МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

имени академика М.Д. Миллионщикова

«УТВЕРЖДАЮ»
Первый проректор –
проректор по ОД
И.Г. Гапрабеков

«25»__06__2026 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Технологии распределенного проектирования»

Направление подготовки

09.04.02 Информационные системы и технологии

Направленность (профиль)

«Информационные системы и технологии»

Квалификация

магистр

Год начала подготовки - 2026

Грозный – 2026

1. Цели и задачи дисциплины

Цели дисциплины: раскрыть смысл ключевых понятий из области распределенной разработки, сформировать представление о современных архитектурах, моделях, методах и технологиях организации распределенной разработки, привить навыки работы в команде при разработке программного обеспечения.

Задачи дисциплины: приобретение студентами базового набора знаний из области распределенной разработки.

2. Место дисциплины в структуре образовательной программы

Данная дисциплина относится к части, формируемая участниками образовательных отношений образовательной программы по направлению 09.04.02 Информационные системы и технологии.

Предшествующие дисциплины, освоение которых необходимо для изучения данной дисциплины:

- Методы и системы принятия решений;
- Поисковые системы для научных исследований, обработка и представление результатов научных исследований;
- Методы и средства для распознавания образов и визуализации;
- Распознавание образов и когнитивная графика.

Последующие дисциплины, для которых данная дисциплина является предшествующей:

- Облачные технологии и сервисы.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Таблица 1

Код по ФГОС	**Индикаторы достижения**	**Планируемые результаты обучения по дисциплине (ЗУВ)**
Профессиональные компетенции		
ПК-1. Способен управлять внедрением, предоставлением, использованием и развитием цифровых и информационных технологий	**ПК-1.1.** Осуществляет планирование научных и прикладных исследований в области информационных технологий **ПК-1.2**. Организует исполнение научных и прикладных исследований в области информационных технологий **ПК-1.3.** Производит контроль качества научных и прикладных исследований в области информационных	**Знать:** - модели жизненного цикла и методологии управления – каскадная, спиральная, Agile, их применимость в распределенных командах. - архитектурные подходы и требования – клиент-серверная, многоуровневая, микросервисная архитектура; функциональные и нефункциональные требования. - средства автоматизации и контейнеризации – CI/CD, Docker, оркестрация, облачные технологии, основы DevSecOps. **Уметь:** - обоснованно выбирать модель разработки и методологию под конкретный распределенный проект с учетом состава

	технологий	команды, географии и требований. - организовывать совместную работу с использованием Git, систем управления задачами и эффективных коммуникаций. -настраивать конвейеры CI/CD и применять контейнеризацию для упаковки приложений. **Владеть:** - навыками работы с распределенными репозиториями и проектными досками – ведение бэклога, планирование спринтов, контроль выполнения задач. -приемами контроля качества кода – Code Review, статический анализ, написание автоматизированных тестов (модульных, интеграционных, регрессионных). -практиками использования Docker и CI/CD-инструментов – создание образов, управление контейнерами, настройка пайплайнов и базовые меры безопасности.
ПК - 6. Способен управлять этапами жизненного цикла методической и технологической инфраструктуры анализа больших данных в организации; разработкой продуктов, услуг и решений на основе больших данных; разработкой и внедрением новых методов и технологий исследований больших данных	**ПК – 6.1.** Управляет получением, хранением, передачей, обработкой больших данных **ПК – 6.2.** Управляет защитой и обеспечением конфиденциальности больших данных **ПК – 6.3.** Участвует в разработке сервисов на основе аналитики больших данных **ПК – 6.4.** Участвует в совершенствовании и разработке рекомендаций новых методов, моделей, алгоритмов, технологий и инструментальных средств работы с большими данными	**Знать:** - модели ЖЦ и гибкие методологии. - архитектурные паттерны для распределённых систем обработки данных. - **с**редства автоматизации (CI/CD), контейнеризация (Docker) и DevSecOps. **Уметь:** - планировать и координировать разработку компонентов системы больших данных. - организовывать совместную разработку (Git, Code Review, системы задач). - настраивать CI/CD и контейнеризацию с учётом безопасности. **Владеть:** - навыками работы с распределёнными репозиториями и проектными. - навыками контейнеризация (Docker) и основы оркестрации. - навыками контроля качества кода, автоматизированное тестирование и базовые меры безопасности.

4. Объем дисциплины и виды учебной работы

Таблица 2

Вид учебной работы	**Всего часов/ зач.ед.**

		4 семестр	
		ОФО	**ЗФО**
Контактная работа (всего)		**32/0,9**	**18/0,5**
В том числе:			
Лекции		8/0,23	6/0,2
Практические занятия			
Семинары			
Лабораторные работы		24/0,67	12/0,3
Самостоятельная работа (всего)		**112/3,1**	**126/3,5**
В том числе:			
Курсовая работа (проект)			
Расчетно-графические работы			
ИТР			
Задания повышенной сложности		36/1	90/2,5
Доклады			
Презентации			
И (или) другие виды самостоятельной работы:			
Подготовка к лабораторным работам		36/1	36/1
Подготовка к практическим занятиям			
Подготовка к экзамену		3/0,9	36/1
Вид отчетности		**зачет**	**зачет**
Общая трудоемкость дисциплины	**ВСЕГО в часах**	**144**	**144**
	ВСЕГО в зач. единицах	**4**	**4**

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Таблица 3

№ п/п	Наименование раздела дисциплины по семестрам	Лекц. зан. часы		Лаб. зан. часы		Всего часов	
		ОФО	ЗФО	ОФО	ЗФО	ОФО	ЗФО
1	Основы распределенного проектирования	2	2	6	2	8	4
2	Проектирование программных систем	2	2	6	4	8	6
3	Средства коллективной разработки	2	1	6	4	8	5
4	Современные технологии разработки	2	1	6	2	8	3
	Всего	**8**	**6**	**24**	**12**	**32**	**18**

5.2. Лекционные занятия

Таблица 4

№ п/п	Наименование раздела дисциплины	Содержание раздела

1.	Основы распределенного проектирования	Введение в распределенное проектирование Жизненный цикл программного обеспечения Методологии управления проектами
2.	Проектирование программных систем	Анализ и управление требованиями Архитектурное проектирование программных систем
3.	Средства коллективной разработки	Системы контроля версий Платформы управления проектами Коммуникации в распределенных командах Совместная разработка программного обеспечения
4.	Современные технологии разработки	Тестирование программного обеспечения Непрерывная интеграция и доставка Облачные технологии и контейнеризация Безопасность распределенной разработки Современные тенденции распределенного проектирования

5.3. Лабораторный практику

Таблица 4

№ п/п	**Наименование раздела дисциплины**	**Наименование лабораторных работ**
1.	Основы распределенного проектирования	ЛР1. Выбор модели ЖЦ и методологии управления ЛР2. Планирование проекта в системе управления задачами
2.	Проектирование программных систем	ЛР3. Разработка клиент-серверного приложения ЛР4. Микросервисная архитектура и взаимодействие
3.	Средства коллективной разработки	ЛР5. Организация совместной разработки с Git ЛР6. Code Review, стандарты кодирования и качество кода
4.	Современные технологии разработки	ЛР7. Контейнеризация и оркестрация ЛР8. CI/CD и безопасность

5.4. Практические (семинарские) занятия: нет

Таблица 6

№ п/п	**Наименование раздела дисциплины**	**Содержание раздела**
1.	**-**	**-**

6. Самостоятельная работа студентов по дисциплине

В качестве самостоятельной работы студент выполняет задания повышенной сложности.

1. Создание микросервисной системы с нулевым временем простоя (Zero-Downtime Deployment)

Реализовать кластер из 3–4 микросервисов с базой данных, настроить blue-green деплой или canary-релизы с использованием Kubernetes и CI/CD.

2. Разработка распределённой системы мониторинга и логирования

Интегрировать стек ELK (Elasticsearch, Logstash, Kibana) или Prometheus + Grafana для сбора метрик и логов с нескольких сервисов, настроить алертинг.

3. Внедрение практик DevSecOps в пайплайн CI/CD

Добавить в существующий конвейер статический анализ кода, сканирование уязвимостей образов (Trivy), проверку секретов и политик безопасности, настроить автоматическое оповещение об уязвимостях.

4. Проектирование гибридной архитектуры для обработки потоковых данных

Спроектировать и реализовать систему приёма и обработки данных в реальном времени (Kafka + Spark Streaming / Flink), с сохранением результатов в хранилище (ClickHouse / MinIO), с автоматическим масштабированием.

5. Разработка сервиса аналитики с применением Low-code / No-code платформы

Используя одну из платформ (Node-RED, Budibase, Appsmith), создать интерфейс для визуализации аналитических отчётов, подключив внешние API и базы данных, и организовать контейнеризацию всего решения.

6. Исследование влияния архитектурных решений на производительность в распределённой системе

Провести серию нагрузочных тестов для монолита и микросервисной версии одного и того же приложения, сравнить задержки, пропускную способность, надёжность, сделать выводы и рекомендации.

Образец задания: *Разработка и развертывание микросервисной системы интернет-магазина с применением CI/CD, контейнеризации и централизованного логирования*

Цель работы:

Спроектировать, реализовать и развернуть в облачной среде (или локальном кластере) работающий прототип интернет-магазина, состоящий из трёх микросервисов (каталог товаров, корзина, заказы), а также настроить автоматизированный пайплайн непрерывной интеграции и доставки, организовать сбор логов и метрик.

Задачи (этапы выполнения):

1. **Проектирование**
   - Сформулировать функциональные и нефункциональные требования.
   - Выбрать модель ЖЦ и методологию (например, Scrum), составить бэклог и спринт-план.
   - Разработать диаграмму архитектуры (взаимодействие сервисов, используемые протоколы, базы данных).
2. **Реализация**
   - Написать код каждого микросервиса на выбранном языке (Python/Java/Go) с REST API.
   - Каждый сервис должен иметь собственную базу данных (PostgreSQL / MongoDB).

- Обеспечить взаимодействие через синхронные (REST) и асинхронные (RabbitMQ/Kafka) каналы (хотя бы для одного сценария).

3. **Контейнеризация**
   - Написать Dockerfile для каждого сервиса.
   - Создать docker-compose для локального развёртывания всей системы с зависимостями.
4. **Организация совместной разработки**
   - Вести репозиторий на GitHub/GitLab с использованием Git Flow (ветки develop, feature, release).
   - Настроить автоматические проверки (линтеры, тесты) на каждый Pull Request.
   - Провести минимум один Code Review внутри группы (при групповой работе).
5. **Настройка CI/CD пайплайна**
   - В GitHub Actions / GitLab CI реализовать:
     - автоматическую сборку образов при push в main/develop;
     - запуск модульных и интеграционных тестов;
     - публикацию образов в контейнерный регистр (Docker Hub / GitHub Container Registry);
     - автоматическое развертывание на тестовый стенд (или в minikube) при успешном прохождении тестов.
6. **Мониторинг и логирование**
   - Подключить централизованное логирование (ELK или Loki + Grafana) для всех сервисов.
   - Настроить сбор основных метрик (время ответа, количество запросов, ошибки) и дашборд.
7. **Обеспечение безопасности**
   - Включить сканирование уязвимостей образов в CI/CD.
   - Использовать переменные окружения / секреты для паролей и ключей API.
   - Настроить базовую аутентификацию (JWT) для доступа к сервисам.
8. **Документирование и отчёт**
   - Подготовить описание архитектуры, использованных технологий и принятых решений.
   - Приложить ссылку на репозиторий, инструкцию по развёртыванию, скриншоты работающей системы и дашбордов.
   - Провести нагрузочное тестирование (например, с помощью JMeter или k6) и проанализировать результаты.

Требования к результату:

- Работающий прототип, доступный для проверки (локально или в облаке).
- Репозиторий с историей коммитов, ветками, Pull Request.
- Отчёт (10–15 страниц) с описанием всех этапов, трудностей и найденных решений.

Учебно-методическое обеспечение для самостоятельной работы:

1. Зыков, С. В. Проектирование и разработка корпоративных информационных систем / С. В. Зыков. — Москва: Ай Пи Ар Медиа, 2023. — 394 с.

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

7. Оценочные средства

7.1. Вопросы к текущему контролю:

Часть 1

1. Основные модели организации разработки в распределённых командах.
2. Преимущества и недостатки распределённой работы по сравнению с офисной.
3. Этапы жизненного цикла ПО и кратко охарактеризовать каскадную модель.
4. Agile, Scrum и Kanban – сравнение и ключевые отличия.
5. Примеры функциональных и нефункциональных требований для интернет-магазина.
6. Разница между клиент-серверной и многоуровневой архитектурами.
7. Описать принципы микросервисного подхода и его основные преимущества.
8. Источники требований и способы управления изменениями в требованиях.

Часть 2

9. Основные команды Git для работы с ветками и слиянием изменений.
10. Описать назначение систем контроля версий и роль Pull Request в совместной работе.
11. Функции платформ управления проектами (на примере Jira или Trello).
12. Практики Code Review и стандарты кодирования, повышающие качество кода.
13. Различия между модульным, интеграционным и регрессионным тестированием.
14. Этапы конвейера CI/CD и его значение для распределённой разработки.
15. Основные возможности Docker и его роль в контейнеризации приложений.
16. Меры по защите исходного кода и данных при распределённой работе (DevSecOps).

7.2. Вопросы к зачету / экзамену

Вопросы к зачету

1. Основные модели организации разработки в распределённых командах.
2. Преимущества и недостатки распределённой работы по сравнению с офисной.
3. Этапы жизненного цикла ПО и кратко охарактеризовать каскадную модель.
4. Agile, Scrum и Kanban – сравнение и ключевые отличия.
5. Примеры функциональных и нефункциональных требований для интернет-магазина.
6. Разница между клиент-серверной и многоуровневой архитектурами.
7. Описать принципы микросервисного подхода и его основные преимущества.
8. Источники требований и способы управления изменениями в требованиях.
9. Основные команды Git для работы с ветками и слиянием изменений.
10. Описать назначение систем контроля версий и роль Pull Request в совместной работе.
11. Функции платформ управления проектами (на примере Jira или Trello).
12. Практики Code Review и стандарты кодирования, повышающие качество кода.
13. Различия между модульным, интеграционным и регрессионным тестированием.

14. Этапы конвейера CI/CD и его значение для распределённой разработки.
15. Основные возможности Docker и его роль в контейнеризации приложений.
16. Меры по защите исходного кода и данных при распределённой работе (DevSecOps).

Образец билета к зачету

Грозненский Государственный Нефтяной Технический Университет
им. акад. М.Д. Миллионщикова
Кафедра «Информационные технологии»
Дисциплина «Технологии распределенного проектирования»
Группа: **Семестр: 4**

Билет №2

1. Преимущества и недостатки распределённой работы по сравнению с офисной
3. Этапы конвейера CI/CD и его значение для распределённой разработки

Подпись преподавателя_____ **Подпись заведующего кафедрой**_____

7.3. Текущий контроль

Образец типового задания для лабораторной работы

Лабораторная работа № 5 Организация совместной разработки с Git

Цель работы:

Освоить работу с Git на Windows, научиться создавать локальные и удаленные репозитории, работать с ветками, pull request, code review и разрешать конфликты.

1. Подготовка к работе

1. Установить Git для Windows:
   - Скачать с https://git-scm.com/download/win
   - Установить Git Bash (для командной строки) и Git GUI (графический интерфейс).
2. Установить текстовый редактор для работы с кодом:
   - **Visual Studio Code** (https://code.visualstudio.com) (если не установлен)
3. Создать аккаунт на платформе:
   - **GitHub /GitLab**

2. Создание удаленного репозитория

1. Войти на выбранную платформу и создать новый репозиторий, например:
   lab1-distributed-project-windows
2. Установить видимость (Private или Public)
3. Скопировать ссылку для клонирования (HTTPS или SSH)

3. Клонирование репозитория на Windows

Через Git Bash:

cd /c/Users/<ВашеИмя>/Documents/

git clone <URL вашего репозитория>

cd lab1-distributed-project-windows

git status

Через Git GUI:

- Открыть Git GUI → Clone Existing Repository → Вставить URL → выбрать локальную папку.

4. Создание и работа с ветками

Через Git Bash:

git checkout -b feature/first-task

1. Создать файл README.md через **VS Code**:

Лабораторная работа 1

Работа с Git и распределенным репозиторием

2. Добавить и закоммитить изменения:

git add README.md

git commit -m "Добавлен README для лабораторной работы 1 (Windows)"

git push origin feature/first-task

Через Git GUI:

- Stage Changed → Commit → Push

5. Pull request и code review

1. На GitHub/GitLab создаем Pull Request из feature/first-task в main/master.
2. Добавляем описание изменений.
3. Если в команде, коллеги выполняют review, оставляют комментарии.
4. Merge после согласования.

6. Разрешение конфликтов

Сценарий: Изменения в README.md были сделаны и в ветке main.

Через Git Bash:

git checkout main

git merge feature/first-task

- Git покажет конфликт, откройте файл README.md в VS Code и исправьте конфликт вручную.
- После исправления:

git add README.md

git commit -m "Разрешен конфликт при слиянии веток"

git push origin main

Через Git GUI:

- Pull → Resolve Conflicts → Stage → Commit → Push

Отчет по лабораторной работе
В отчете должны быть:

1. Скриншоты репозитория до и после работы.
2. История коммитов (git log) в Git Bash.
3. Скриншоты Pull Request и code review.
4. Скриншоты разрешения конфликта.

7.4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

Таблица 7

Планируемые результаты освоения компетенции	Критерии оценивания результатов обучения				Наименование оценочного средства
	менее 41 баллов (неудовлетворительно)	**41-60 баллов (удовлетворительно)**	**61-80 баллов (хорошо)**	**81-100 баллов (отлично)**	
***ПК-1**: Способен управлять внедрением, предоставлением, использованием и развитием цифровых и информационных технологий*					
Знать: - модели жизненного цикла и методологии управления – каскадная, спиральная, Agile, их применимость в распределенных командах. - архитектурные подходы и требования – клиент-серверная, многоуровневая, микросервисная архитектура; функциональные и нефункциональные требования. - средства автоматизации и контейнеризации – CI/CD, Docker, оркестрация, облачные технологии, основы DevSecOps.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	Комплект заданий для выполнения лабораторных работ, темы докладов с презентациями, вопросы по темам / разделам дисциплины
Уметь: - обоснованно выбирать модель разработки и методологию под конкретный распределенный проект с учетом состава команды, географии и требований. - организовывать совместную работу с использованием Git, систем управления задачами и эффективных коммуникаций. -настраивать конвейеры CI/CD и применять контейнеризацию для упаковки приложений.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	

Владеть: - навыками работы с распределенными репозиториями и проектными досками – ведение бэклога, планирование спринтов, контроль выполнения задач. -приемами контроля качества кода – Code Review, статический анализ, написание автоматизированных тестов (модульных, интеграционных, регрессионных). -практиками использования Docker и CI/CD-инструментов – создание образов, управление контейнерами, настройка пайплайнов и базовые меры безопасности.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	
ПК – 6: *Способен управлять этапами жизненного цикла методической и технологической инфраструктуры анализа больших данных в организации; разработкой продуктов, услуг и решений на основе больших данных; разработкой и внедрением новых методов и технологий исследований больших данных*					
Знать: - модели ЖЦ и гибкие методологии. - архитектурные паттерны для распределённых систем обработки данных. - средства автоматизации (CI/CD), контейнеризация (Docker) и DevSecOps	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	Комплект заданий для выполнения лабораторных работ, темы докладов с презентациями, вопросы по темам / разделам дисциплины
Уметь: - планировать и координировать разработку компонентов системы больших данных. - организовывать совместную разработку (Git, Code Review, системы задач). - настраивать CI/CD и контейнеризацию с учётом безопасности.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: - навыками работы с распределёнными репозиториями и проектными. - навыками контейнеризация (Docker) и основы оркестрации. - навыками контроля качества кода, автоматизированное тестирование и базовые меры безопасности.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

8. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся созданы фонды оценочных средств, адаптированные для инвалидов и лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе. Форма проведения текущей аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При тестировании для слабовидящих студентов используются фонды оценочных средств с укрупненным шрифтом. На экзамен приглашается сопровождающий, который обеспечивает техническое сопровождение студенту. При необходимости студенту-инвалиду предоставляется дополнительное время для подготовки ответа на экзамене (или зачете). Обучающиеся с ограниченными возможностями здоровья и обучающиеся инвалиды обеспечиваются печатными и электронными образовательными ресурсами (программы, учебные пособия для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

1) для инвалидов и лиц с ограниченными возможностями здоровья **по зрению:**

- **для слепых:** задания для выполнения на семинарах и практических занятиях оформляются рельефно-точечным шрифтом Брайля или в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом; письменные задания выполняются на бумаге рельефно-точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых либо надиктовываются ассистенту; обучающимся для выполнения задания при необходимости предоставляется комплект письменных принадлежностей и бумага для письма рельефно-точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;

- **для слабовидящих:** обеспечивается индивидуальное равномерное освещение не менее 300 люкс; обучающимся для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; задания для выполнения заданий оформляются увеличенным шрифтом;

2) для инвалидов и лиц с ограниченными возможностями здоровья **по слуху:**

- **для глухих и слабослышащих:** обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования; предоставляются услуги сурдопереводчика;

- **для слепоглухих** допускается присутствие ассистента, оказывающего услуги тифлосурдопереводчика (помимо требований, выполняемых соответственно для слепых и глухих);

3) для лиц с тяжелыми нарушениями речи, глухих, слабослышащих лекции и семинары, проводимые в устной форме, проводятся в письменной форме;

4) для инвалидов и лиц с ограниченными возможностями здоровья, **имеющих нарушения опорно-двигательного аппарата:**

- для лиц с нарушениями опорно-двигательного аппарата, нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей: письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту; выполнение заданий (тестов, контрольных работ), проводимые в письменной форме, проводятся в устной форме путем опроса, беседы с обучающимся.

9. Учебно-методическое и информационное обеспечение дисциплины

1. Зыков, С. В. Проектирование и разработка корпоративных информационных систем / С. В. Зыков. — Москва: Ай Пи Ар Медиа, 2023. — 394 с.

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

10. Материально-техническое обеспечение дисциплины

10.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Перечень материально-технических средств учебной аудитории для проведения занятий по дисциплине:

- учебная аудитория, доска;
- стационарные компьютеры;
- мультимедийный проектор;
- настенный экран.

10.2. Помещения для самостоятельной работы

Учебная аудитория для самостоятельной работы – 4-06.

Методические указания по освоению дисциплины «Технологии распределенного проектирования»

1. Методические указания для обучающихся по планированию и организации времени, необходимого для освоения дисциплины.

Изучение рекомендуется начать с ознакомления с рабочей программой дисциплины, ее структурой и содержанием разделов (модулей), фондом оценочных средств, ознакомиться с учебно-методическим и информационным обеспечением дисциплины.

Дисциплина «Разработка информационных хранилищ» состоит из 4 связанных между собою разделов, обеспечивающих последовательное Технологии распределенного проектирования изучение материала.

Обучение по дисциплине «Технологии распределенного проектирования» осуществляется в следующих формах:

1. Аудиторные занятия (лекции, лабораторные занятия).
2. Самостоятельная работа студента (подготовка к лекциям, лабораторным занятиям, докладам и иным формам письменных работ, индивидуальная консультация с преподавателем).
3. Интерактивные формы проведения занятий (коллоквиум, лекция-дискуссия и др.формы).

Учебный материал структурирован и изучение дисциплины производится в тематической последовательности. Каждой лабораторно работе и самостоятельному изучению материала предшествует лекция по данной теме. Обучающиеся самостоятельно проводят предварительную подготовку к занятию, принимают активное и творческое участие в обсуждении теоретических вопросов, разборе проблемных ситуаций и поисков путей их решения. Многие проблемы, изучаемые в курсе, носят дискуссионный характер, что предполагает интерактивный характер проведения занятий на конкретных примерах.

Описание последовательности действий обучающегося:

При изучении курса следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий:

1. После окончания учебных занятий для закрепления материала просмотреть и обдумать текст лекции, прослушанной сегодня, разобрать рассмотренные примеры (10 – 15 минут).
2. При подготовке к лекции следующего дня повторить текст предыдущей лекции, подумать о том, какая может быть следующая тема (10 - 15 минут).
3. В течение недели выбрать время для работы с литературой в библиотеке (по 1 часу).
4. При подготовке к лабораторному занятию основные понятия по теме, изучить примеры. Решая конкретную ситуацию, - предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить 1 - 2 практические ситуации (лаб.работы).

2. Методические указания по работе обучающихся во время проведения лекций.

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает преподаватель, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью

разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, необходимо использовать не только основную литературу, но и ту литературу, которую дополнительно рекомендовал преподаватель. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

Тематика лекций дается в рабочей программе дисциплины.

3. Методические указания обучающимся по подготовке к практическим/семинарским занятиям.

На лабораторных занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительныйматериал по тематике семинарских занятий.

Студенту рекомендуется следующая схема подготовки к семинарскому занятию:

1. Ознакомление с планом лабораторного занятия, который отражает содержание предложенной темы;

2. Проработать конспект лекций;

3. Прочитать основную и дополнительную литературу.

В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов отношение к конкретной проблеме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса;

4. Ответить на вопросы плана лабораторного занятия;

5. Выполнить домашнее задание;

6. Проработать тестовые задания и задачи;

7. При затруднениях сформулировать вопросы к преподавателю.

Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы практикума, выступать и участвовать в коллективном обсуждении вопросов изучаемой темы, правильно выполнять практические задания и иные задания, которые даются в фонде оценочных средств дисциплины.

3. Методические указания обучающимся по организации самостоятельной работы.

Цель организации самостоятельной работы по дисциплине «Технологии распределенного проектирования» - это получение новых знаний и навыков в разработке и проектировании распределенных информационных систем.

Самостоятельная работа обучающихся является важнейшим видом освоения содержания дисциплины, подготовки к практическим занятиям и к контрольной работе. Сюда же относятся и самостоятельное углубленное изучение тем дисциплины. Самостоятельная работа представляет собой постоянно действующую систему, основу образовательного процесса и носит исследовательский характер, что послужит в будущем основанием для написания выпускной квалификационной работы,

практического применения полученных знаний.

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению, с учетом потребностей и возможностей личности.

Правильная организация самостоятельных учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет студентам развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивать высокий уровень успеваемости в период обучения, получить навыки повышения профессионального уровня.

Самостоятельная работа реализуется:

– непосредственно в процессе аудиторных занятий - на лекциях, лабораторных занятиях;

– в контакте с преподавателем вне рамок расписания - на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.

– в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Составители:

Старший преподаватель  /Шабазов И. М./

СОГЛАСОВАНО:

Зав. выпускающей каф. «ИТ»  / Усамов И.Р./

Руководитель направления
магистерской подготовки  / Алисултанова Э.Д./

Директор ДУМР  / Магомаева М.А./