

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Романчук Иван Сергеевич
Должность: Ректор
Дата подписания: 14.02.2025 09:52:52
Уникальный программный ключ:
6319edc2b582ffda443f01d5779368d0957ac34f5cd074d81181530452479

Приложение к рабочей программе дисциплины

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Наименование дисциплины	Разработка систем обработки данных в предметных областях
Направление подготовки / Специальность	02.03.03 Математическое обеспечение и администрирование информационных систем
Направленность (профиль)/ Специализация	Технологии программирования и анализа больших данных
Форма обучения	очная
Разработчик(и)	Плотоненко Ю.А., доцент кафедры программного обеспечения

1. Темы дисциплины для самостоятельного освоения обучающимися
Отсутствуют

2. План самостоятельной работы

п/п	Учебные встречи	Виды самостоятельной работы	Форма отчетности/ контроля	Количество баллов	Рекомендуемый бюджет времени на выполнение (ак.ч.)*
1.	УВ №2. Изучение проблематики в области обработки данных.	Ознакомиться с основными проблемами, возникающими в процессе обработки данных, а также с современными подходами и методами их решения	Презентация: тема, обоснование актуальности и проблемы, предполагаемое решение проблемы.	3	6
2.	УВ №3. Подбор и изучение литературных источников.	Работа с рекомендуемой по теме литературой.	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	3	6
3.	УВ № 5–6. Углубленное изучение проблемы и уточнение цели проектирования. Проектирование технического задания.	Обоснование цели и задач проекта, составление технического задания	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	6	14
4.	УВ № 9–10. Проектирование функциональной модели ИС.	Формулирование функциональных требований и разработка модели ИС, которая поможет эффективно реализовать проектируемую систему	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	8	16
5.	УВ № 11–12. Проектирование информационного обеспечения.	Проектирование информационных систем, основанных на организации и обработке информации, а также на изучение принципов создания структуры и процессов для эффективного управления	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	6	16

		данными в рамках проекта.			
6.	УВ № 14–15. Реализация БД. Выбор технологии доступа к БД.	Проектирования, создания и реализации базы данных, а также анализ и выбор оптимальной технологии для доступа к БД в зависимости от требований проекта.	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	6	16
7.	УВ № 17–18. Проектирование ИС.	Проектирование, выбор и реализация алгоритмов и моделей для обработки данных, а также написание программного кода для реализуемой информационной системы.	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе. Код приложения.	8	16
8.	УВ № 19–20. Разработка архитектуры ИС.	Проектирования архитектуры приложения, которая включают в себя выбор структуры, компонентов и принципов взаимодействия для создания эффективного и масштабируемого приложения	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	8	16
9.	УВ № 22–23. Разработка интерфейса пользователя.	Проектирование и создание пользовательского интерфейса (UI) для проектируемой системы.	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе. Код приложения	8	16
10.	УВ № 25. Тестирование прототипа приложения.	Изучение методов тестирования приложений. Проведения тестовых испытаний реализуемой системы.	Собеседование с руководителем, ответы на вопросы. Письменный отчет о проделанной работе.	4	8
11.	Защита проектного решения	Подготовка к защите проектного решения	Отчет, презентация	40	30
	ИТОГО: часов самостоятельной работы				162

3. Требования и рекомендации по выполнению самостоятельных работ обучающихся, критерии оценивания

УВ №2. Практическое занятие 1. «Изучение проблематики в области обработки данных»

Самостоятельная работа студентов может быть организована таким образом, чтобы студенты не только усваивали теоретический материал, но и применяли знания на практике,

анализируя актуальные вопросы и вызовы в данной области. Важно, чтобы работа включала как исследовательские, так и практические элементы.

Цель самостоятельной работы — ознакомиться с основными проблемами, возникающими в процессе обработки данных, а также с современными подходами и методами их решения. Студенты должны анализировать текущие тенденции, проблемы и возможности в этой области, а также развивать навыки критического мышления и применения теоретических знаний на практике. Осуществить выбор темы проектирования.

Задачи:

- Выбрать тему проектного решения.
- Исследовать основные проблемы, связанные с обработкой больших данных (big data), в том числе вопросы хранения, скорости обработки и безопасности. Оценить влияние технологии машинного обучения и искусственного интеллекта на улучшение процессов обработки данных.
- Рассмотреть этические аспекты работы с данными, включая проблемы конфиденциальности и защиты персональных данных.
- Изучить современные инструменты и методы обработки данных (например, Hadoop, Spark, SQL, NoSQL).
- Проанализировать актуальность, выявить проблему и обосновать предполагаемое решение выбранного проектного решения.

Ожидаемые результаты:

Студенты должны подготовить презентацию, в которой они:

- Обоснуют актуальность проектного решения.
- Представят обзор текущих проблем в рамках предметной области выбранного проектного решения.
- Проанализируют реальные решения по данной проблематике.
- Предложат возможные пути решения изученных проблем, опираясь на современные инструменты и технологии.

Рекомендуемая структура работы:

Описание актуальности темы.

Обзор проблем в рамках выбранного решения

Разбор конкретных случаев для решения проблемы. Это может быть анализ успешных кейсов, использование современных инструментов и технологий.

Предполагаемое решение проблемы.

Оценка:

Оценка работы будет зависеть от полноты анализа, креативности в решении проблем и умения представлять информацию.

УВ №3. Практическое занятие 2. «Подбор и изучение литературных источников.»

Самостоятельная работа студентов направлена на развитие навыков поиска, анализа и критического осмысления научных и учебных материалов, которые будут использованы в процессе написания научных или исследовательских работ.

Цель самостоятельной работы:

Целью самостоятельной работы является развитие навыков эффективного поиска, выбора и анализа литературных источников, которые имеют отношение к теме проекта. Студенты должны научиться грамотно подходить к выбору источников, оценивать их научную ценность, а также эффективно использовать полученную информацию в контексте своего проектного решения.

Задачи:

- Научиться находить и использовать различные виды литературных источников: книги, научные статьи, отчеты, диссертации, онлайн-ресурсы и

базы данных.

- Овладеть методами оценки качества и релевантности источников.
- Научиться систематизировать и структурировать информацию, полученную из различных источников.
- Применить полученные знания для подготовки теоретической части проекта.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Подготовить список литературных источников, которые будут использованы в проекте.
- Проанализировать каждый источник, выделив ключевые идеи и концепты, которые будут полезны для дальнейшего исследования.
- Составить аннотированный библиографический список (краткое описание каждого источника с оценкой его полезности для проекта).

Рекомендуемая структура работы:

Введение: Описание выбранной темы проекта. Краткий обзор основных аспектов, которые будут исследованы.

Основная часть: Перечень использованных источников, включая книги, статьи, диссертации, материалы конференций и т.д. Краткое содержание и критический анализ каждого источника. Оценка релевантности источников и их связи с темой проекта.

Заключение: Основные выводы из анализа источников. Перспективы дальнейшего использования этих источников в проекте. План по дальнейшему поиску и изучению литературы, если это необходимо для завершения работы.

Список литературы: Правильное оформление всех использованных источников в соответствии с установленными стандартами.

Методы выполнения работы:

- Поиск информации: Использование научных библиотек, онлайн-ресурсов и баз данных (например, Google Scholar, JSTOR, Scopus, Web of Science, РИНЦ).
- Критический анализ: Оценка актуальности, достоверности и научной ценности каждого источника.
- Систематизация и структурирование: Организация литературных источников по ключевым направлениям исследования, выстраивание логической структуры для использования материалов в проекте.
- Оформление библиографии: Применение правильных стандартов оформления цитат и ссылок.

Оценка:

Оценка работы будет включать следующие критерии:

- Правильность выбора и разнообразие источников (книги, статьи, диссертации, ресурсы).
- Качество анализа и критического осмысления каждого источника.
- Соответствие литературных источников теме проекта.
- Правильность и точность оформления списка литературы.

Рекомендации:

- Использование качественных и проверенных источников: для написания научных работ важно опираться на авторитетные и актуальные публикации.
- Оценка источников: стоит внимательно оценивать, как научную ценность, так и методологическую составляющую материалов, чтобы избежать использования устаревших или ненадежных данных.

- Систематизация информации: важно не только подобрать литературу, но и грамотно систематизировать ее для дальнейшего использования в исследовательской работе.

УВ №5 -6. Практическое занятие 3–4. «Углубленное изучение проблемы и уточнение цели проектирования. Проектирование технического задания.»

Самостоятельная работа студентов направлена на развитие навыков детального анализа проблемы, формулировки целей и задач проекта и составления технического задания, которое является основой для дальнейшей работы над проектом.

Цель самостоятельной работы

Целью самостоятельной работы является углубленное понимание выбранной проблемы и формулировка точных целей и задач проектирования. Студенты должны научиться проводить детальный анализ проблемной области, уточнять цели и задачи проектирования, а также составлять техническое задание, которое будет служить руководством для реализации проекта.

Задачи:

- Углубленно изучить проблему, на решение которой направлен проект.
- Сформулировать четкие цели и задачи проектирования на основе анализа проблемы.
- На основе поставленных целей разработать техническое задание, которое будет являться основой для дальнейшей разработки проекта.
- Уточнить критерии успешности выполнения проекта, сроки, ограничения и ресурсы.

Ожидаемые результаты:

По итогам выполнения работы студент предоставляет отчет, в котором должен:

- Провести глубокий анализ проблемы, сформулировать основную цель и задачи проектирования, которые помогут достичь желаемого результата.
- Разработать техническое задание, которое будет включать в себя все необходимые компоненты: требования к проекту, ограничения, критерии качества и успешности, а также прогнозируемые сроки выполнения.

Рекомендуемая структура работы:

Введение: Анализ проблемы. Углубленное исследование проблемы и ее аспектов (технических, экономических, социальных и т.д.). Выявление основных трудностей и ограничений. Оценка рисков, которые могут возникнуть при проектировании. Формулировка цели и задач проектирования, которые будут способствовать достижению общей цели.

Проектирование технического задания: Указание требований к проектируемому продукту или решению (функциональные, технические, экономические и т.д.). Описание ограничений (время, бюджет, ресурсы). Формулировка критериев оценки успешности выполнения проекта.

Заключение: Обобщение основных выводов. Рекомендации по дальнейшему проектированию и уточнению технического задания.

Список использованных источников: Перечень литературы и других материалов, использованных при подготовке работы.

Методы выполнения работы:

- Формулировка целей: Разработка целей и задач проекта с учетом специфики проблемы, требований заказчика или потенциальных пользователей.

- Проектирование технического задания: Создание подробного технического задания, которое учитывает все аспекты проектирования, от требований к продукту до критериев оценки успешности.
- Использование стандартов и методик: Студенты должны применить стандарты и методики проектирования технического задания, соответствующие области их исследований (например, ГОСТ, ISO и другие стандарты).

Оценка:

Оценка работы будет базироваться на следующих критериях:

- Ясность и четкость сформулированных целей и задач проектирования.
- Соответствие технического задания поставленной цели и задачам проекта.
- Умение учитывать ограничения и риски при разработке технического задания.
- Качество и грамотность составления документации (техническое задание, план работы).

Рекомендации:

Подробный анализ: перед тем как приступить к составлению технического задания, важно детально проанализировать все аспекты проблемы, включая потребности пользователей и специфику решения.

Четкость и точность: цели и задачи проектирования должны быть сформулированы максимально ясно, чтобы их можно было четко интерпретировать на всех этапах реализации проекта.

Использование стандартов: техническое задание должно соответствовать принятым в отрасли стандартам и методикам, чтобы обеспечить высокое качество проектирования и легкость в реализации.

УВ № 9–10. Практическое занятие 5–6. «Проектирование функциональной модели ИС»

Самостоятельная работа студентов направлена на развитие навыков проектирования, моделирования и системного подхода при создании функциональных моделей информационных систем. Студенты должны научиться формулировать функциональные требования и разрабатывать модели, которые помогут эффективно реализовать проектируемую систему.

Цель самостоятельной работы:

Цель самостоятельной работы — овладение методами проектирования функциональных моделей информационных систем, способность формулировать функциональные требования и разрабатывать схемы, отображающие работу системы с точки зрения ее функций и взаимодействий между компонентами.

Задачи:

- Изучить и определить ключевые функции, которые должна выполнять информационная система в рамках выбранной темы проекта.
- Проанализировать и выделить важнейшие процессы и их взаимосвязи в рамках будущей системы.
- Разработать функциональную модель ИС с использованием подходящих инструментов и методов.
- Описать взаимодействия между компонентами системы и определить их функциональные характеристики.
- Представить функциональную модель в виде схем, диаграмм или других моделей, которые позволяют четко визуализировать структуру и работу системы.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Определить и описать ключевые функции информационной системы, которые необходимо реализовать для решения поставленных задач.
- Разработать и представить функциональную модель ИС, отражающую структуру и взаимосвязь компонентов.
- Использовать современные методы моделирования и инструментальные средства для проектирования (например, UML, DFD, BPMN и другие).
- Предложить описание взаимодействия компонентов системы, их роль и функцию в процессе работы ИС.

Рекомендуемая структура работы:

Введение: обоснование выбора функциональной модели для проектирования ИС. Краткое описание основных целей и задач проектирования. Анализ требований: Изучение и описание основных функциональных требований, которые предъявляются к информационной системе.

Основная часть: описание бизнес-процессов и ключевых задач, решаемых системой. Выделение основных процессов и функций, которые должны быть включены в модель. Разработка функциональной модели ИС, используя подходящие методы (например, диаграммы потоков данных (DFD), диаграммы использования (use case), диаграммы классов и т. д.). Описание взаимодействий компонентов системы (пользователи, базы данных, внешние системы и т. д.). Описание взаимосвязей и интерфейсов между различными компонентами системы. Приведение диаграмм и схем, иллюстрирующих работу модели.

Заключение: Основные выводы из выполненной работы. Перспективы и возможные улучшения функциональной модели.

Список литературы: Перечень использованных источников, включая книги, статьи, стандарты, методики и другие материалы.

Методы выполнения работы:

- Анализ функциональных требований: Студенты должны провести анализ задач и процессов, которые должна решать разрабатываемая информационная система.
- Моделирование: Создание диаграмм и схем, отражающих функции системы и взаимодействия между ее компонентами. Студенты могут использовать различные методы моделирования, такие как:
 - Диаграммы потоков данных (DFD).
 - Диаграммы использования (use case).
 - Диаграммы классов UML.
 - BPMN для моделирования бизнес-процессов.
- Инструменты проектирования: Студенты могут использовать программные средства для моделирования, такие как Microsoft Visio, Lucidchart, StarUML, Bizagi или другие.
- Описание взаимодействий: Важно детально описать, как различные компоненты системы взаимодействуют между собой, какие данные передаются, какие процессы выполняются и как это все влияет на работу системы в целом.

Оценка:

Оценка работы будет включать следующие критерии:

- Полнота и точность функциональных требований, определенных для системы.
- Правильность и наглядность представленных моделей и диаграмм.
- Логичность и последовательность в проектировании функциональной модели.
- Соответствие функциональной модели требованиям, предъявляемым к проекту.
- Качество и грамотность документации, сопровождающей модель.

Рекомендации:

- Понимание системы: перед началом проектирования важно тщательно проанализировать, какие функции должны быть реализованы в информационной системе, и какие процессы она будет поддерживать.
- Моделирование на разных уровнях: использование различных типов диаграмм позволяет более четко и полно представить систему. Не ограничивайтесь одной моделью, а используйте несколько, чтобы показать все аспекты работы системы.
- Документирование модели: каждая диаграмма должна быть хорошо описана, чтобы понять, что она представляет. Убедитесь, что все компоненты и их взаимодействия четко объяснены.
- Учет всех аспектов работы ИС: Функциональная модель должна учитывать все возможные взаимодействия с внешними системами и пользователями, чтобы обеспечить полноценную работу системы в реальных условиях.

УВ № 11–12. Практическое занятие 7–8. «Проектирование информационного обеспечения»

Самостоятельная работа студентов направлена на развитие навыков проектирования информационных систем, основанных на организации и обработке информации, а также на изучение принципов создания структуры и процессов для эффективного управления данными в рамках проекта. Студенты будут работать с задачами, связанными с разработкой информационного обеспечения, которое обеспечивает поддержку процессов в организации или системе.

Цель самостоятельной работы:

Целью самостоятельной работы является овладение методами проектирования информационного обеспечения, направленного на создание эффективных структур для сбора, обработки, хранения и предоставления информации, необходимой для функционирования системы или организации.

Задачи:

- Изучить ключевые принципы и методы проектирования информационного обеспечения.
- Разработать структуру информационного обеспечения для выбранной задачи или проекта, учитывая требования пользователей и специфики работы системы.
- Определить и спроектировать процессы сбора, обработки, хранения и представления информации.
- Разработать схемы, диаграммы и другие документы, которые описывают взаимодействие компонентов информационного обеспечения.
- Оценить и представить возможные инструменты и технологии, которые могут быть использованы для реализации информационного обеспечения.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Разработать структуру информационного обеспечения, которое будет соответствовать целям проекта.
- Описать процессы работы с информацией, включая ее сбор, хранение, обработку и вывод.
- Представить проект информационного обеспечения в виде схем, диаграмм и описаний, которые четко показывают взаимодействие между различными компонентами системы.
- Выбрать соответствующие инструменты и технологии для реализации информационного обеспечения.

Рекомендуемая структура работы:

Введение: Изучение требований пользователей к информационному обеспечению. Определение задач и целей, которые должна решать система информационного обеспечения. Описание функциональных и нефункциональных требований.

Основная часть: Разработка структуры, включающей компоненты для сбора, обработки и хранения информации. Описание взаимодействий между компонентами системы (например, баз данных, интерфейсы, процессы). Разработка процессов, связанных с обработкой информации, начиная от ввода и заканчивая выводом. Проектирование механизмов обработки данных (например, алгоритмы обработки, правила для обработки информации). Описание подходящих технологий для реализации информационного обеспечения (например, базы данных, системы управления контентом, аналитические инструменты и т. д.).

Заключение: Обобщение результатов работы. Рекомендации по дальнейшему улучшению информационного обеспечения.

Список литературы: Перечень использованных источников и материалов.

Методы выполнения работы:

- Анализ требований: студенты должны проводить анализ потребностей пользователей и специфики проекта, чтобы правильно спроектировать информационное обеспечение.
- Проектирование структуры: разработка компонентов системы, которые будут собирать, обрабатывать, хранить и предоставлять информацию. Важно учитывать как технические, так и организационные аспекты.
- Проектирование процессов: описание последовательности шагов, необходимых для работы с информацией, а также взаимодействие различных компонентов системы.
- Использование инструментов проектирования: студенты могут использовать различные инструменты для визуализации своих проектов, такие как диаграммы потоков данных (DFD), схемы базы данных, схемы взаимодействия пользователей с системой и т. д.
- Выбор технологий: студенты должны изучить и выбрать подходящие инструменты и технологии для реализации системы, учитывая требования к безопасности, масштабируемости, производительности и удобству работы с информацией.

Оценка:

Оценка работы будет проводиться по следующим критериям:

- Полнота и правильность анализа требований.
- Логичность и последовательность в проектировании структуры информационного обеспечения.
- Четкость и ясность в описании процессов работы с информацией.
- Соответствие выбранных технологий и инструментов требованиям проекта.
- Качество и наглядность представленных диаграмм и схем.
- Грамотность документации, поясняющей проект.

Рекомендации:

- Понимание потребностей пользователей: важно точно понять, какие задачи и проблемы необходимо решить с помощью информационного обеспечения, чтобы проектировать систему, которая будет удовлетворять эти потребности.
- Использование стандартов: проектируя структуру и процессы работы с информацией, важно следовать отраслевым стандартам и методологиям, чтобы проект был успешным и легко масштабируемым.
- Гибкость и адаптивность: информационное обеспечение должно быть гибким, чтобы оно могло адаптироваться к изменениям в требованиях и технологии.
- Практическое применение технологий: при выборе технологий нужно учитывать не только их актуальность, но и возможность их внедрения и поддержки в реальных условиях.

УВ № 14–15. Практическое занятие 9–10. «Реализация БД. Выбор технологии доступа к БД»

Самостоятельная работа студентов направлена на развитие навыков проектирования, создания и реализации базы данных, а также на анализ и выбор оптимальной технологии для доступа к БД в зависимости от требований проекта.

Цель самостоятельной работы:

Целью самостоятельной работы является освоение процесса проектирования, реализации и настройки базы данных, а также выбор подходящей технологии для доступа к базе данных. Студенты должны изучить различные типы баз данных, их архитектуру, способы доступа к ним и принципы оптимизации работы с данными.

Задачи:

- Изучить основные принципы проектирования баз данных.
- Разработать структуру базы данных с учетом требований проекта.
- Реализовать базу данных с использованием подходящей системы управления базами данных (СУБД).
- Оценить различные технологии доступа к БД, включая SQL и NoSQL решения.
- Выбрать и обосновать выбор технологии доступа к базе данных, учитывая особенности проекта (например, объем данных, скорость запросов, масштабируемость, безопасность).
- Реализовать подключение и взаимодействие с базой данных через выбранную технологию доступа.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Спроектировать структуру базы данных, включая таблицы, связи и ключи.
- Реализовать базу данных с использованием выбранной СУБД (например, MySQL, PostgreSQL, MongoDB, SQLite).
- Описать и реализовать механизмы доступа к базе данных через выбранные технологии (например, использование SQL-запросов или API).
- Оценить требования проекта и выбрать соответствующую технологию для доступа к базе данных.
- Реализовать примеры взаимодействия с базой данных (например, выполнение запросов на создание, обновление, удаление данных).

Рекомендуемая структура работы:

Проектирование базы данных: Описание требований к базе данных (например, описание сущностей, атрибутов и их связей). Проектирование схемы базы данных (таблицы, связи, индексы). Выбор подходящей СУБД для реализации (реляционная или нереляционная).

Реализация базы данных: Создание базы данных в выбранной СУБД. Разработка структуры таблиц, настройка связей и ограничений. Применение нормализации данных (если необходимо).

Выбор технологии доступа к базе данных: Описание различных технологий для доступа к данным (например, SQL, NoSQL, RESTful API, ORM). Оценка преимуществ и недостатков каждой технологии в контексте выбранного проекта. Обоснование выбора технологии. Реализация механизма доступа к базе данных с использованием выбранной технологии. Пример использования запросов для работы с данными (например, создание, чтение, обновление и удаление данных). Реализация подключений и обработки ошибок.

Заключение: Выводы по результатам работы. Рекомендации по улучшению реализации базы данных.

Список литературы: Перечень использованных источников (книги, статьи,

официальная документация).

Методы выполнения работы:

- Проектирование структуры базы данных: студенты должны проводить анализ требований, моделировать данные, проектировать таблицы и связи между ними, используя принципы нормализации.
- Выбор СУБД: оценка различных СУБД (например, реляционные — MySQL, PostgreSQL, или NoSQL — MongoDB, Redis) на основе специфики проекта (например, структура данных, масштабируемость, скорость работы с данными).
- Реализация базы данных: на основе выбранной СУБД студент должен создать базу данных и реализовать ее структуру.
- Выбор и реализация технологии доступа: студент должен изучить и выбрать подходящую технологию доступа к базе данных, такую как:
- SQL-запросы: для реляционных баз данных, например, MySQL или PostgreSQL.
- API RESTful: для работы с данными через HTTP, часто используется с NoSQL СУБД или реляционными БД в веб-разработке.
- ORM: использование библиотек для объектно-реляционного отображения (например, SQLAlchemy для Python, Hibernate для Java) для работы с БД через объектно-ориентированный подход.
- Тестирование и оптимизация: проверка корректности работы запросов и оптимизация работы с базой данных (например, индексирование, создание оптимальных запросов).

Оценка:

Оценка работы будет базироваться на следующих критериях:

- Качество проектирования базы данных: корректность структуры таблиц, нормализация, описание сущностей и связей.
- Правильность выбора СУБД: обоснование выбора СУБД и технологий доступа в контексте проекта.
- Реализация доступа: корректность и эффективность реализации механизма доступа (например, работа с запросами, использование API).
- Оптимизация и тестирование: процесс тестирования, исправление ошибок и оптимизация запросов.
- Документация: ясность и полнота пояснений, корректность оформления работы.

Рекомендации:

- Понимание требований проекта: при выборе СУБД и технологии доступа важно учитывать особенности данных (например, их объем, структуру, требования к скорости обработки) и сценарии работы с ними.
- Использование индексов и оптимизация запросов: важно не только создать базу данных, но и правильно оптимизировать работу с ней, применяя индексы и правильное построение запросов.
- Выбор подходящей технологии: важно учитывать, что не всегда SQL и реляционные базы данных являются наилучшим решением, особенно для работы с большими объемами данных или нереляционными данными. В таких случаях выбор NoSQL может быть более оправданным.

УВ № 17–18. Практическое занятие 11–12. «Проектирование ИС»

Самостоятельная работа студентов направлена на развитие навыков проектирования, выбора и реализации алгоритмов для обработки данных, а также на написание программного кода для модуля информационной системы. В ходе работы студенты должны научиться разрабатывать функциональные части системы, выбирать подходящие методы обработки данных и реализовывать их с помощью программного кода.

Цель самостоятельной работы:

Целью самостоятельной работы является освоение процесса проектирования информационной системы, выбор и реализация алгоритмов, моделей обработки данных, а также разработка программного кода для реализации этих алгоритмов, моделей и модулей системы в рамках проектного решения.

Задачи:

- Проектировать структуру информационной системы, выделяя ключевые модули и их взаимодействие.
- Выбрать и обосновать выбор алгоритмов, моделей обработки данных, которые необходимы для решения поставленных задач.
- Реализовать выбранные алгоритмы, модели в виде программного кода.
- Написать код для модулей информационной системы, обеспечивающий выполнение обработки данных.
- Протестировать разработанные модули, проверив корректность работы алгоритмов, моделей и их взаимодействие в системе.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Разработать проект информационной системы с учетом выбранной темы, выделив основные модули и их функциональность.
- Выбрать и обосновать выбор алгоритмов и моделей для обработки данных, подходящих для решения поставленных задач.
- Написать программный код для реализации этих алгоритмов и моделей в рамках информационной системы.
- Провести тестирование модулей и убедиться в корректной работе и соответствии заданным требованиям.

Рекомендуемая структура работы:

Разработка ИС: разработка общей структуры системы, выделение основных компонентов и модулей. Описание взаимодействия между модулями. Разработка архитектуры и схемы системы (например, диаграммы компонентов, диаграммы потоков данных). Выбор и реализация алгоритмов и моделей обработки данных. Описание выбранных алгоритмов и моделей, их принципов работы и обоснование выбора. Реализация алгоритмов в виде программного кода: описание структуры данных, с которыми будут работать алгоритмы (например, списки, массивы, деревья). Написание кода для реализации модулей системы. Реализация функционала обработки данных согласно алгоритмам и моделям. Описание работы кода и его структуры.

Тестирование и отладка: проверка корректности работы алгоритмов и программного кода. Проведение тестов для проверки правильности обработки данных. Исправление ошибок и оптимизация кода.

Заключение: Обобщение выполненной работы, выводы о достигнутых результатах. Рекомендации по возможным улучшениям системы.

Список литературы: Перечень использованных источников и материалов.

Методы выполнения работы:

- Анализ и проектирование системы: Студенты должны анализировать требования проекта, выделить ключевые компоненты системы и спроектировать взаимодействие между ними.
- Выбор алгоритмов и моделей: Студенты выбирают алгоритмы и модели обработки данных, которые лучше всего подходят для решения задач, учитывая тип данных и требования к производительности. Это могут быть алгоритмы сортировки, поиска, обработки графов, модели машинного обучения и другие.
- Реализация алгоритмов и моделей: Студенты реализуют алгоритмы на выбранном

языке программирования, например, Python, Java, C++, используя структурированные подходы и написание эффективного кода.

- Интеграция алгоритмов и моделей в систему: Реализованные алгоритмы и модели должны быть интегрированы в общий модуль информационной системы, обеспечивая выполнение всех необходимых операций с данными.
- Тестирование и отладка: Студенты должны протестировать систему на различных наборах данных, выявить и устранить ошибки, а также провести оптимизацию кода для улучшения производительности.

Оценка:

Оценка работы будет включать следующие критерии:

- Качество проектирования информационной системы: Логичность и правильность структуры системы, четкость взаимосвязей между модулями.
- Выбор и обоснование алгоритмов и моделей: Оценка выбранных алгоритмов и моделей с точки зрения их эффективности и применимости к задаче.
- Качество написанного кода: читаемость, эффективность, отсутствие ошибок, использование современных практик программирования.
- Тестирование и корректность работы системы: проверка того, насколько корректно и эффективно работают алгоритмы и модули.
- Документация: пояснения к проекту, описания алгоритмов и моделей, структуры кода и результатов тестирования.

Рекомендации:

- Правильный выбор алгоритмов и моделей: важно выбирать алгоритмы и модели, которые оптимальны с точки зрения сложности и соответствуют требованиям проекта. Например, для обработки больших объемов данных может быть полезно использовать алгоритмы с более низкой временной сложностью.
- Модульность и структура кода: Код должен быть структурированным, с четким разделением на модули, что поможет легко поддерживать и расширять систему в будущем.
- Оптимизация и тестирование: обратите внимание на оптимизацию алгоритмов и корректность их работы в реальных условиях. Регулярное тестирование на разных данных поможет выявить и исправить ошибки.
- Документирование кода: пишите комментарии к коду и алгоритмам, чтобы другие разработчики могли легко понять логику работы программы.

УВ № 19–20. Практическое занятие 13–14. «Разработка архитектуры ИС»

Самостоятельная работа студентов направлена на формирование у студентов навыков проектирования архитектуры программных приложений, которые включают в себя выбор структуры, компонентов и принципов взаимодействия для создания эффективного и масштабируемого приложения. Студенты должны научиться определять основные требования к системе и разрабатывать архитектурные решения, которые обеспечат надежную и продуктивную работу приложения.

Цель самостоятельной работы:

Целью самостоятельной работы является освоение принципов разработки архитектуры приложения, изучить различные подходы и методы проектирования архитектурных решений, а также научиться разрабатывать масштабируемую и надежную архитектуру для приложения по выбранной теме проекта.

Задачи:

- Изучить и проанализировать требования к проекту для понимания всех аспектов и функций будущего приложения.
- Разработать общую архитектуру приложения, включая выбор компонентов, их

взаимодействие и распределение задач.

- Оценить и выбрать подходящие архитектурные стили (например, клиент-сервер, многослойная архитектура, микросервисы и т. д.).
- Спроектировать взаимодействие между компонентами приложения (например, базы данных, пользовательский интерфейс, серверная часть).
- Описать используемые технологии и инструменты, подходящие для реализации архитектуры.
- Применить архитектурные паттерны и принципы проектирования для создания устойчивой и масштабируемой системы.
- Протестировать предложенную архитектуру на примере небольшого функционала или схем.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Проектировать архитектуру приложения, обеспечивающую выполнение всех функциональных требований.
- Выбрать подходящие архитектурные стили и компоненты для реализации приложения.
- Разработать подробное описание архитектуры, включая схемы, диаграммы и описания компонентов.
- Обосновать выбор технологий и инструментов для реализации архитектуры.
- Оценить возможные проблемы и предложить решения для обеспечения масштабируемости, безопасности и надежности приложения.

Рекомендуемая структура работы:

Введение: определение ключевых компонентов системы и их взаимодействий. Определение пользовательских сценариев, которые должны быть учтены при проектировании архитектуры. Обоснование выбора подходящего стиля архитектуры в зависимости от специфики проекта (например, для масштабируемости, безопасности или удобства разработки).

Проектирование компонентов системы: разработка схемы архитектуры, включая описание основных компонентов (например, серверная часть, клиентская часть, база данных, внешние сервисы и т. д.). Описание взаимодействий между компонентами, включая использование API, протоколов и технологий. Применение архитектурных паттернов (например, MVC, MVVM, микросервисная архитектура).

Технологии и инструменты: выбор технологий для реализации компонентов приложения (например, языки программирования, фреймворки, базы данных, облачные решения и т. д.). Описание инструментов для реализации архитектуры, таких как инструменты для управления зависимостями, интеграции и тестирования.

Моделирование архитектуры: создание диаграмм компонентов, диаграмм потоков данных, диаграмм взаимодействия и других визуальных моделей. Представление схем взаимодействий между клиентом, сервером и базой данных.

Заключение: обобщение выполненной работы, анализ архитектурных решений. Рекомендации по улучшению архитектуры или возможным изменениям в будущем.

Список литературы: перечень использованных источников (книги, статьи, научные публикации, официальная документация по инструментам).

Методы выполнения работы:

- Анализ требований: Студенты должны сначала провести тщательный анализ требований к приложению. Это включает в себя понимание нужд пользователей, целевых платформ и возможностей расширения системы.
- Выбор архитектурного стиля: На основе требований студент должен выбрать наиболее подходящий архитектурный стиль и обосновать его выбор. Например, для проектов с высокой нагрузкой и потребностью в масштабируемости может быть

выбрана микросервисная архитектура.

- Проектирование компонентов и их взаимодействия: Проектирование каждого компонента системы, включая серверную часть, клиентскую часть, базу данных и внешние сервисы. Важно продумать, как компоненты будут взаимодействовать друг с другом, какие протоколы и API будут использоваться.
- Моделирование архитектуры: Студенты должны использовать диаграммы и схемы для отображения архитектуры системы. Например, можно использовать UML-диаграммы или диаграммы потоков данных для визуализации компонентов и их взаимодействий.
- Выбор технологий: Необходимо выбрать и обосновать выбор технологий для реализации архитектуры. Например, если проект предполагает использование веб-приложения, то могут быть выбраны фреймворки для фронтенда (например, React, Angular) и для бэкенда (например, Django, Flask).

Оценка:

Оценка работы будет базироваться на следующих критериях:

- Качество проектирования архитектуры: Насколько хорошо спроектирована структура системы, ее компоненты и взаимодействия.
- Правильность выбора архитектурного стиля: Обоснование выбора подходящего архитектурного стиля и его соответствие требованиям проекта.
- Качество описания компонентов и их взаимодействий: Насколько понятно и полно описаны компоненты системы, их роль и способы взаимодействия.
- Использование современных технологий и инструментов: Оценка выбранных технологий с точки зрения их актуальности и соответствия требованиям системы.
- Документация и схемы: Четкость и наглядность представленных диаграмм и схем, ясность в описаниях архитектурных решений.

Рекомендации:

- Понимание требований проекта: Очень важно детально проанализировать требования проекта, чтобы архитектура точно соответствовала целям и задачам.
- Масштабируемость и надежность: Архитектура должна быть спроектирована так, чтобы система могла эффективно масштабироваться и выдерживать рост нагрузки.
- Использование паттернов проектирования: Архитектурные паттерны, такие как MVC, слоистая архитектура или микросервисы, могут значительно улучшить структуру системы, сделав ее более удобной для разработки и поддержки.

УВ № 22–23. Практическое занятие 15–16. «Разработка интерфейса пользователя»

Самостоятельная работа студентов направлена на формирование у студентов навыков проектирования и создания удобных, интуитивно понятных и функциональных пользовательских интерфейсов (UI) для приложений или информационных систем. Студенты должны научиться разрабатывать интерфейсы, которые соответствуют требованиям пользователей, а также обеспечивают удобство работы с системой.

Цель самостоятельной работы:

Целью самостоятельной работы является овладение навыками проектирования и разработки интерфейса пользователя для системы по выбранной теме проекта, с учетом принципов удобства, доступности, визуальной привлекательности и функциональности интерфейса.

Задачи:

- Провести анализ целевой аудитории и потребностей пользователей системы.
- Разработать дизайн интерфейса, который будет удобен, понятен и эффективен для пользователей.
- Проектировать структуру интерфейса, включая расположение элементов

управления, кнопок, меню, форм и других элементов.

- Выбрать и применить подходящие инструменты для разработки интерфейса (например, фреймворки, библиотеки, графические редакторы).
- Реализовать интерфейс пользователя с использованием выбранных технологий и инструментов.
- Протестировать интерфейс, получить обратную связь от пользователей и внести улучшения в дизайн.
- Описать методы улучшения пользовательского опыта (UX) и подходы к адаптации интерфейса для разных устройств и экранов (мобильных, десктопных).

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Разработать макеты и прототипы пользовательского интерфейса для системы.
- Осуществить разработку и реализацию интерфейса, соответствующего принципам удобства, функциональности и визуальной привлекательности.
- Протестировать интерфейс, выявить и устранить возможные проблемы.
- Описать и обосновать выбор элементов интерфейса, а также технологические решения для их реализации.

Рекомендуемая структура работы:

Введение: описание темы проекта и цели разработки интерфейса. Задачи, которые должен решать интерфейс пользователя (например, упрощение взаимодействия с системой, улучшение пользовательского опыта). Обоснование выбора подхода к разработке интерфейса.

Анализ требований: описание целевой аудитории и потребностей пользователей (например, кто будет использовать систему, какие задачи они должны решать). Анализ функциональных и нефункциональных требований к интерфейсу. Изучение существующих решений и пользовательских предпочтений в аналогичных системах.

Проектирование интерфейса: разработка структурных элементов интерфейса (например, меню, панели инструментов, окна и формы). Создание прототипов и макетов интерфейса (например, с использованием инструментов для создания wireframe — Figma, Sketch, Adobe XD). Описание взаимодействия пользователя с системой через интерфейс: как должны быть размещены элементы управления, какие действия доступны пользователю.

Выбор технологий для разработки интерфейса: описание инструментов и технологий, которые будут использованы для разработки интерфейса (например, HTML, CSS, JavaScript, фреймворки типа React, Angular, Vue.js для веб-разработки или Swift для iOS, Kotlin для Android). Оценка выбранных технологий с точки зрения простоты, гибкости и удобства.

Разработка интерфейса: реализация интерфейса на основе разработанных макетов и прототипов. Использование выбранных технологий и фреймворков для создания функциональных элементов интерфейса. Применение принципов адаптивности интерфейса (например, для различных размеров экранов и устройств).

Тестирование интерфейса: проведение тестирования интерфейса на разных устройствах и платформах. Сбор обратной связи от пользователей (например, через тестирование с реальными пользователями или коллегами). Исправление обнаруженных проблем (например, улучшение удобства навигации, исправление ошибок в дизайне).

Заключение: обобщение выполненной работы, выводы о созданном интерфейсе. Рекомендации по дальнейшему улучшению интерфейса и увеличению удобства использования.

Список литературы: перечень использованных источников, таких как книги, статьи, видеоуроки и документация по инструментам и технологиям.

Методы выполнения работы:

- Анализ потребностей пользователей: Началом разработки интерфейса является анализ того, что нужно пользователю, какова его целевая аудитория и какие задачи система должна помочь решить. Важно понимать особенности работы пользователя, его уровень компетенции и предпочтений.
- Проектирование прототипов: Студенты должны использовать инструменты для создания прототипов и макетов интерфейса, такие как Figma, Adobe XD или Sketch. Эти прототипы могут быть как статичными (изображения), так и интерактивными (кликбельные модели, демонстрирующие взаимодействие).
- Выбор технологий: Студенты должны выбрать подходящие технологии и инструменты для реализации интерфейса. В случае веб-приложений это могут быть фреймворки для фронтенда (например, React, Angular или Vue.js), а для мобильных приложений — Android Studio (для Android) или Xcode (для iOS).
- Реализация интерфейса: Студенты должны создать интерфейс с использованием выбранных технологий, приложив усилия к его адаптивности и удобству, включая такие элементы, как формы ввода данных, кнопки, меню, диалоговые окна, всплывающие подсказки и т. д.
- Тестирование и улучшение интерфейса: Важно проводить тестирование интерфейса, выявлять и устранять проблемы, связанные с удобством, производительностью и функциональностью.

Оценка:

Оценка работы будет основываться на следующих критериях:

- Качество проектирования интерфейса: насколько интерфейс соответствует потребностям пользователей, насколько удобен и интуитивно понятен.
- Технологическое исполнение: использование современных инструментов и технологий для разработки интерфейса, а также качество реализации.
- Тестирование и оптимизация: применение методов тестирования интерфейса, корректность работы на различных устройствах и платформах.
- Документация: четкость и полнота описания разработки интерфейса, а также аргументация выбора технологий и решений.

Рекомендации:

- Соблюдение принципов UX/UI: важно следовать современным рекомендациям по дизайну пользовательских интерфейсов, таким как простота, визуальная иерархия, минимализм и последовательность.
- Тестирование с реальными пользователями: проводите тестирование интерфейса с реальными пользователями, чтобы удостовериться в его удобстве и доступности.
- Адаптивность: обеспечьте, чтобы интерфейс хорошо смотрелся и был удобен на различных устройствах (мобильных телефонах, планшетах, десктопах).

УВ № 25. Практическое занятие 17. «Тестирование прототипа приложения»

Самостоятельная работа студентов направлена на развитие навыков проверки работоспособности и качества интерфейса и функционала прототипа приложения. Студенты должны изучить методы тестирования, научиться проводить различные виды тестов и анализировать результаты с целью улучшения приложения.

Цель самостоятельной работы:

Целью самостоятельной работы является освоение методов тестирования прототипа приложения, включая функциональное, юзабилити и производительное тестирование, а также способность выявлять и устранять дефекты на ранних стадиях разработки.

Задачи:

- Проанализировать прототип приложения, выявить ключевые функциональные элементы и пользовательские сценарии.

- Разработать тестовый план для проверки прототипа, включая виды тестирования и сценарии.
- Провести функциональное тестирование для проверки корректности работы основных функций.
- Осуществить юзабилити-тестирование для оценки удобства использования интерфейса.
- Провести тестирование на производительность, чтобы удостовериться в том, что приложение работает быстро и эффективно.
- Оценить результаты тестирования и предложить рекомендации для улучшения приложения.
- Провести исправления, если выявлены ошибки или недочеты, и повторно протестировать систему.

Ожидаемые результаты:

По итогам выполнения работы студент должен:

- Разработать план тестирования, включая описание типов тестов и сценариев.
- Провести тестирование прототипа приложения по выбранным направлениям (функциональное, юзабилити, производительное).
- Оценить результаты тестирования и предложить улучшения.
- Обнаружить и устранить выявленные ошибки или дефекты в прототипе.

Рекомендуемая структура работы:

Введение: обоснование выбора методов тестирования и их актуальности для проекта.

Разработка тестового плана: описание целей тестирования, методов, этапов и видов тестов. Составление списка тестов для разных типов проверок: функциональных, юзабилити и производительности. Разработка тестовых сценариев и наборов данных для тестирования.

Функциональное тестирование: проведение проверки основных функциональных элементов приложения: проверка правильности работы кнопок, форм, переходов и других действий. Оценка работы всех функций прототипа в условиях реального использования. Подготовка отчетов по выявленным ошибкам или некорректному поведению.

Юзабилити-тестирование: оценка удобства использования интерфейса приложения. Проведение тестов на основе реальных пользователей (например, студентов, коллег или других людей), чтобы понять, насколько интерфейс интуитивно понятен. Сбор обратной связи от тестируемых и анализ полученных данных.

Тестирование производительности: оценка времени отклика приложения при выполнении различных действий (например, загрузка страниц, обработка запросов). Проведение нагрузочного тестирования (например, проверка работы приложения при увеличении числа пользователей или больших объемах данных). Оценка результатов и выявление возможных узких мест в производительности.

Оценка результатов тестирования: сравнение результатов с исходными требованиями и ожиданиями. Оценка выявленных проблем и ошибок. Формулировка рекомендаций для улучшения прототипа на основе тестирования.

Заключение: обобщение работы, выводы о качестве прототипа приложения. Рекомендации по дальнейшему улучшению приложения на основе проведенного тестирования.

Список литературы: перечень использованных источников: книги, статьи, исследования и документация по тестированию приложений.

Методы выполнения работы:

- Анализ требований: Прежде чем начать тестирование, важно понять, какие задачи должна решать система, какие функции имеют наибольшее значение для пользователей. Это помогает сосредоточиться на критичных аспектах прототипа.

- Разработка тестового плана: Студенты должны создать план тестирования, который будет включать разные типы тестов (функциональные, юзабилити, производительные). План должен быть подробным и четким, включая конкретные сценарии, которые необходимо проверить.
- Функциональное тестирование: Студенты должны провести тестирование на предмет того, работает ли приложение в соответствии с заявленными требованиями. Это может включать проверку работы всех функций, таких как формы ввода, кнопки, переходы между страницами и т. д.
- Юзабилити-тестирование: Важно оценить, насколько удобно использовать интерфейс. Студенты могут провести тестирование с реальными пользователями, чтобы выявить, какие моменты вызывают трудности и где интерфейс можно улучшить.
- Производительное тестирование: Тестирование производительности позволяет оценить, как приложение ведет себя при больших нагрузках, например, при увеличении числа пользователей или объема данных.
- Документирование результатов: Результаты тестирования необходимо документировать, чтобы точно зафиксировать выявленные ошибки, недостатки и предложенные улучшения.

Оценка:

Оценка работы будет основана на следующих критериях:

- Качество тестового плана: оценка полноты и продуманности плана тестирования, охватывает ли он все важные аспекты прототипа.
- Результаты функционального тестирования: оценка выявленных ошибок, насколько правильно работают основные функции приложения.
- Оценка юзабилити: оценка удобства интерфейса, его интуитивности и соответствия ожиданиям пользователей.
- Производительность: оценка быстродействия приложения и его реакции на высокие нагрузки.
- Документирование и анализ результатов: четкость и полнота отчетов по тестированию, наличие рекомендаций по улучшению.

Рекомендации:

- Использование реальных пользователей для тестирования: при проведении юзабилити-тестирования желательно привлекать реальных пользователей, чтобы лучше понять, как интерфейс воспринимается на практике.
- Тестирование на разных устройствах и платформах: если приложение предполагает работу на различных устройствах (например, мобильных и десктопных), важно тестировать его в этих условиях.
- Нагрузочное тестирование: проверка на производительность должна включать сценарии с высоким числом пользователей или сложными запросами, чтобы выявить узкие места в системе.

4. Рекомендации по самоподготовке к промежуточной аттестации по дисциплине

Вопросы к экзамену

1. Изучение проблематики в области обработки данных

- Какие основные проблемы могут возникать при обработке больших данных?
- Какие технологии и подходы используются для обработки данных в реальном времени?
- В чем заключается отличие между структурированными и неструктурированными данными?
- Какую роль в обработке данных играют алгоритмы машинного обучения?

- Какие методы предобработки данных наиболее популярны в практике обработки больших данных?

2. Подбор и изучение литературных источников по выбранной теме проекта

- Какие типы источников следует учитывать при исследовании научной литературы по проекту?
- Как правильно сформулировать тему и цель исследования при подборе источников?
- Какие критерии качества литературных источников существуют?
- Как отличить первичные и вторичные источники информации?
- Почему важно использовать актуальные и проверенные научные исследования при подборе литературы для проекта?

3. Углубленное изучение проблемы и уточнение цели проектирования. Проектирование технического задания

- Какие компоненты обязательно должны быть включены в техническое задание?
- Какие этапы включает в себя процесс уточнения цели проектирования?
- В чем заключается разница между функциональными и нефункциональными требованиями?
- Как правильно составить описание проблемы, которая решается в проекте?
- Какие методы помогают определить возможные ограничения и риски проекта при составлении технического задания?

4. Проектирование функциональной модели ИС по выбранной теме проекта

- Что такое функциональная модель информационной системы?
- Какие основные компоненты должны быть включены в функциональную модель ИС?
- В чем разница между логической и физической моделью ИС?
- Каковы основные принципы проектирования функциональной модели?
- Какие преимущества и недостатки могут быть у выбранной функциональной модели?

5. Проектирование информационного обеспечения

- Какие основные требования предъявляются к информационному обеспечению системы?
- Как связаны концепции безопасности и доступа к данным с проектированием информационного обеспечения?
- Какие виды баз данных чаще всего используются для информационного обеспечения ИС?
- Каковы основные этапы проектирования информационного обеспечения?
- Какие методы и инструменты проектирования информационного обеспечения позволяют улучшить производительность системы?
- Реализация БД. Выбор технологии доступа к БД
- Какие основные типы баз данных существуют и чем они различаются?
- Какие факторы нужно учитывать при выборе технологии доступа к базе данных?
- В чем разница между реляционными и нереляционными базами данных?
- Какую роль играют индексы в ускорении работы с базой данных?
- Какие инструменты можно использовать для интеграции базы данных в информационную систему?

7. Проектирование информационной системы по выбранной теме проекта, выбор и реализация алгоритмов моделей обработки данных, написание программного кода модуля проектного решения

- Какие типы алгоритмов обработки данных могут быть использованы для решения конкретной задачи?
- В чем заключается разница между блок-схемами и псевдокодом?
- Как правильно выбрать подходящий алгоритм для обработки данных на основе требований проекта?

- Какие факторы нужно учитывать при написании кода для модуля проектного решения?
- Что такое декомпозиция задач и как она применяется при проектировании ИС?

8. Разработка архитектуры приложения по выбранной теме проекта

- Какие ключевые принципы необходимо учитывать при разработке архитектуры приложения?
- Каковы основные архитектурные стили и когда каждый из них целесообразен?
- Что такое микросервисная архитектура и в каких случаях она подходит?
- Какие компоненты входят в многослойную архитектуру приложения?
- Как обеспечивается масштабируемость и отказоустойчивость приложения при проектировании архитектуры?

9. Разработка интерфейса пользователя разрабатываемой системы по выбранной теме проекта

- Какие основные принципы интерфейсного дизайна существуют для улучшения пользовательского опыта?
- Как различаются понятия UI (User Interface) и UX (User Experience)?
- Почему важна адаптивность интерфейса, и как это влияет на разработку?
- Какие методы могут помочь сделать интерфейс интуитивно понятным для пользователя?
- Что такое дизайн-паттерны, и как их использование улучшает интерфейс?

10. Тестирование прототипа приложения

- Что включает в себя функциональное тестирование и какие типы ошибок оно помогает выявить?
- Что такое юзабилити-тестирование, и какие методы используются для его проведения?
- Каковы особенности проведения нагрузочного тестирования на прототипе?
- Какие критерии используются для оценки производительности приложения в процессе тестирования?
- Какова роль тестирования с реальными пользователями в процессе разработки прототипа?

Система оценивания

При проведении промежуточной аттестации результаты, полученные обучающимся в семестре, переводятся в формат традиционной оценки в соответствии со шкалой перевода баллов:

- 60 баллов и менее – «неудовлетворительно»;
- от 61 до 75 баллов – «удовлетворительно»;
- от 76 до 90 баллов – «хорошо»;
- от 91 до 100 баллов – «отлично».

Форма промежуточной аттестации – экзамен. экзамен проходит в форме публичной защиты проектного решения.

Критерии для оценки публичной защиты:

1. Проектная работа соответствует цели и отвечает на проблемные вопросы. (0–8 баллов)
2. Участником проекта проведены коллективные обсуждения в ходе работы над проектом, учтены все замечания. (0–6 баллов)
3. Результаты работы, представленные при помощи компьютерных средств, оформлены в соответствии с правилами. (0–6 баллов)
4. Устное выступление участника логично, отсутствуют грамматические и лексические

ошибки. (0–5 баллов)

5. Выступление не повторяет текст презентации или публикации. (0–5 баллов)

6. В ходе устного выступления даны ответы на вопросы. (0–10 баллов)

Критерии выставления оценки:

Оценка «отлично»

1. Проектная работа соответствует цели и отвечает на проблемные вопросы.
2. Участником проекта проведены коллективные обсуждения в ходе работы над проектом, учтены все замечания.
3. Результаты работы, представленные при помощи компьютерных средств, оформлены в соответствии с правилами.
4. Устное выступление участника логично, отсутствуют грамматические и лексические ошибки.
5. Выступление не повторяет текст презентации или публикации.
6. В ходе устного выступления даны ответы на все вопросы.

Оценка «хорошо»

1. Проектная работа соответствует цели и отвечает на некоторые проблемные вопросы.
2. Участником проекта проведены коллективные обсуждения в ходе работы над проектом, учтены практически все замечания.
3. Результаты работы, представленные при помощи компьютерных средств, содержат незначительные ошибки в оформлении.
4. Устное выступление участника логично, присутствуют незначительные грамматические и лексические ошибки, не мешающие пониманию материала.
5. Выступление частично повторяет текст презентации или публикации.
6. В ходе устного выступления даны ответы на некоторые вопросы.

Оценка «удовлетворительно»

1. Проектная работа не совсем точно отражает цель проекта и его проблемные вопросы.
2. Участник проекта провел коллективное обсуждение в ходе работы над проектом, но замечания учтены не полностью.
3. Результаты работы, представленные при помощи компьютерных средств, содержат незначительные ошибки в оформлении.
4. Устное выступление участника не всегда логично, присутствуют грамматические и лексические ошибки, которые затрудняют понимание.
5. Выступление полностью повторяет текст презентации или публикации.
6. Обучающийся затруднялся давать правильные ответы на вопросы.

Оценка «неудовлетворительно»

Отсутствуют или не представлены результаты проектного решения.