

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 16:00:03
Уникальный программный ключ:
990bea3aeec848c23bd8699e48288f8d1960c600

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
А.А. Порохня

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Конструирование программного обеспечения

Направление подготовки
Направленность (профиль)

Год начала обучения
Форма обучения
Реализуется в семестре

09.03.04 Программная инженерия
Разработка и сопровождение
программного обеспечения
2026
очная
6

Разработано
Заведующая межинститутской базовой
кафедрой департамента цифровых,
робототехнических систем и электроники
(должность разработчика)
Новикова Елена Николаевна
Ф.И.О.

Ставрополь 2026 г.

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины

Сформировать у студентов практические навыки промышленного конструирования программного обеспечения высокого качества, включая:

- продвинутое тестирование (интеграционное, нагрузочное, профилирование);
- рефакторинг и управление техническим долгом;
- безопасное программирование (JWT, OAuth2, защита от атак);
- реактивное и асинхронное программирование;
- работу с альтернативными интерфейсами (GraphQL, gRPC) и NoSQL;
- автоматизацию качества через CI/CD и мониторинг.

Задачи освоения дисциплины:

1. Освоить продвинутый тест-дизайн (классы эквивалентности, граничные значения, pairwise) и TDD для сложных сценариев.
2. Сформировать навыки интеграционного тестирования (TestContainers) и нагрузочного тестирования (JMeter/k6, профилирование).
3. Научиться выявлять «запахи кода», оценивать технический долг и выполнять безопасный рефакторинг.
4. Освоить реализацию JWT (access/refresh), OAuth2, RBAC, защиту от CSRF/XSS/SQLi и хеширование BCrypt.
5. Научиться оптимизировать ORM-запросы (устранение N+1, стратегии загрузки) и управлять транзакциями и миграциями.
6. Освоить асинхронное и реактивное программирование (CompletableFuture, Reactor, WebFlux, backpressure).
7. Научиться проектировать GraphQL-схемы, использовать DataLoader, а также работать с protobuf и gRPC.
8. Сформировать навыки работы с NoSQL (MongoDB, Neo4j) и полиглотным хранением данных.
9. Освоить статический анализ кода (SonarQube) и внедрение проверок в CI/CD (GitHub Actions / GitLab CI).
10. Научиться настраивать мониторинг (Prometheus, Grafana), логирование (ELK/Loki) и алерты.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Конструирование программного обеспечения» относится к части, формируемой участниками образовательных отношений.

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-3 Владеет навыками разработки программного обеспечения: способен реализовывать алгоритмы,	ИД-1пк-3 Разрабатывает алгоритмы и реализует их на	Способен разрабатывать алгоритмы решения прикладных задач и реализовывать их в виде программного кода на языке Java

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
писать код на современных языках программирования, создавать веб- и desktop-приложения, работать с базами данных	языках C++, C#, Python, Java, JavaScript/TypeScript, Go с использованием эффективных структур данных и стандартных библиотек	с использованием эффективных структур данных (коллекции, стримы, опциональные типы) и стандартных библиотек.
	ИД-2пк-3 Создает клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)	Способен создавать клиент-серверные и веб-приложения с использованием Spring Boot (REST API), интегрировать фронтенд и бэкенд, применять протоколы взаимодействия (REST, gRPC, WebSocket) и современные фреймворки.
	ИД-3пк-3 Проектирует, мигрирует и оптимизирует запросы к реляционным и нереляционным БД с использованием SQL и ORM (Hibernate, Entity Framework,	Способен проектировать реляционные схемы баз данных (нормализация до 3НФ), выполнять миграции (Liquibase/Flyway), оптимизировать SQL-запросы и использовать ORM (Hibernate, Spring Data JPA) для доступа к данным.

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
	Django ORM)	
	ИД-4пк-3 Пишет модульный, документированный и тестируемый код, соблюдая стандарты оформления и принципы SOLID, использует системы сборки (Make, Gradle, npm)	Способен писать модульный, документированный (JavaDoc, комментарии) и тестируемый (JUnit, Mockito) код, соблюдая стандарты оформления (Java Code Conventions) и принципы SOLID, а также использовать системы сборки (Maven/Gradle).
ПК-4 Способен проводить тестирование, отладку и оценивать качество программного обеспечения, применяя инструменты автоматизированного тестирования и методы обеспечения надежности и безопасности	ИД-1пк-4 Применяет методы тест-дизайна (классы эквивалентности, граничные значения, pairwise) и пишет unit-, интеграционные и E2E-тесты с использованием JUnit, pytest, Jest, Cypress, Selenium	Способен применять методы тест-дизайна (классы эквивалентности, граничные значения, pairwise), писать unit-тесты (JUnit 5), интеграционные тесты (TestContainers) и E2E-тесты для проверки работоспособности приложения.
	ИД-2пк-4 Использует инструменты отладки (GDB, IDE debugger), профилирования (perf, JProfiler) и	Способен использовать отладчик IntelliJ IDEA для пошаговой отладки кода, инструменты профилирования (JProfiler, async profiler) для выявления узких мест производительности, а также JMeter/k6 для нагрузочного

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
	нагрузочного тестирования (JMeter, k6) для выявления узких мест	тестирования.
	ИД-3пк-4 Проводит нагрузочное, стресс-тестирование и оценку надёжности (MTBF, MTTR) с применением chaos-инжиниринга, статического (SAST) и динамического (DAST) анализа безопасности	Способен проводить нагрузочное и стресс-тестирование (JMeter/k6), оценивать надёжность системы (MTBF, MTTR), применять статический анализ безопасности (SonarQube, SAST) и выполнять базовые сценарии chaos-инжиниринга.
	ИД-4пк-4 (дополнительный индикатор из файла) Оценивает надёжность и безопасность ПО: анализирует обработку ошибочных ситуаций (watchdog, контроль чётности, таймауты), устойчивость к сбоям питания и разрывам	Способен анализировать обработку ошибочных ситуаций в коде (исключения, таймауты, повторные попытки), оценивать устойчивость приложения к сбоям и обеспечивать корректное восстановление после ошибок.

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
	соединения	
	ИД-5пк-4 (дополнительный индикатор из файла) Интегрирует тестирование в CI/CD (GitLab CI, GitHub Actions), настраивает генерацию отчетов (Allure), пороги покрытия кода и автоматический прогон тестов при каждом коммите	Способен настраивать CI/CD пайплайн (GitHub Actions / GitLab CI) для автоматического запуска тестов, генерации отчетов (Allure) и контроля порогов покрытия кода (JaCoCo) при каждом коммите.

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 4 з.е. 144 ак.ч.	ОФО, в ак. часах
Контактная работа:	54
Лекции/из них практическая подготовка	18
Лабораторных работ/из них практическая подготовка	36/36
Практических занятий/из них практическая подготовка	
Самостоятельная работа	36
Формы контроля	
Экзамен	54
Зачет	
Зачет с оценкой	
Расчетно-графические работы	
Курсовые работа	
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	Продвинутый тест-дизайн и TDD: Классы эквивалентности, граничные значения, попарное тестирование. TDD в сложных сценариях (интеграции, legacy). ЛАБОРАТОРНАЯ РАБОТА: Тест-дизайн: TDD с JUnit 5 и Mockito		2		4/4	4
2	Интеграционное и нагрузочное тестирование: TestContainers, Selenium. Стратегии интеграции. JMeter, профилирование (JProfiler, async profiler). ЛАБОРАТОРНАЯ РАБОТА: Интеграционное тестирование: TestContainers + PostgreSQL		2		4/4	4
3	Рефакторинг и технический долг: Каталог Фаулера. Оценка техдолга. Стратегии управления. Автоматический рефакторинг в IDE. ЛАБОРАТОРНАЯ РАБОТА: Нагрузочное тестирование: JMeter против Spring Boot API		2		4/4	4
4	Безопасность на уровне кода: JWT (access/refresh), OAuth2 (Authorization Code Flow), RBAC. Защита от CSRF, XSS, SQL injection. BCrypt. ЛАБОРАТОРНАЯ РАБОТА: Рефакторинг: устранение запахов кода в legacy-проекте		2		4/4	4
5	Продвинутое ORM и работа с БД: N+1 проблема, стратегии загрузки (join fetch, batch). Транзакции (ACID, уровни изоляции). Миграции (Liquibase/Flyway). ЛАБОРАТОРНАЯ РАБОТА: Безопасность: JWT + Spring Security +		2		4/4	4

	RBAC					
6	Асинхронное и реактивное программирование: CompletableFuture, Project Reactor (Mono/Flux), Spring WebFlux. Backpressure. Сравнение с блокирующим I/O. ЛАБОРАТОРНАЯ РАБОТА: Безопасность: OAuth2 (вход через GitHub / Google)		2		4/4	4
7	Альтернативы REST: GraphQL и gRPC: GraphQL (схема, резолверы, DataLoader для N+1). gRPC (protobuf, streaming). Сравнение производительности. ЛАБОРАТОРНАЯ РАБОТА: ORM: устранение N+1 в Spring Data JPA		2		4/4	4
8	NoSQL и гибридное хранение: MongoDB (документная), Neo4j (графовая). Когда использовать NoSQL. Полиглотное хранение ЛАБОРАТОРНАЯ РАБОТА: Реактивное программирование: WebFlux + Reactor (сравнение с MVC)		2		4/4	4
9	Качество кода и CI/CD для конструирования: Статический анализ (SonarQube, Checkstyle). Метрики (покрытие, сложность). CI/CD пайплайны с качеством ЛАБОРАТОРНАЯ РАБОТА: Альтернативные API: GraphQL vs REST (сравнение)		2		4/4	4
	Экзамен	54				
	ИТОГО за 6 семестр		18		36/36	58
	ИТОГО		18		36/36	58

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. Соммервилл, И. Инженерия программного обеспечения = Software Engineering : учебник / И. Соммервилл ; пер. с англ. – 10-е изд. – Москва : Вильямс, 2021. – 816 с. – ISBN 978-5-907144-60-2. – Текст : непосредственный.
2. Мартин, Р. С. Чистый код: создание, анализ и рефакторинг = Clean Code : A Handbook of Agile Software Craftsmanship / Р. С. Мартин ; пер. с англ. – Санкт-Петербург : Питер, 2020. – 464 с. – ISBN 978-5-4461-0960-9. – Текст : непосредственный.

3. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / под науч. ред. проф. А. Н. Громова. – 3-е изд., пер. и доп. – Москва : Издательство Юрайт, 2026. – 285 с. – (Высшее образование). – ISBN 978-5-534-17952-6. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/545309>
4. Уоллс, К. Spring в действии (Spring in Action) / К. Уоллс ; пер. с англ. – 6-е изд. – Москва : ДМК Пресс, 2022. – 520 с. – ISBN 978-5-97060-985-7. – Текст : непосредственный.
5. Базы данных. PostgreSQL : учебное пособие для вузов / под ред. С. В. Кузнецова. – Москва : Издательство Юрайт, 2026. – 312 с. – (Высшее образование). – ISBN 978-5-534-18923-5. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/547123>

8.1.2. Перечень дополнительной литературы:

1. Гамма, Э. Паттерны проектирования = Design Patterns: Elements of Reusable Object-Oriented Software / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес ; пер. с англ. – Санкт-Петербург : Питер, 2020. – 448 с. – ISBN 978-5-4461-1329-3. – Текст : непосредственный.
2. Бейзер, Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер ; пер. с англ. – Санкт-Петербург : Питер, 2018. – 320 с. – ISBN 978-5-4461-0642-4. – Текст : непосредственный.
3. Postgres Pro Enterprise. Официальная документация. – URL: <https://www.postgrespro.ru/docs/> (дата обращения: 04.06.2026). – Текст : электронный.

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

1. Учебно-методический комплекс дисциплины «Конструирование программного обеспечения» для студентов направления подготовки 09.03.04 «Программная инженерия». – Ставрополь : СКФУ, 2026. – Текст : электронный // Электронная информационно-образовательная среда СКФУ.
2. Методические указания по выполнению лабораторных работ по дисциплине «Конструирование программного обеспечения» (Java-стек). – Ставрополь : СКФУ, 2026. – 64 с. – Текст : электронный.
3. Методические указания по организации самостоятельной работы студентов по дисциплине «Конструирование программного обеспечения». – Ставрополь : СКФУ, 2026. – 32 с. – Текст : электронный.

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

1. Официальная документация Spring (русскоязычный раздел) – <https://ru.spring.io>
2. Документация Postgres Pro (отечественная СУБД) – <https://www.postgrespro.ru/docs>
3. Российская платформа совместной разработки GitFlic – <https://gitflic.ru>
4. Учебник по Java и Spring (MetaNit) – <https://metanit.com/java/>
5. Сообщество Habr – статьи по Spring и Java – <https://habr.com/ru/hub/spring/>
6. Spring Roadmap – образовательные материалы – <https://roadmap.sh/spring>
7. IntelliJ IDEA Community Edition (русский дистрибутив) – <https://www.jetbrains.com/ru-ru/idea/download>

8. Документация по CI/CD GitHub Actions – <https://github.com/features/actions>
9. Официальная документация JUnit 5 – <https://junit.org/junit5/docs/current/user-guide/>
10. Официальная документация Mockito – <https://site.mockito.org/>
11. Документация по TestContainers – <https://www.testcontainers.org/>
12. Документация по Apache JMeter – <https://jmeter.apache.org/>
13. Документация по k6 – <https://k6.io/docs/>
14. Официальная документация Docker – <https://docs.docker.com/>
15. Документация по Spring Security – <https://docs.spring.io/spring-security/reference/>
16. OWASP Top Ten (актуальный перечень уязвимостей) – <https://owasp.org/www-project-top-ten/>
17. Национальный открытый университет «Интуйт» – <https://www.intuit.ru/>
18. Электронная библиотечная система «Юрайт» – <https://urait.ru>
19. Электронная библиотечная система «Лань» – <https://e.lanbook.com>
20. Электронная библиотечная система «Znanium» – <https://znanium.com>

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. Информационно-правовая система «Консультант Плюс» – справочная правовая система, содержащая нормативно-правовые акты РФ (ФЗ-152, ФЗ-149, лицензирование ПО, авторское право). Доступна в локальной сети университета.
2. Электронная информационно-образовательная среда СКФУ – внутренний ресурс университета, содержащий учебно-методические комплексы, методические указания, фонды оценочных средств и электронные курсы по дисциплине.
3. Единое окно доступа к образовательным ресурсам (window.edu.ru) – государственный образовательный портал, предоставляющий доступ к учебным и научным материалам.
4. Научная электронная библиотека eLibrary.ru – российский информационно-аналитический портал в области науки, технологии, медицины и образования.

Программное обеспечение:

Системное и инструментальное ПО

1. Astra Linux (версия «Орел») – отечественная операционная система, базовая платформа для разработки. Распространяется на свободных условиях для образовательных учреждений.
2. IntelliJ IDEA Community Edition – среда разработки (IDE) для Java, распространяемая под свободной лицензией Apache 2.0. Доступна на территории РФ без ограничений.
3. Visual Studio Code – легковесный редактор кода, свободная лицензия MIT.
4. Java Development Kit (Liberica JDK / OpenJDK) – комплект для разработки Java-

- приложений. Liberica JDK сертифицирован для работы в составе Astra Linux. Свободная лицензия GPL.
5. Apache Maven – система сборки и управления зависимостями для Java-проектов. Лицензия Apache 2.0.
 6. Gradle – система автоматизации сборки, поддерживающая многопроектные конфигурации. Лицензия Apache 2.0.
- Базы данных
7. Postgres Pro Enterprise – отечественная реляционная СУБД на основе PostgreSQL. Бесплатная академическая лицензия для вузов. Рекомендована в качестве основной СУБД.
 8. PostgreSQL – реляционная СУБД с открытым исходным кодом, свободная лицензия PostgreSQL License.
- Системы контроля версий
9. Git – распределённая система контроля версий. Свободная лицензия GPL.
 10. GitFlic – отечественная платформа совместной разработки (аналог GitHub). Специализированный модуль GitFlic Education для вузов. Включена в реестр отечественного ПО.
- Контейнеризация и CI/CD
11. Docker Desktop – платформа для контейнеризации приложений. Бесплатная лицензия для образовательных учреждений.
 12. Docker Compose – инструмент для оркестрации многоконтейнерных приложений. Свободная лицензия Apache 2.0.
 13. GitHub Actions / GitLab CI – встроенные инструменты CI/CD для автоматизации сборки, тестирования и развертывания.
- Тестирование и качество кода
14. SonarQube Community Build – платформа для статического анализа и оценки качества кода. Лицензия LGPLv3, доступна для свободного использования в учебных целях.
 15. JUnit 5 – фреймворк для модульного тестирования Java-приложений. Свободная лицензия EPL 2.0.
 16. Mockito – библиотека для создания mock-объектов в тестах. Свободная лицензия MIT.
 17. AssertJ – библиотека для выразительных утверждений (fluent assertions) в тестах. Лицензия Apache 2.0.
 18. TestContainers – библиотека для интеграционного тестирования с использованием Docker-контейнеров. Свободная лицензия MIT.
 19. Apache JMeter – инструмент для нагрузочного и стресс-тестирования. Свободная лицензия Apache 2.0.
 20. k6 – инструмент для нагрузочного тестирования с написанием сценариев на JavaScript. Свободная лицензия AGPLv3.
 21. JaCoCo – библиотека для измерения покрытия кода тестами. Лицензия EPL.
- Веб-фреймворки и библиотеки
22. Spring Boot (Open Source) – основной фреймворк для разработки корпоративных Java-приложений. Лицензия Apache 2.0. В учебных целях допустим, для промышленной разработки после 2026 года рекомендован отечественный форк Axiom Spring.
 23. Axiom Spring – отечественный форк Spring Boot с полной поддержкой и выпуском обновлений безопасности. Рекомендован для промышленной разработки после прекращения Community-поддержки оригинального Spring Boot.
 24. Spring Security – фреймворк для аутентификации, авторизации и защиты от атак.

- Лицензия Apache 2.0.
25. JWT – библиотека для создания и валидации JWT (JSON Web Token). Свободная лицензия Apache 2.0.
 26. Project Reactor – реализация реактивных потоков (Reactive Streams) для JVM с типами Mono и Flux. Лицензия Apache 2.0.
 27. Spring WebFlux – реактивный веб-фреймворк на основе Netty. Лицензия Apache 2.0.
 28. Spring for GraphQL – официальная интеграция GraphQL со Spring. Лицензия Apache 2.0.
 29. WebClient – неблокирующий HTTP-клиент для реактивных приложений. Лицензия Apache 2.0.
 30. Hibernate – реализация JPA (Java Persistence API) для объектно-реляционного отображения. Лицензия LGPL.
- Офисный пакет для оформления отчётов
31. Р7-Офис – отечественный офисный пакет, совместимый с форматами Microsoft Office. Бесплатен для образовательных учреждений.
 32. МойОфис – российский офисный пакет для работы с документами, таблицами и презентациями. Имеет бесплатную версию для вузов.
 33. LibreOffice – свободный офисный пакет с открытым исходным кодом, доступен на территории РФ без ограничений.

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.