

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Андреевич

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 14:50:30

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии канд.
техн. наук, доцент Порохня А.А.

Ф.И.О.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Продвинутая разработка программного обеспечения

Направление подготовки
Направленность (профиль)

09.04.04 Программная инженерия
Разработка, мониторинг и управление
жизненным циклом инженерных систем
2026
очная
2

Год начала обучения
Форма обучения
Реализуется в семестре

Разработано

К.ф.-м.н., доцент, заведующая межинститутской
базовой кафедрой

Новикова Елена Николаевна

Ставрополь 2026 г.

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины: формирование у обучающихся системы теоретических знаний и практических навыков в области продвинутых методов и инструментов разработки программного обеспечения, обеспечивающих возможность проектировать, реализовывать, оптимизировать и сопровождать сквозные конвейеры непрерывной интеграции, поставки и развертывания (CI/CD) для программных систем и встроенного ПО, интегрируя практики автоматизированного тестирования, проверки безопасности, управления артефактами и конфигурациями в единый автоматизированный процесс поставки качественного и безопасного программного продукта.

Задачи освоения дисциплины:

1. Проектирование CI/CD пайплайнов
 - Изучить архитектуру и компоненты сквозных конвейеров непрерывной интеграции, поставки и развертывания.
 - Освоить принципы построения пайплайнов для различных типов приложений (монолитные, микросервисные, встраиваемые системы).
2. Реализация автоматизированного тестирования в CI/CD
 - Научиться интегрировать unit-тесты, интеграционные тесты и тесты покрытия кода в конвейер.
 - Освоить настройку автоматического запуска тестов при каждом изменении кода и анализ результатов тестирования.
3. Интеграция проверки безопасности в пайплайн
 - Изучить инструменты статического анализа безопасности (SAST) и их встраивание в CI/CD.
 - Научиться настраивать автоматическую блокировку поставки при обнаружении критических уязвимостей.
4. Управление артефактами сборки
 - Освоить практики создания, версионирования (SemVer) и хранения артефактов сборки (библиотеки, Docker-образы).
 - Научиться интегрировать бинарные репозитории в процесс непрерывной поставки.
5. Управление конфигурациями в CI/CD
 - Изучить подход «конфигурация как код» (Configuration as Code).
 - Освоить безопасное управление секретами и переменными окружения в пайплайне (переменные CI/CD, внешние секрет-менеджеры).
6. Оптимизация производительности пайплайнов
 - Научиться применять асинхронное и многопоточное выполнение этапов для ускорения конвейеров.
 - Освоить параллелизацию задач (parallel jobs, matrix builds) и кэширование зависимостей.
7. Реализация продвинутых стратегий развертывания
 - Изучить стратегии Blue-Green, Canary, Rolling Update и их автоматизацию через CD-пайплайн.
 - Научиться настраивать автоматический откат при сбоях на основе health-проверок.
8. Обеспечение наблюдаемости пайплайна и деплоя
 - Освоить интеграцию метрик, логов и алертов в процесс развертывания для контроля состояния системы.
 - Научиться автоматически проверять SLI/SLO после деплоя.

9. Внедрение GitOps подхода
 - Изучить принципы GitOps для управления инфраструктурой и приложениями через CI/CD.
 - Освоить инструменты синхронизации кластера с репозиторием (ArgoCD).
10. Разработка сквозного CI/CD проекта
 - Сформировать практические навыки построения полного пайплайна «от коммита до деплоя» с интеграцией всех изученных практик на примере реального (учебного) программного проекта.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Продвинутая разработка программного обеспечения» относится к части, формируемой участниками образовательных отношений (вариативная часть), и является одной из ключевых дисциплин профиля «Разработка, мониторинг и управление жизненным циклом инженерных систем».

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-1. Способен проектировать, реализовывать, оптимизировать и поддерживать сквозные конвейеры непрерывной интеграции, поставки и развертывания (CI/CD) для программного обеспечения и встроенных систем, интегрируя практики автоматизированного тестирования, проверки безопасности, управления артефактами и конфигурациями	ИД-1 ПК-1. Проектирует архитектуру CI/CD пайплайна с учётом требований к тестированию, безопасности и развертыванию для различных типов приложений (монолитные, микросервисные, встраиваемые системы)	Способен проектировать архитектуру CI/CD пайплайна, выбирая состав этапов (линтинг, тестирование, SAST, сборка, деплой) в зависимости от типа приложения и требований к качеству, и описывать её в форматах конвейеров как код (.gitlab-ci.yml, .sourcecraft.yml).
	ИД-2 ПК-1. Реализует пайплайн непрерывной интеграции, включающий этапы линтинга, автоматизированного тестирования (unit, интеграционные), статического анализа безопасности (SAST) и измерения покрытия кода	Способен реализовывать пайплайн непрерывной интеграции, интегрируя в него этапы линтинга, unit- и интеграционных тестов, SAST-анализа и измерения покрытия кода с настройкой автоматической блокировки сборки при нарушении пороговых значений Quality Gate.
	ИД-3 ПК-1. Настраивает автоматическую сборку, версионирование (SemVer) и публикацию артефактов (библиотеки, Docker-образы) в бинарный репозиторий	Способен настраивать автоматическую сборку, семантическое версионирование и публикацию артефактов (Python wheel, Docker-образы) в бинарный репозиторий с автоматическим формированием тегов на основе git-метаданных (tag, commit hash).

	ИД-4 ПК-1 Интегрирует управление конфигурациями и секретами в пайплайн, обеспечивая безопасную подстановку переменных окружения для различных сред (dev/staging/prod)	Анализирует и внедряет управление конфигурациями как код и безопасное хранение секретов, обеспечивая подстановку переменных окружения для разных сред (dev/staging/prod) через переменные CI/CD или интеграцию с HashiCorp Vault.
	ИД-5 ПК-1 Оптимизирует производительность пайплайна за счёт параллелизации этапов, кэширования зависимостей и использования асинхронных/многопоточных подходов	Способен оптимизировать производительность CI/CD пайплайна, применяя параллелизацию этапов (matrix, parallel), кэширование зависимостей и асинхронные/многопоточные подходы для ускорения I/O-операций и CPU-bound задач.
	ИД-6 ПК-1. Реализует стратегии развертывания (Blue-Green, Canary, Rolling Update) и автоматический откат при сбоях на основе health-проверок	Реализует стратегии развертывания (Blue-Green, Canary, Rolling Update) и настраивает автоматический откат на основе health-проверок (readinessProbe, livenessProbe) и анализа кодов ошибок (5xx, таймауты).
	ИД-7 ПК-1. Внедряет GitOps подход с использованием ArgoCD для синхронизации состояния кластера с репозиторием манифестов	Внедряет GitOps подход, настраивая ArgoCD для синхронизации состояния Kubernetes кластера с репозиторием манифестов и обеспечивая автоматическое применение изменений при обновлении тега образа через CI/CD.
	ИД-8 ПК-1. Настраивает наблюдаемость пайплайна (метрики, логи, алерты) и проверку SLI/SLO после деплоя	Настраивает наблюдаемость CI/CD пайплайна, интегрируя сбор метрик (Prometheus), визуализацию (Grafana) и автоматическую проверку SLI/SLO после деплоя с возможностью отката при нарушении целевых показателей надёжности.

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 4 з.е. 144 ак.ч.	ОФО, в астр. часах
Контактная работа:	48
Лекции/из них практическая подготовка	16
Лабораторных работ/из них практическая подготовка	32
Практических занятий/из них практическая подготовка	
Самостоятельная работа	96
Формы контроля	
Экзамен	
Зачет	
Зачет с оценкой	±
Расчетно-графические работы	
Курсовые работы	±
Контрольные работы	

* Дисциплина (модуль) предусматривает применение электронного обучения, дистанционных образовательных технологий

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием количества часов и видов занятий

№	Раздел (тема) дисциплины и краткое содержание	Формируемые компетенции, индикаторы	очная форма			
			Контактная работа обучающихся с преподавателем /из них в форме практической подготовки, часов			Самостоятельная работа, часов
			Лекции	Практические занятия	Лабораторные работы	
1	Тема 1. Качество кода и рефакторинг (SonarQube, ESLint) 1. Роль статического анализа в CI/CD пайплайне (прекоммит-хуки, этап сборки). 2. Настройка Quality Gate в SonarQube для блокировки поставки при падении качества. 3. ESLint в пайплайне: автоматическая проверка и фикс стиля кода. 4. Выявление «запахов кода» и технического долга как артефактов контроля качества. 5. Интеграция проверок безопасности (SAST) в конвейер — ворота безопасности. 6. Формирование отчётов об артефактах качества (отчёт Sonar, выводы линтера).	ИД-1 _{ПК-1} ИД-2 _{ПК-1}	2		2	10

2	Тема 2. SOLID принципы 1. Как SOLID влияет на тестируемость модулей в автоматизированных тестах (unit test stage в CI). 2. DIP и внедрение зависимостей — удобство подмены компонентов при интеграционном тестировании. 3. OCP и расширение функциональности без полной пересборки всего конвейера. 4. LSP и надёжность замены компонентов в пайплайне развёртывания. 5. SRP — разделение ответственности для параллельных стадий CI/CD (build, test, deploy). 6. Связь SOLID с управлением артефактами сборки		2		2	10
3	Тема 3. Паттерны проектирования GoF 1. Паттерн «Стратегия» для смены стратегий тестирования или развёртывания в CI/CD. 2. Паттерн «Наблюдатель» для событий пайплайна (уведомления, триггеры последующих этапов). 3. «Строитель» для конфигурирования сложных конвейеров (declarative pipeline). 4. «Цепочка обязанностей» для последовательных стадий: commit → test → build → deploy. 5. «Фабрика» для создания артефактов сборки в зависимости от окружения. 6. Антипаттерны конвейеров: монолитный pipeline, пропуск проверок.		2		2	10
4	Тема 4. Чистая архитектура (Hexagonal, Onion) 1. Чистая архитектура как способ легко подменять окружения в конвейере (test/staging/prod). 2. Понятие портов и адаптеров — замена БД, очереди, API без правки core-модуля. 3. Как это упрощает конфигурацию пайплайна (разные конфиги для разных стадий). 4. Интеграционные тесты в CI против адаптеров, а не всей системы. 5. Разделение слоёв для параллельной сборки (domain, infrastructure, application). 6. Влияние на управление артефактами (разные версии адаптеров).	ИД-1 _{ПК-1}	2		2	10

5	Тема 5. Функциональное программирование 1. Чистые функции и иммутабельность — почему важно для воспроизводимых сборок. 2. Отсутствие побочных эффектов на этапе тестирования (стабильные результаты). 3. Композиция функций для построения этапов пайплайна (build → test → scan → deploy). 4. Монадическая обработка ошибок в скриптах конвейера без прерывания всей цепочки. 5. Использование неизменяемых артефактов сборки (контейнеры — тоже иммутабельность). 6. Функциональные подходы в Jenkins Pipeline / GitLab CI (DSL).	ИД-5 _{ПК-1}	2		2	10
6	Тема 6. Асинхронное программирование (async/await, Rx) 1. Асинхронный запуск этапов пайплайна (параллельные тесты, сборки). 2. Обработка таймаутов и retry в асинхронных задачах конвейера. 3. Поток событий (Rx) для мониторинга прогресса пайплайна в реальном времени. 4. Async/await для управления последовательностью стадий с зависимостями. 5. Backpressure в pipeline при лимитах ресурсов системы CI. 6. Паттерн Fallback при сбое асинхронного этапа поставки.	ИД-5 _{ПК-1}	2		2	10

7	Тема 7. Многопоточность и параллелизм 1. Параллельный запуск тестов (unit, integration, e2e) — сокращение времени конвейера. 2. Потокбезопасное управление общими ресурсами (кэши, артефакты). 3. Параллелизм и конкурентность в раннерах CI (GitLab runners, Jenkins agents). 4. Обработка состояния stages при многопоточном выполнении (проблемы гонок). 5. Использование пулов потоков для стадий деплоя на множество узлов. 6. Deadlock в конвейере: взаимоблокировка стадий — обнаружение и профилактика.	ИД-5 _{ПК-1}	2		2	10
8	Тема 8. Архитектура корпоративных приложений (DDD, CQRS) 1. Разделение команд и запросов (CQRS) как модель для разных пайплайнов. 2. Event Sourcing и управление версиями событий — задача для миграций в CI/CD. 3. Ограниченные контексты как границы независимых конвейеров поставки. 4. Как DDD определяет набор микросервисов → множественные пайплайны. 5. Автоматизация тестирования агрегатов и доменных событий. 6. Артефакты в CQRS-системе: слепки (snapshots), события (domain events), команды.	ИД-2 _{ПК-1} ИД-4 _{ПК-1} ИД-6 _{ПК-1}	2			10

9	Тема 9. CI/CD как экосистема: безопасность, артефакты, наблюдаемость конвейера 1. Управление артефактами в пайплайне: бинарные репозитории (Nexus, Artifactory), версионирование, хранение метаданных. 2. Автоматизированная проверка безопасности в CI/CD (SAST, DAST, сканирование зависимостей, контейнеров). 3. Политики безопасности как код (OPA) в конвейере — пример интеграции. 4. Наблюдаемость пайплайна: метрики, логи, трейсы этапов поставки (Observability самого CI/CD). 5. Взаимодействие CI/CD с системами мониторинга деплоя (Prometheus, Grafana) — обратная связь в пайплайн. 6. GitOps как модель управления артефактами и конфигурациями через CI/CD (ArgoCD, Flux). 7. Создание «безопасного пайплайна по умолчанию»: шаблон с SAST, тестами, сканированием состава ПО (SBOM).	ИД-3 _{ПК-1} ИД-4 _{ПК-1} ИД-6 _{ПК-1} ИД-7 _{ПК-1} ИД-8 _{ПК-1}				
	Зачет с оценкой		2		2	16
	ИТОГО за 2 семестр		16		32	96
	ИТОГО		16		32	96

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств (ФОС) по дисциплине (модулю) базируется на перечне осваиваемых компетенций с указанием индикаторов. ФОС обеспечивает объективный контроль достижения запланированных результатов обучения. ФОС включает в себя:

- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций (включаются в методические указания по тем видам работ, которые предусмотрены учебным планом и предусматривают оценку сформированности компетенций);
- типовые оценочные средства, необходимые для оценки знаний, умений и уровня сформированности компетенций.

ФОС является приложением к данной программе дисциплины.

7. Методические указания для обучающихся по освоению дисциплины

Приступая к работе, каждый студент должен принимать во внимание следующие положения.

Дисциплина построена по тематическому принципу, каждая тема представляет собой логически завершённый раздел.

Практические занятия проводятся с целью закрепления усвоенной информации,

приобретения навыков ее применения при решении практических задач в соответствующей предметной области.

Самостоятельная работа студентов направлена на самостоятельное изучение дополнительного материала, подготовку к лабораторным занятиям, а также выполнения всех видов самостоятельной работы.

Для успешного освоения дисциплины, необходимо выполнить все виды самостоятельной работы, используя рекомендуемые источники информации.

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

8.1.1. Перечень основной литературы:

1. Уилсон, К. Грокаем Continuous Delivery / К. Уилсон. — Санкт-Петербург : Питер, 2024. — 320 с. — ISBN 978-5-4461-2325-2.
2. Кольцов, Д. М. Справочник Python : кратко, быстро, под рукой / Д. М. Кольцов, Е. В. Дубовик. — 2-е изд., испр. и перераб. — Санкт-Петербург : Наука и Техника, 2023. — 300 с. — ISBN 978-5-94387-717-9.
3. Чернышев, С. А. Python: продвинутый уровень : учебное пособие / С. А. Чернышев. — Санкт-Петербург : ГУАП, 2024. — 180 с

8.1.2. Перечень дополнительной литературы:

1. Мартин, Р. С. Чистый код: создание, анализ и рефакторинг / Р. С. Мартин. — Санкт-Петербург : Питер, 2022. — 464 с. — ISBN 978-5-4461-0960-9.
2. Фаулер, М. Рефакторинг: улучшение существующего кода / М. Фаулер. — Санкт-Петербург : Питер, 2020. — 432 с. — ISBN 978-5-4461-1168-6.
3. Блум, Р. PostgreSQL. Руководство администратора / Р. Блум. — Москва : ДМК Пресс, 2023. — 368 с. — ISBN 978-5-93700-154-2.
4. Файн, Я. DevOps для разработчиков / Я. Файн. — Санкт-Петербург : Питер, 2022. — 384 с. — ISBN 978-5-4461-2014-5. Савич, В. В. Контейнеризация и оркестрация приложений / В. В. Савич. — Москва : ДМК Пресс, 2023. — 216 с. — ISBN 978-5-93700-170-2.
5. Савич, В. В. Контейнеризация и оркестрация приложений / В. В. Савич. — Москва : ДМК Пресс, 2023. — 216 с. — ISBN 978-5-93700-170-2.

8.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

1. Методические указания по выполнению лабораторных работ по дисциплине «Продвинутая разработка программного обеспечения». — Ставрополь : СКФУ, 2026. — 112 с. (размещены в локальной сети университета и в ЭИОС СКФУ)
2. Конспект лекций по дисциплине «Продвинутая разработка программного обеспечения». — Ставрополь : СКФУ, 2026. (электронное издание)
3. Фонд оценочных средств по дисциплине «Продвинутая разработка программного обеспечения». — Ставрополь : СКФУ, 2026. (электронное издание)

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

№	Ресурс	Адрес	Назначение
1	SourceCraft (Яндекс)	https://sourcecraft.dev/portal/docs/ru	CI/CD

№	Ресурс	Адрес	Назначение
	официальная документация		платформа, встроенный механизм пайплайнов
2	GitFlic официальная документация	https://docs.gitflic.ru	Российская Git-платформа, Container Registry, CI/CD
3	SonarQube официальная документация	https://docs.sonarqube.org	Статический анализ кода, Quality Gate
4	Docker официальная документация (рус.)	https://docs.docker.com/get-started/	Контейнеризация, Dockerfile, Compose
5	Prometheus официальная документация	https://prometheus.io/docs	Сбор метрик, мониторинг
6	Grafana официальная документация	https://grafana.com/docs/	Визуализация дашбордов
7	Argo CD официальная документация	https://argo-cd.readthedocs.io	GitOps, синхронизация Kubernetes
8	Microsoft Learn / SonarQube SSO	https://learn.microsoft.com/ru-ru/entra/identity/saas-apps/sonarqube-tutorial	Интеграция SonarQube с SSO
9	Электронно-библиотечная система «Лань»	https://e.lanbook.com	Доступ к учебной литературе
10	Научная библиотека СКФУ	https://find.ncfu.ru	Доступ к печатным и электронным изданиям вуза
11	Habr — статьи по Kubernetes и GitOps	https://habr.com/ru/	Публикации по DevOps (например, Argo CD)

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При чтении лекций используется компьютерная техника, демонстрации презентационных мультимедийных материалов. На семинарских и практических занятиях студенты представляют презентации, подготовленные ими в часы самостоятельной работы.

Информационные справочные системы:

Информационно-справочные и информационно-правовые системы, используемые при изучении дисциплины:

1. Официальный интернет-портал правовой информации (www.pravo.gov.ru)
2. Справочно-правовая система «КонсультантПлюс»
3. Электронно-библиотечная система «Лань» (e.lanbook.com)

Программное обеспечение:

1.	Python 3.9+	Язык программирования для разработки приложений и скриптов автоматизации	Свободное (лицензия PSF)
2.	Git	Система контроля версий, управление исходным кодом	Свободное (GPL)
3.	Docker Desktop / Docker Engine	Контейнеризация приложений, изолированное тестирование	Бесплатная версия для образования
4.	Visual Studio Code	Среда разработки, редактор кода	Свободное (MIT)
5.	SonarQube Community Edition	Статический анализ кода, управление качеством	Свободное (LGPL)
6.	SourceCraft (Яндекс)	CI/CD платформа, система управления	Отечественная разработка, доступна

		репозиториями, container registry	на https://sourcecraft.dev
7.	GitFlic	Российская Git-платформа с поддержкой CI/CD, Container Registry и управления артефактами	Отечественная разработка, доступна на https://gitflic.ru
8.	GitLab CE	Система CI/CD с открытым исходным кодом	Свободное (MIT)
9.	Jenkins	Сервер автоматизации для CI/CD пайплайнов	Свободное (MIT)
10.	Argo CD	Инструмент GitOps для синхронизации Kubernetes кластера с репозиторием	Свободное (Apache 2.0)
11.	pytest	Фреймворк для unit-тестирования на Python	Свободное (MIT)
12.	pytest-cov	Измерение покрытия кода тестами	Свободное (MIT)
13.	Bandit	Инструмент статического анализа безопасности (SAST) для Python	Свободное (Apache 2.0)
14.	Flake8	Линтер для проверки стиля кода Python	Свободное (MIT)
15.	SonarQube Scanner	Клиент для интеграции SonarQube в CI/CD	Свободное (LGPL)

16.	Minikube	Локальный Kubernetes кластер для тестирования	Свободное (Apache 2.0)
17.	kubectl	Утилита управления Kubernetes кластером	Свободное (Apache 2.0)
18.	Helm	Пакетный менеджер для Kubernetes	Свободное (Apache 2.0)
19.	Docker Compose	Оркестрация многоконтейнерных приложений	Свободное (Apache 2.0)
20.	Prometheus	Система сбора и хранения метрик	Свободное (Apache 2.0)
21.	Grafana	Визуализация метрик, построение дашбордов	Свободное (AGPL)
22.	k6	Инструмент нагрузочного тестирования	Свободное (AGPL)
23.	Sonatype Nexus OSS	Бинарный репозиторий для хранения артефактов	Свободное (Eclipse Public License)
24.	HashiCorp Vault	Управление секретами, безопасное хранение паролей и ключей	Свободное (Business Source License)
25.	Ubuntu 20.04+ / Debian 11+	Базовая ОС для выполнения лабораторных работ (рекомендуется)	Свободное (GPL/Linux Foundation)
26.	Windows 10/11 с WSL2	Альтернативная платформа для	Коммерческая (по лицензии ОС)

		выполнения работ	
27.	Astra Linux (Special Edition)	Отечественная ОС (для специализации на импортозамещение)	Доступно для образования

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Лабораторные работы	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Самостоятельная работа	Помещение для самостоятельной работы обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и возможностью доступа к электронной информационно-образовательной среде университета

11. Особенности освоения дисциплины (модуля) лицами с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины (модуля) обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
- письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,
- специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),
- индивидуальное равномерное освещение не менее 300 люкс,
- при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

- обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата (в том числе с тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

- по желанию студента задания могут выполняться в устной форме.

12. Особенности реализации дисциплины с применением дистанционных образовательных технологий и электронного обучения

Согласно части 1 статьи 16 Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» под *электронным обучением* понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников. Под *дистанционными образовательными технологиями* понимаются образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

Реализация дисциплины может быть осуществлена с применением дистанционных образовательных технологий и электронного обучения полностью или частично. Компоненты УМК дисциплины (рабочая программа дисциплины, оценочные и методические материалы, формы аттестации), реализуемой с применением дистанционных образовательных технологий и электронного обучения, содержат указание на их использование.

При организации образовательной деятельности с применением дистанционных образовательных технологий и электронного обучения могут предусматриваться асинхронный и синхронный способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет».

При применении дистанционных образовательных технологий и электронного обучения в расписании по дисциплине указываются: способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет» (ВКС-видеоконференцсвязь, ЭТ – электронное тестирование); ссылки на электронную информационно-образовательную среду СКФУ, на образовательные платформы и ресурсы иных организаций, к которым предоставляется открытый доступ через информационно-телекоммуникационную сеть «Интернет»; для синхронного обучения - время проведения онлайн-занятий и преподаватели; для асинхронного обучения - авторы онлайн-курсов.

При организации промежуточной аттестации с применением дистанционных образовательных технологий и электронного обучения используются Методические рекомендации по применению технических средств, обеспечивающих объективность результатов при проведении промежуточной и государственной итоговой аттестации по

образовательным программам высшего образования - программам бакалавриата, программам специалитета и программам магистратуры с применением дистанционных образовательных технологий (Письмо Минобрнауки России от 07.12.2020 г. № МН-19/1573-АН "О направлении методических рекомендаций").

Реализация дисциплины с применением электронного обучения и дистанционных образовательных технологий осуществляется с использованием электронной информационно-образовательной среды СКФУ, к которой обеспечен доступ обучающихся через информационно-телекоммуникационную сеть «Интернет», или с использованием ресурсов иных организаций, в том числе платформ, предоставляющих сервисы для проведения видеоконференций, онлайн-встреч и дистанционного обучения (МТС-Линк), а также с использованием возможностей социальных сетей для осуществления коммуникации обучающихся и преподавателей.

Учебно-методическое обеспечение дисциплины, реализуемой с применением электронного обучения и дистанционных образовательных технологий, включает представленные в электронном виде рабочую программу, учебно-методические пособия или курс лекций, методические указания к выполнению различных видов учебной деятельности обучающихся, предусмотренных дисциплиной, и прочие учебно-методические материалы, размещенные в информационно-образовательной среде СКФУ.