособие / Т. Н. Ананьева, Н. Г. Новикова, Г. Н. Исаев. – Москва : ИНФРА-М, 2025. – 232 с.

8. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. – Москва : ФОРУМ : ИНФРА-М, 2023. – 400 с.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования
«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ Т. Ф. ГОРБАЧЕВА»

¶14пт интервал1,5

Кафедра _____

¶14пт интервал1,5

¶14пт интервал1,5

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ -16 пт

по профессиональному модулю ¶-14пт

14пт интервал1,5

¶14пт интервал1,5

Выполнил:

Студент группы ИСт-261

Иванов И. И.

¶14пт интервал1,5

Руководитель:

Преподаватель СПО

ФИО

14пт интервал1,5

Оценка _____

«___» _____ 20__ г.

_____/_____
Подпись (расшифровка подписи)

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

СОГЛАСОВАНО

Руководитель практики от профильной
организации

подпись Ф.И.О.

Руководитель практики из числа НПП КузГТУ

подпись Ф.И.О.

УТВЕРЖДАЮ

Заведующий кафедрой

подпись Ф.И.О.

«_____» _____ 20_ г.

ЗАДАНИЕ
на учебную/производственную практику

по профессиональному
модулю _____

(индекс и наименование профессионального модуля)

Обучающийся _____

Институт/факультет _____

Специальность _____

(код специальности)

Курс _____

Группа _____

Вид практики _____

Способ прохождения практики _____

Период прохождения практики с _____

по _____

Профильная организация _____

(наименование, местонахождение)

Руководитель практики из числа НПП КузГТУ _____

ФИО, должность

Руководитель практики от профильной организации _____

ФИО, должность

Индивидуальное задание на практику:

(в соответствии с ФОС)

Виды работ	Объем часов
1	
2	
...	

(в соответствии с разделом 2.2 программы практики)

Формируемые компетенции:

(в соответствии с разделом 1 программы практики)

Проведен инструктаж обучающегося по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка

«_____» _____ 20____ г.

Инструктаж провел _____

ФИО, должность руководителя практики от профильной организации
подпись

Инструктаж пройден _____

ФИО обучающегося

подпись обучающегося

ДНЕВНИК УЧЕБНОЙ/ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

ФИО, должность подпись

[illegible]

Приложение Г. Бланк аттестационного листа по УП, ПП

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

**АТТЕСТАЦИОННЫЙ ЛИСТ
по учебной/производственной практике**

по профессиональному модулю _____
(индекс и наименование профессионального модуля)

Обучающийся	
Институт/факультет	
Специальность	
(код специальности)	
Курс	Группа
Вид практики	
Способ прохождения практики	
Период прохождения практики с	по
Профильная организация	
(наименование, местонахождение)	

Во время прохождения практики обучающимся были освоены следующие профессиональные и общие компетенции

Наименование компетенции	Оценка	
	Освоена	Не освоена

Руководитель практики из числа НПП КузГТУ

ФИО, должность, подпись

Руководитель практики от профильной организации

ФИО, должность, подпись

Приложение Д. Бланк характеристики на обучающегося в период прохождения УП, ПП

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

ХАРАКТЕРИСТИКА
на обучающегося по освоению общих и профессиональных компетенций
в период прохождения учебной/производственной практики

по профессиональному модулю _____
(индекс и наименование профессионального модуля)

Обучающийся	
Институт/факультет	
Специальность	
(код специальности)	
Курс	Группа
Вид практики	
Способ прохождения практики	
Период прохождения практики с	по
Профильная организация	
(наименование, местонахождение)	

Виды и качество выполненных работ:

Виды работ	Критерии выполнения работ		
	Выполнены полностью самостоятельно	Выполнены с незначительной помощью наставника	Выполнены с помощью наставника

Руководитель практики из числа НПР КузГТУ

ФИО, должность подпись
Руководитель практики от профильной организации
ФИО, должность подпись