

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 14:50:30
Уникальный программный ключ:
990bea3aeec8c23bd8699e48288f8d1960c600

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.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Автоматизация тестирования и качество ПО

Направление подготовки	09.04.04 Программная инженерия
Направленность (профиль)	Разработка, мониторинг и управление жизненным циклом инженерных систем
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	1
Разработано	
<u>К.ф.-м.н., доцент, заведующая межинститутской базовой кафедрой</u>	
<u>Новикова Елена Николаевна</u>	

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины: формирование у обучающихся системных знаний и практических навыков в области стратегий, методов и инструментов автоматизированного тестирования для обеспечения качества сложных программных и программно-аппаратных (инженерных) систем, а также их эффективной интеграции в сквозные конвейеры CI/CD.

Задачи освоения дисциплины:

1. Изучить современные подходы к обеспечению качества ПО (QA) и место автоматизированного тестирования в жизненном цикле разработки.
2. Освоить различные уровни и типы тестирования (модульное, интеграционное, системное, регрессионное, нагрузочное, E2E) и их роль в CI/CD.
3. Сформировать навыки разработки модульных и интеграционных тестов с использованием популярных фреймворков (pytest, JUnit, Jest).
4. Изучить методики анализа и интерпретации метрик качества кода: покрытие тестами, сложность, запахи кода.
5. Научиться реализовывать автоматизированные сценарии тестирования API (REST, GraphQL) и пользовательских интерфейсов (UI).
6. Освоить методы нагрузочного и стресс-тестирования для оценки производительности и устойчивости систем.
7. Изучить инструменты статического (SAST) и динамического (DAST) анализа безопасности кода.
8. Сформировать навыки интеграции всех видов автоматизированных тестов в CI/CD-пайплайн с настройкой качественных ворот (Quality Gates).

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Автоматизация тестирования и качество ПО» относится к части, формируемой участниками образовательных отношений (вариативная часть) и является дисциплиной по выбору (Б1.В.ДВ.01.02).

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	ИД-1 ПК-1. Проектирует архитектуру CI/CD-конвейеров, определяя этапы сборки, тестирования и развертывания для различных типов приложений и платформ.	Способен проектировать архитектуру CI/CD-пайплайна, определяя оптимальную последовательность и состав этапов тестирования (модульного, интеграционного, E2E, нагрузочного, безопасности) на основе пирамиды тестирования и специфики приложения. Анализирует требования к качеству ПО и на их основе обосновывает выбор стратегии тестирования и целевых метрик качества для каждого этапа пайплайна. Исследует влияние различных факторов (тип приложения, архитектура, критичность сервиса) на структуру тестового конвейера.

автоматизированного тестирования, проверки безопасности, управления артефактами и конфигурациями;	ИД-2 ПК-1. Реализует автоматизированные процессы непрерывной интеграции, включая компиляцию кода, запуск модульных и интеграционных тестов, статический анализ.	азрабатывает и отлаживает модульные и интеграционные тесты с использованием фреймворков pytest, JUnit, Jest, применяя фикстуры, моки и параметризацию. Настраивает инструменты статического анализа безопасности (Semgrep, Bandit) и интерпретирует их отчёты для выявления уязвимостей на ранних этапах. Реализует автоматизированные сценарии тестирования REST API (pytest + requests, Postman/Newman) и организует их выполнение в CI/CD-среде. Способен писать сценарии нагрузочного тестирования на k6 (smoke, spike, stress) и анализировать метрики производительности (RPS, P95, P99, error rate).
	ИД-3 ПК-1. Интегрирует инструменты безопасности (SAST, DAST, SCA) в конвейеры CI/CD для автоматического сканирования уязвимостей на ранних этапах разработки.	Настраивает автоматический запуск SAST (Semgrep, Bandit) и DAST (OWASP ZAP) в CI/CD-пайплайне с генерацией структурированных отчётов (JSON, SARIF, JUnit). Интегрирует Quality Gate по безопасности, при котором пайплайн блокируется при обнаружении HIGH/CRITICAL уязвимостей. Классифицирует выявленные уязвимости по CWE и CVSS, анализирует ложноположительные срабатывания и применяет механизмы их подавления (nosemgrep, контексты ZAP). Сравнивает результаты SAST и DAST, формулирует выводы о комплементарности методов для формирования целостной стратегии DevSecOps.
	ИД-7 ПК-1. Оценивает эффективность работы CI/CD-конвейеров и оптимизирует их для сокращения времени цикла разработки и повышения надежности развертываний.	Настраивает качественные ворота (Quality Gates) на основе метрик покрытия кода ($\geq 80\%$), успешности прохождения тестов (100%) и производительности (P95 < SLA). Оценивает эффективность тестового пайплайна с использованием метрик DORA (частота развертываний, время выполнения, частота отказов). Оптимизирует время выполнения пайплайна за счёт параллелизации тестов, шардирования (test splitting), кэширования зависимостей и условного выполнения этапов. Анализирует нестабильные тесты (flaky tests) и разрабатывает меры по повышению стабильности и надёжности

		пайплайна.
--	--	------------

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 4 з.е. 144 ак.ч.	ОФО, в астр. часах
Контактная работа:	32
Лекции/из них практическая подготовка	16
Лабораторных работ/из них практическая подготовка	16
Практических занятий/из них практическая подготовка	
Самостоятельная работа	76
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 и 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. Основы автоматизации тестирования и качество ПО. 1.1. Понятие качества ПО. Роль QA и тестирования в DevOps-культуре. 1.2. Место автоматизированных тестов в CI/CD-пайплайне. 1.3. Пирамида, "мороженое" и "шампанское" тестирования: соотношение unit, integration, E2E тестов. 1.4. Классификация тестов по уровням (модульные, интеграционные, системные, приемочные), целям (функциональные, нефункциональные) и результату (позитивные, негативные). 1.5. Обзор инструментов автоматизации: pytest, JUnit, Selenium, k6, Postman, OWASP ZAP, Semgrep.	ИД-1 _{ПК-1}	2		2	5

2	Тема 2. Модульное тестирование и анализ покрытия кода. 2.1. Принципы модульного тестирования: изоляция, скорость, повторяемость (FIRST). 2.2. Структура теста (паттерн AAA: Arrange-Act-Assert). 2.3. Фреймворки для разработки модульных тестов (pytest для Python, JUnit для Java). 2.4. Фикстуры (pytest.fixture), моки (unittest.mock), параметризация (pytest.mark.parametrize). 2.5. Метрики покрытия кода: строки, ветви, функции. Инструменты coverage.py, pytest-cov, JaCoCo, Istanbul. 2.6. Настройка порогов покрытия и практики TDD (Test-Driven Development).	ИД1 _{ПК-1} ИД2 _{ПК-1}	2		2	5
3	Тема 3. Интеграционное тестирование REST API. 3.1. Цели интеграционного тестирования: проверка взаимодействия компонентов. 3.2. Тестирование REST API: HTTP-методы, статус-коды, форматы JSON/XML. 3.3. Инструменты API-тестирования: pytest + requests, Postman, Newman, REST Assured. 3.4. Создание коллекций Postman: пре-запросы (Pre-request Script), пост-ответы (Post-response), переменные окружения. 3.5. Data-Driven тестирование через CSV/JSON. 3.6. Запуск коллекций через Newman в CI/CD с генерацией отчётов (JUnit XML, HTML).	ИД1 _{ПК-1} ИД2 _{ПК-1}	2		2	10

4	Тема 4. Автоматизация тестирования UI и E2E-сценариев. 4.1. End-to-End (E2E) тестирование пользовательских сценариев. 4.2. Паттерн Page Object Model (ПОМ) для структурирования UI-тестов. Инкапсуляция локаторов и действий. 4.3. Паттерн Component Object для повторяющихся элементов (Header, Footer). 4.4. Обзор инструментов UI-автоматизации: Selenium WebDriver, Cypress, Playwright. 4.5. Автоожидания (auto-waiting), работа с динамическими элементами. 4.6. Проблемы E2E-тестов: флаки-тесты (flaky tests), скорость, стоимость поддержки.	ИД1 _{ПК-1} ИД2 _{ПК-1}	2		2	10
5	Тема 5. Нагрузочное и стресс-тестирование. 5.1. Цели нагрузочного тестирования: определение точки насыщения, проверка стабильности, выявление узких мест. 5.2. Типы нагрузочного тестирования: smoke, load, spike, stress, soak, breakpoint. 5.3. Ключевые метрики производительности: RPS (запросов/сек), latency (p50, p95, p99), error rate, количество активных пользователей. 5.4. Инструмент k6 (Grafana Labs): написание сценариев на JavaScript, модель исполнения (closed vs open). 5.5. Настройка порогов (thresholds) для автоматического Pass/Fail теста. 5.6. Анализ результатов и поиск точки насыщения системы.	ИД2 _{ПК-1} ИД7 _{ПК-1}	2		2	10

6	Тема 6. Статический и динамический анализ безопасности (SAST / DAST). 6.1. Концепция "Shift Left" для безопасности. 6.2. Статический анализ безопасности кода (SAST): принципы работы (AST-матчинг, паттерны), обзор инструментов (Semgrep, Bandit, CodeQL). 6.3. Типовые уязвимости, обнаруживаемые SAST: SQLi, XSS, hardcoded secrets, eval, pickle. 6.4. Классификация уязвимостей по CWE и CVSS. 6.5. Динамический анализ безопасности (DAST): принципы работы ("чёрный ящик"), инструмент OWASP ZAP (Baseline Scan, Full Scan, Spider, Active Scan). 6.6. Сравнение SAST и DAST: методологии, выявляемые уязвимости, ложноположительные срабатывания, комплементарность.	ИД2_{ПК-1} ИД3_{ПК-1}	2		2	10
7	Тема 7. Интеграция тестирования в CI/CD. Качественные ворота (Quality Gates). 7.1. Стратегии запуска тестов в пайплайне: последовательно, параллельно, по матрице. 7.2. Качественные ворота (Quality Gates): автоматическое принятие решения об остановке пайплайна. 7.3. Настройка порогов для покрытия кода ($\geq 80\%$), успешности тестов (100%), уязвимостей высокого риска (0 HIGH/CRITICAL), времени ответа ($P95 < SLA$). 7.4. Форматы отчётов о тестировании: JUnit XML, Cobertura XML, SARIF. 7.5. Публикация отчётов как артефактов пайплайна. 7.6. Демонстрация срабатывания Quality Gates (успешный пайплайн vs заблокированный).	ИД1_{ПК-1} ИД3_{ПК-1} ИД7_{ПК-1}	2		2	10

8	Тема 8. Тестирование асинхронных систем и управление тестовым окружением. 8.1. Особенности тестирования асинхронных систем (очереди сообщений). 8.2. Ключевые понятия AMQP (RabbitMQ): Producer, Consumer, Exchange, Queue, Binding, ACK/NACK, Dead Letter Queue (DLQ). 8.3. Стратегии обработки ошибок: retry с exponential backoff, DLQ, идемпотентность Consumer'а. 8.4. Библиотека Testcontainers: поднятие реальных зависимостей (БД, очереди) в Docker из тестов. 8.5. Тестирование с network disruption (toxiproxy). 8.6. Сравнение подходов к тестированию очередей: мок-брокер vs эмбедед vs Testcontainers.	ИД2 _{ПК-1} ИД7 _{ПК-1}				
	Зачет с оценкой					
	ИТОГО за 1 семестр		16		16	76
	ИТОГО		16		16	76

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. Лотт, С. Объектно-ориентированный Python, 4-е изд. / С. Лотт, Д. Филлипс. — СПб. : Питер, 2024. — 704 с. — ISBN 978-5-4461-1995-0
2. Тюлькин, И. В. Тестирование, автоматизация и процессы на экспертном уровне : руководство / И. В. Тюлькин. — М. : ДМК Пресс, 2025. — 184 с. — ISBN 978-5-93700-401-7.
3. Игнатьев, А. В. Тестирование программного обеспечения : учебное пособие для вузов / А. В. Игнатьев. — 3-е изд., стер. — СПб. : Лань, 2023. — 56 с. — ISBN 978-5-507-45425-9.
4. Василькова, А. Н. Технологии оценки качества программного обеспечения : пособие / А. Н. Василькова, А. В. Воробей, О. С. Медведев, А. М. Прудник ; под ред. Т. В. Казак. — Минск : БГУИР, 2025. — 85 с.
5. Старолетов, С. М. Основы тестирования программного обеспечения. Учебное пособие для СПО. 5-е изд., стереотипное / С. М. Старолетов. — СПб. : Лань, 2025. — 190 с. — ISBN 978-5-507-53820-1.

8.1.2. Перечень дополнительной литературы:

1. Окран, М. pytest: руководство по эффективному тестированию на Python / М. Окран. — М. : ДМК Пресс, 2022. — 368 с. — ISBN 978-5-97060-987-4.
2. Уинтерс, Д. Selenium WebDriver: практическое руководство по автоматизации тестирования / Д. Уинтерс. — М. : ДМК Пресс, 2021. — 256 с. — ISBN 978-5-97060-913-3.
3. Криспин, Л. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд / Л. Криспин, Д. Грегори. — М. : Вильямс, 2021. — 592 с. — ISBN 978-5-907144-56-5.
4. Сундарам, А. Playwright для автоматизации тестирования / А. Сундарам. — М. : ДМК Пресс, 2023. — 284 с. — ISBN 978-5-93700-230-9.
5. Кулаков, С. М. Основы тестирования и верификации программного обеспечения : учебное пособие / С. М. Кулаков. — М. : ИНФРА-М, 2022. — 304 с. — ISBN 978-5-16-017583-6.

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

1. Методические указания по выполнению лабораторных работ по дисциплине «Автоматизация тестирования и качество ПО» (2026). Содержат цели, задачи, теоретическое обоснование, варианты индивидуальных заданий (базовый, средний, повышенный уровни), контрольные вопросы, требования к отчёту и критерии оценивания для каждой из 8 лабораторных работ.
2. Методические указания по организации и проведению самостоятельной работы по дисциплине «Автоматизация тестирования и качество ПО» (2026). Включают темы для самостоятельного изучения, рекомендации по работе с литературой, порядок выполнения сквозного проекта, вопросы для собеседования и критерии оценивания компетенций.
3. Фонд оценочных средств по дисциплине «Автоматизация тестирования и качество ПО» (2026). Содержит типовые контрольные задания, тесты,

вопросы к экзамену/зачёту, а также описание показателей и критериев оценивания компетенций.

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

www.biblioclub.ru - Электронная библиотечная система «Университетская библиотека онлайн»

www.eLibrary.ru - Научная электронная библиотека

www.iprbookshop.ru - Электронно-библиотечная система IPRbooks

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. www.biblioclub.ru — Электронная библиотечная система «Университетская библиотека онлайн».
2. www.iprbookshop.ru — Электронно-библиотечная система IPRbooks.
3. www.elibrary.ru — Научная электронная библиотека eLIBRARY.RU.
4. e.lanbook.com — Электронно-библиотечная система издательства «Лань».
5. znanium.com — Электронно-библиотечная система Znanium (издательства «ИНФРА-М», «Вузовский учебник» и др.).
6. habr.com/ru/hub/testing/ — Хаб «Тестирование IT-систем» на Habr (статьи, кейсы, обсуждения инструментов, best practices).
7. habr.com/ru/hub/devops/ — Хаб «DevOps» (практики CI/CD, качество ПО, автоматизация развертывания).
8. protesting.ru — Портал «Протестинг» (библиотека знаний по тестированию ПО, форум, вебинары).
9. software-testing.ru — Русскоязычное сообщество тестировщиков (статьи, курсы, конференции).
10. docs.pytest.org — Официальная документация фреймворка модульного тестирования pytest.
11. playwright.dev — Официальная документация инструмента автоматизации UI-тестирования Playwright.
12. k6.io/docs — Официальная документация инструмента нагрузочного тестирования k6.
13. zaproxy.org/docs — Официальная документация инструмента динамического анализа безопасности OWASP ZAP.
14. semgrep.dev/docs — Официальная документация статического анализатора безопасности Semgrep.
15. docs.gitlab.com/ee/ci — Официальная документация GitLab CI/CD.
16. docs.github.com/ru/actions — Официальная документация GitHub Actions.
17. testcontainers.com — Официальный сайт и документация библиотеки Testcontainers.
18. www.iso.org — Международная организация по стандартизации (доступ к стандартам качества ПО, например ISO 25010).

19. www.omg.org — Object Management Group (стандарты UML, моделирования процессов).
20. www.sqa.ru — Портал «Вопросы обеспечения качества программного обеспечения» (библиотека, статьи, ссылки).
21. owasp.org — Официальный сайт OWASP (документация, Top 10 уязвимостей, инструменты, руководства).
22. cwe.mitre.org — База данных Common Weakness Enumeration (типы уязвимостей, классификация).
23. www.terraform.io/docs — Официальная документация Terraform (как источник сведений об инфраструктуре как код).
24. docs.ansible.com — Официальная документация Ansible (автоматизация конфигураций).
25. argo-cd.readthedocs.io — Официальная документация Argo CD (GitOps).
26. stepik.org — Платформа онлайн-образования (курсы по тестированию ПО, автоматизации на Python, pytest).
27. www.learntesting.ru — Портал «Учимся тестированию» (бесплатные уроки, учебные материалы, книги).
28. otus.ru — Портал онлайн-курсов OTUS (открытые вебинары и записи курсов по автоматизации тестирования).

Программное обеспечение:

1. Git - система контроля версий (Open Source, GPL v2).
2. Python 3.x - язык программирования для написания тестов и тестируемых приложений (Open Source, PSF License).
3. pytest - фреймворк для модульного тестирования на Python (Open Source, MIT).
4. pytest-cov - плагин для измерения покрытия кода (Open Source, MIT).
5. coverage.py - инструмент для измерения покрытия кода Python (Open Source, Apache 2.0).
6. Postman - инструмент для ручного и автоматизированного тестирования API (Freemium).
7. Newman - CLI-запускающий коллекций Postman (Open Source, Apache 2.0).
8. Playwright - инструмент для автоматизации UI-тестирования (Open Source, Apache 2.0).
9. Selenium WebDriver - классический инструмент UI-автоматизации (Open Source, Apache 2.0).
10. k6 - инструмент нагрузочного и стресс-тестирования (Open Source, AGPL v3).
11. Semgrep (OSS-версия) - статический анализатор безопасности кода (SAST) (Open Source, LGPL 2.1).
12. Bandit - SAST-инструмент, специализированный для Python (Open Source, Apache 2.0).
13. OWASP ZAP - инструмент динамического анализа безопасности (DAST) (Open Source, Apache 2.0).
14. Testcontainers - библиотека для поднятия реальных зависимостей (БД, очереди) в Docker из тестов (Open Source, MIT).
15. Docker / Docker Compose - платформа для контейнеризации тестового окружения (базовый функционал - Open Source, Apache 2.0).
16. RabbitMQ - брокер сообщений для тестирования асинхронных систем (Open Source, MPL 2.0).
17. pytest-html - плагин для генерации HTML-отчётов о прохождении тестов (Open

- Source, MIT).
18. Allure Framework - фреймворк для формирования наглядных отчётов о тестировании (Open Source, Apache 2.0).
 19. Visual Studio Code - среда разработки (Free, Microsoft).
 20. GitLab CE - DevOps-платформа для CI/CD, репозитория и реестра артефактов (self-hosted, Open Source, MIT).
 21. GitFlic - российская платформа для хостинга кода и CI/CD (в Реестре отечественного ПО, Free для малых команд).
 22. GitVerse - российская DevOps-среда с CI/CD и хранилищем артефактов (российская разработка, бесплатно с квотами).
 23. Mos.hub - российская облачная платформа для совместной разработки с CI/CD (бесплатно).
 24. SASTAV - российский статический анализатор кода (в Реестре отечественного ПО, коммерческая / ознакомительная версия).
 25. AppSec.Wave - российский высокоскоростной SAST-анализатор (российская разработка, коммерческая).

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.