

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ

Проректор по учебной работе,
качеству образования – первый
проректор

Хагуров Т.А.

подпись

«30» мая 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.О.32 «Технологии создания и поддержки программного
обеспечения»

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) Искусственный интеллект и машинное обучение

Форма обучения очная

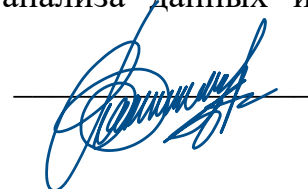
Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Технологии создания и поддержки программного обеспечения» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.03 Прикладная информатика.

Программу составил(и):

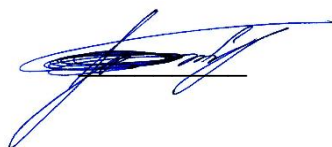
Р.Ю. Вишняков, канд., техн., наук, доцент кафедры анализа данных и искусственного интеллекта



Рабочая программа дисциплины «Технологии создания и поддержки программного обеспечения» утверждена на заседании кафедры анализа данных и искусственного интеллекта протокол № 13 «20» мая 2025 г.

Заведующий кафедрой (разработчика)

Коваленко А.В.



Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 4 от « 23 » мая 2025 г.

Председатель УМК факультета

Коваленко А.В.



Рецензент (-ы):

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»

Уртенев Махамед Али Хусеевич, профессор кафедры прикладной математики Кубанского государственного университета, доктор физико-математических наук, профессор

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Цели изучения дисциплины определены государственным образовательным стандартом высшего образования и соотнесены с общими целями ООП ВО по направлению подготовки 09.03.03 Прикладная информатика, в рамках которой преподается дисциплина.

Цели дисциплины «Технологии создания и поддержки программного обеспечения»: формирование у студентов комплексных знаний и практических навыков в области современных технологий создания и поддержки программного обеспечения на протяжении всего жизненного цикла разработки (ЖЦР).

1.2 Задачи дисциплины

- Изучить основные модели ЖЦР ПО и их особенности.
- Освоить современные методологии разработки ПО, включая Agile-подходы.
- Научиться использовать системы контроля версий для совместной разработки.
- Приобрести навыки тестирования программного обеспечения на различных уровнях.
- Познакомиться с принципами автоматизации сборки, развертывания и мониторинга ПО (CI/CD).
- Освоить базовые концепции DevOps и их применение в практике разработки.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Технологии создания и поддержки программного обеспечения» относится к профильным дисциплинам, обеспечивающим формирование профессиональных компетенций выпускника по направлению подготовки 09.03.03 – Прикладная информатика.

Для успешного освоения дисциплины необходимы следующие пререквизиты: информатика, основы программирования, операционные системы и сети, технологии программирования.

Дисциплина «Технологии создания и поддержки программного обеспечения» является основой для изучения следующих дисциплин.

Постреквизиты: Б1.О.29 «Анализ, проектирование и разработка БД», Б1.В.05 «Облачные технологии и бэкэндразработка», Б1.В.07 «Разработка мобильных приложений», а также других дисциплин связанных с: разработкой, тестированием, поддержкой программного обеспечения, разработкой архитектур информационных систем. Углубляет знания, полученные в рамках дисциплин Б1.О.10 «Основы программирования», Б1.О.25 «Объектно-ориентированное программирование», Б1.О.27 «Разработка пользовательского WEB интерфейса», Б1.В.01 «Основы программирования на Python».

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

КОД компетенции	Результат
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.

ПК-1	Способен решать актуальные и значимые задачи прикладной информатики.
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства и использовать их при решении задач профессиональной деятельности.
ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;
ОПК-8	Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.
ПК-3	Способен разрабатывать и адаптировать прикладное программное обеспечение.
ПК-5	Способен выявить естественнонаучную сущность проблем, возникающих в ходе профессиональной деятельности в области моделирования и анализа сложных естественных и искусственных систем.
ПК-6	Способен моделировать прикладные (бизнес) процессы и предметную область.
ПК-8	Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции.

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 6 зачетных единиц (216 часов), их распределение по видам работ представлено в таблице

Виды работ	Всего Часов	Форма обучения	
		Очная	
		4 семестр (часы)	5 семестр (часы)
Контактная работа, в том числе:	124,4	72,2	52,2
Аудиторные занятия (всего):	118	68	50
занятия лекционного типа	50	34	16
лабораторные занятия	68	34	34
практические занятия	-	-	-
семинарские занятия	-	-	-
Иная контактная работа:	6,4	4,2	2,2
Контроль самостоятельной работы (КСР)	6	4	2

Промежуточная аттестация (ИКР)		0,4	0,2	0,2
Самостоятельная работа, в том числе:		91,6	35,8	55,8
Проработка и повторение лекционного материала и материала учебников и учебных пособий		91,6	35,8	55,8
Подготовка к текущему контролю				
Контроль:		-	-	-
Подготовка к экзамену		-	-	-
Общая трудоемкость	час.	216	108	108
	в том числе контактная работа	124,4	72,2	52,2
	зач. Ед	6	3	3

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые в 3 семестре (очная форма обучения)

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная Работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1.	Жизненный цикл разработки программного обеспечения (ЖЦР)	24,2	4		8	12,2
2.	Методологии разработки ПО	28,2	8		8	12,2
3.	Системы контроля версий (Git)	26,2	6		8	12,2
4.	Тестирование программного обеспечения	26,2	6		8	12,2
5.	Автоматизация сборки и развертывания ПО (CI/CD)	26,2	6		8	12,2
6.	Основы DevOps	24,2	4		8	12,2
7.	Контейнеризация (Docker)	30,2	8		10	12,2
8.	Оркестрация контейнеров (Kubernetes)	30,2	8		10	12,2
	ИТОГО по разделам дисциплины	215,6	50		68	97,6
	Промежуточная аттестация (ИКР)	0,4				
	Подготовка к текущему контролю					
	Общая трудоемкость по дисциплине	16				

Примечание: Л – лекции, ПЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1.	Жизненный цикл разработки программного обеспечения (ЖЦР)	- Классические модели их преимущества и недостатки, области применения: Waterfall, V-model, Spiral model. - Итеративные и инкрементные модели: Rational Unified Process (RUP). - Agile методологии их принципы, роли, артефакты, церемонии: Scrum, Kanban, Extreme Programming (XP). - Сравнение Agile с традиционными подходами.	Устные ответы; Тест;
2.	Методологии разработки ПО	- Принципы SOLID и DRY. - Паттерны проектирования: основные типы паттернов (создающие, структурные, поведенческие). - Code Review – принципы проведения и инструменты.	Устные ответы; Тест;
3.	Системы контроля версий (Git)	- Основные понятия: репозиторий, коммит, ветка, и т.д. - Команды Git: init, clone, add, commit, push, pull, branch, merge, rebase. - Работа с удаленными репозиториями (GitHub, GitLab), создание репозитория, клонирование, отправка изменений, работа с issue tracker. - Стратегии ветвления: Gitflow, GitHub Flow.	Устные ответы; Тесты; Лабораторная работа.
4.	Тестирование программного обеспечения	- Уровни тестирования: модульное тестирование (unit testing), интеграционное тестирование, системное тестирование, приемочное тестирование. - Виды тестирования: функциональное тестирование, нефункциональное тестирование (нагрузочное, стрессовое, security).	Устные ответы; Тесты; Лабораторная работа.

		– Методики тестирования: black-box testing, white-box testing. – Инструменты автоматизированного тестирования.	
5.	Автоматизация сборки и развертывания ПО (CI/CD)	– Принципы CI/CD. – Инструменты CI/CD: Jenkins, Travis CI, CircleCI, GitLab CI. – Настройка пайплайнов CI/CD для автоматической сборки, тестирования и развертывания приложений.	Устные ответы; Тесты; Лабораторная работа.
6.	Основы DevOps	– Культура DevOps: Collaboration, Automation, Measurement, Sharing (CAMS). – Инфраструктура как код (IaC). – Мониторинг и логирование.	Устные ответы; Тесты; Лабораторная работа.
7.	Контейнеризация (Docker)	– Основные понятия: образы, контейнеры, Dockerfile. – Создание и запуск контейнеров. – Docker Compose для управления многоконтейнерными приложениями. – Разработка Dockerfile для простого приложения.	Устные ответы; Тесты; Лабораторная работа.
8.	Оркестрация контейнеров (Kubernetes)	– Основные понятия: поды, сервисы, деплойменты, namespace. – Развертывание приложений в Kubernetes. – Масштабирование и обновление приложений в Kubernetes. – Практическое развертывание простого приложения в Kubernetes.	Устные ответы; Тесты; Лабораторная работа.

2.3.2 Занятия семинарского типа (практические / семинарские занятия/ лабораторные работы)

№	Наименование раздела (темы)	Тематика занятий/работ	Форма текущего контроля
1.	Системы контроля версий (Git)	Практическое применение Git для совместной разработки небольшого проекта, имитирующего реальный рабочий процесс команды разработчиков. Акцент на понимании принципов ветвления и разрешения конфликтов. Освоить базовые команды Git для управления репозиторием, включая инициализацию, добавление файлов, коммиты, ветвление и слияние. Научиться работать с удаленными репозиториями на GitHub/GitLab, создавать pull requests и проводить code review.	ЛР
2.	Тестирование программного обеспечения	Разработка тестового набора для обеспечения качества кода, выявление ошибок на ранних этапах разработки и повышение надежности программного продукта. Написать модульные тесты (unit tests) для существующего кода с использованием выбранного фреймворка (JUnit, pytest). Реализовать интеграционные тесты для проверки взаимодействия между различными компонентами приложения. Изучить и применить различные техники тестирования (black-box, white-box).	ЛР
3.	Автоматизация сборки и развертывания ПО (CI/CD)	Автоматизация рутинных задач разработки, ускорение процесса поставки программного обеспечения и снижение риска ошибок при развертывании. Настроить CI/CD-пайплайн с использованием Jenkins/Travis CI/GitLab CI для автоматической сборки, тестирования и развертывания небольшого приложения при каждом изменении кода в репозитории.	ЛР
4.	DevOps	Применение принципов DevOps для автоматизации управления инфраструктурой, повышения эффективности работы команды разработки и обеспечения стабильности и масштабируемости приложений.	ЛР

		Использовать инструменты инфраструктуры как код для автоматического создания и настройки виртуальных машин или облачных ресурсов. Настроить базовый мониторинг приложения.	
5.	Контейнеризация (Docker)	Упаковка приложений в изолированные контейнеры для обеспечения переносимости, воспроизводимости и упрощения развертывания. Создать Dockerfile для контейнеризации простого приложения (например, веб-сервера). Научиться собирать образы Docker, запускать контейнеры и управлять ими. Использовать Docker Compose для управления многоконтейнерными приложениями.	ЛР
6.	Оркестрация контейнеров (Kubernetes)	Автоматизация развертывания, управления и масштабирования контейнеризированных приложений в кластерной среде, обеспечение высокой доступности и отказоустойчивости. Развернуть контейнеризированное приложение в Kubernetes-кластере с использованием YAML-манифестов. Настроить масштабирование приложения и обновление версий. Изучить основные компоненты Kubernetes: поды, сервисы, деплойменты.	ЛР

Защита лабораторной работы (ЛР), выполнение курсового проекта (КП), курсовой работы (КР), расчетно-графического задания (РГЗ), написание реферата (Р), эссе (Э), коллоквиум (К), тестирование (Т) и т.д.

2.3.3 Курсовые работы

Курсовые работы не предусмотрены.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Проработка и повторение лекционного материала, материала учебной и научной литературы, подготовка к семинарским занятиям	Методические указания для подготовки к лекционным и семинарским занятиям, утвержденные на заседании кафедры анализа данных и искусственного интеллекта факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 22.03.2023 г. Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры анализа данных и искусственного интеллекта факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 22.03.2023 г.
2	Подготовка к лабораторным занятиям	Методические указания по выполнению лабораторных работ, утвержденные на заседании кафедры анализа данных и искусственного интеллекта факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 22.03.2023 г.
3	Подготовка к решению задач и тестов	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры анализа данных и искусственного интеллекта факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 22.03.2023 г.
4	Подготовка к текущему контролю	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры анализа данных и искусственного интеллекта факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 22.03.2023 г.
5	Подготовка докладов	Методические указания для подготовки эссе, рефератов, курсовых работ, утвержденные на заседании кафедры анализа данных и искусственного интеллекта факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 22.03.2023 г.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.
- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.
- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.
- Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:
 - Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.
 - Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.
 - Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.
 - Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.
 - Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеперечисленных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

7. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Технологии создания и поддержки программного обеспечения».

Оценочные средства включают контрольные материалы для проведения **текущего контроля** в форме тестов и лабораторных работ, и **промежуточной аттестации** в форме вопросов к зачету.

Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы и предполагает овладение материалами лекций, литературы, программы, работу студентов в ходе проведения лабораторных занятий, а также систематическое выполнение лабораторных работ, решение практических задач и иных заданий для самостоятельной работы студентов. Данный вид контроля стимулирует у студентов стремление к систематической самостоятельной работе по изучению дисциплины. Он предназначен для оценки самостоятельной работы слушателей по решению задач, выполнению лабораторных работ, подведения итогов тестирования. Оценивается также активность и качество результатов практической работы на занятиях, участие в дискуссиях, обсуждениях и т.п. Индивидуальные и групповые самостоятельные, аудиторские работы по всем темам дисциплины организованы единообразным образом. Для контроля освоения содержания дисциплины используются оценочные средства. Они направлены на определение степени сформированности компетенций.

Промежуточная аттестация студентов осуществляется в рамках завершения изучения дисциплины и позволяет определить качество усвоения изученного материала, предполагает контроль и управление процессом приобретения студентами необходимых знаний, умения и навыков, определяемых по ФГОС ВО по соответствующему направлению подготовки в качестве результатов освоения учебной дисциплины.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачете;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№	Результаты обучения	Наименование оценочного средства
---	---------------------	----------------------------------

	Код и наименование индикатора		Текущий контроль	Промежуточная аттестация
1	ИПК-3.2. Разрабатывает самостоятельно технологии и инструментальные средства работы с большими данными	Знает: средства для работы с большими данными Умеет: разрабатывать инструментальные средства работы с большими данными в задачах работы с речью и звуковыми данными Владеет: методами разработки технологий работы с большими данными	Лабораторные работы №1, 2 по разделу «Обработка естественного языка с помощью классических методов машинного обучения», Реферат по темам 1-6	Вопрос на зачете 1-3, 7-10
2	ИПК-4.3. Применяет самостоятельно полученные профессиональные знания для решения больших языковых моделей глубокого обучения	Знает: известные большие языковые модели Умеет: использовать предобученные большие языковые модели Владеет: навыками для решения больших языковых моделей глубокого обучения	Лабораторная работа 5 по разделу «Большие языковые модели», Реферат по темам 9-12, 13-15	Вопрос на зачете 13-15, 18-21, 23, 24
3	ИОПК-5.1. Знает и применяет современные технологии выполнения работ по модернизации программного и аппаратного обеспечения	Знает: методы по модернизации программного обеспечения Умеет: применять современные технологии выполнения работ по модернизации программного и аппаратного обеспечения Владеет: знаниями по совершенствованию программного обеспечения для задач обработки естественного языка и работы со звуком	Лабораторные работы №3, 4 по разделу «Архитектуры нейронных сетей для работы с речью и звуком», Реферат по темам 7, 8	Вопрос на зачете 4-6, 11, 12, 16, 17, 22
4	ИПК-5.2 Применяет знание основных методов искусственного интеллекта в решении задач профессиональной деятельности в качестве научных сотрудников, преподавателей образовательных организаций высшего образования, инженеров, технологов	Знает основных методы искусственного интеллекта в решении задач профессиональной деятельности Умеет Осуществлять теоретическое обобщение научных данных, результатов экспериментов и наблюдений Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач	Лабораторная работа 5 по разделу «Большие языковые модели», Реферат по темам 7, 8	Вопрос на зачете 4-6, 11, 12, 16, 17, 22-24

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Пример опроса по темам с ответами

1. Системы контроля версий (Git)

Вопрос 1: что такое система контроля версий и зачем она нужна в разработке программного обеспечения?

Ответ: система контроля версий (СКВ) – это инструмент, позволяющий отслеживать изменения в файлах проекта. Она необходима для совместной работы над

проектом, сохранения истории изменений, возможности отката к предыдущим версиям и ветвления разработки. Без СКВ сложно поддерживать стабильность кода и эффективно работать в команде.

Вопрос 2: объясните разницу между командами `git merge` и `git rebase`. В каких ситуациях целесообразно использовать каждую из них?

Ответ: `git merge` создает новый коммит, объединяющий изменения из двух веток. История изменений сохраняется в виде ветвей и слияний. Команда `git rebase` перемещает последовательность коммитов одной ветки на вершину другой, создавая линейную историю. Rebase полезен для очистки истории перед отправкой изменений, но может привести к проблемам при совместной работе, если ветка уже опубликована.

Вопрос 3: что такое `.gitignore` и как он используется? Приведите примеры файлов или папок, которые обычно добавляют в этот файл.

Ответ: `.gitignore` – это текстовый файл, содержащий список файлов и папок, которые Git должен игнорировать при отслеживании изменений. Он позволяет исключить из репозитория временные файлы, скомпилированный код, логи и другие артефакты, не относящиеся к исходному коду проекта. Примеры: `/bin/`, `/obj/`, `.idea/`, `*.log`.

Вопрос 4: что такое pull request (PR) и зачем он нужен? Опишите процесс создания и рецензирования PR на GitHub/GitLab.

Ответ: Pull Request – это запрос на слияние изменений из одной ветки в другую. Он позволяет другим разработчикам просмотреть код, оставить комментарии и предложить улучшения перед интеграцией изменений в основную ветку. Процесс включает создание PR, рецензирование кода, внесение правок (при необходимости) и утверждение PR для слияния.

Вопрос 5: как разрешить конфликт при слиянии веток в Git? Опишите шаги, которые необходимо предпринять.

Ответ: конфликты возникают, когда изменения в разных ветках затрагивают одни и те же строки кода. Для разрешения конфликта нужно открыть файл с конфликтом, найти маркеры конфликта (`<<<<<<<<`, `=====`, `>>>>>>>>`), вручную выбрать нужные изменения или объединить их, удалить маркеры и закоммитить результат.

2. Тестирование программного обеспечения

Вопрос 1: какие основные уровни тестирования вы знаете? Опишите каждый из них.

Ответ: модульное тестирование (unit testing) – проверка отдельных модулей или функций кода. Интеграционное тестирование – проверка взаимодействия между различными компонентами системы. Системное тестирование – проверка всей системы в целом на соответствие требованиям. Приемочное тестирование – проверка системы конечным пользователем для подтверждения ее готовности к эксплуатации.

Вопрос 2: что такое TDD (Test-Driven Development)? Опишите процесс разработки по методологии TDD.

Ответ: TDD – это методология разработки, в которой сначала пишутся тесты, а затем код, который эти тесты проходит. Процесс: 1) написать тест, который не проходит (Red). 2) написать минимальный объем кода для прохождения теста (Green). 3) рефакторинг кода для улучшения его структуры и читаемости (Refactor).

Вопрос 3: в чем разница между black-box и white-box тестированием? Приведите примеры.

Ответ: Black-box – тестирование без знания внутренней структуры кода, основанное на входных данных и ожидаемых результатах (например, функциональное тестирование). White-box – тестирование с доступом к исходному коду, позволяющее проверить логику работы программы (например, модульное тестирование).

Вопрос 4: что такое покрытие кода (code coverage) и зачем оно нужно?

Ответ: покрытие кода – это метрика, показывающая процент строк или ветвей кода, которые были выполнены тестами. Оно позволяет оценить эффективность тестового набора и выявить участки кода, которые не протестированы. Высокое покрытие кода не гарантирует отсутствие ошибок, но повышает уверенность в качестве кода.

Вопрос 5: какие инструменты для автоматизации тестирования вы знаете? Приведите примеры.

Ответ: pytest (Python), Selenium (автоматизация веб-тестирования), Jest (JavaScript). Эти инструменты позволяют создавать и запускать тесты автоматически, что ускоряет процесс разработки и повышает качество кода.

3. Автоматизация сборки и развертывания ПО (CI/CD)

Вопрос 1: что такое CI/CD? Объясните значение каждого термина.

Ответ: CI (Continuous Integration) – это практика автоматической интеграции изменений кода от разных разработчиков в общий репозиторий. CD (Continuous Delivery/Deployment) – это практика автоматизации процесса доставки или развертывания программного обеспечения в производственную среду.

Вопрос 2: какие преимущества дает использование CI/CD?

Ответ: ускорение цикла разработки, снижение риска ошибок при развертывании, повышение качества кода, более частые релизы, автоматизация рутинных задач.

Вопрос 3: опишите основные этапы типичного CI/CD-пайплайна.

Ответ: 1) Получение изменений из репозитория. 2) Сборка проекта. 3) Запуск тестов (модульные, интеграционные). 4) Анализ кода (статический анализ). 5) Создание артефакта (например, Docker-образ). 6) Развертывание в тестовую среду. 7) Автоматическое тестирование в тестовой среде. 8) Развертывание в производственную среду (при CD).

Вопрос 4: что такое Jenkinsfile? Для чего он используется?

Ответ: Jenkinsfile – это текстовый файл, содержащий описание CI/CD-пайплайна. Он позволяет определить последовательность шагов сборки, тестирования и развертывания в декларативном формате.

Вопрос 5: какие инструменты можно использовать для мониторинга CI/CD-пайплайна?

Ответ: Jenkins UI, GitLab CI/CD dashboard, CircleCI dashboard, Grafana (для визуализации метрик).

Пример технического задания к лабораторной работе

Техническое задание на выполнение лабораторной работы № [Номер]

По дисциплине: «Технологии создания и поддержки программного обеспечения»

На тему «Автоматизация сборки и развертывания ПО (CI/CD)»

1. Цель работы

Ознакомление с принципами и инструментами CI/CD, приобретение практических навыков автоматизации процессов сборки, тестирования и развертывания программного обеспечения.

2. Задачи работы

- Изучить основные концепции Continuous Integration (CI) и Continuous Delivery/Deployment (CD).
- Освоить работу с системой управления версиями Git.
- Настроить автоматическую сборку проекта при внесении изменений в репозиторий.

- Автоматизировать запуск тестов после сборки проекта.
- Развернуть собранное приложение на тестовый сервер (или локальную виртуальную машину).

3. Исходные данные

- Репозиторий с исходным кодом простого приложения [выдается ссылка на репозиторий].
- Доступ к платформе CI/CD (например, GitLab или GitHub) [указывается конкретная платформа и необходимые учетные данные, если требуются].
- Тестовый сервер (или локальная виртуальная машина) с установленным необходимым окружением для запуска приложения [указываются характеристики сервера/ВМ: ОС, версия языка программирования, веб-сервер и т.д.].

4. Требования к выполняемой работе

- Создать pipeline CI/CD в выбранной платформе.
- Pipeline должен выполнять следующие этапы:
 - Сборка. Клонирование репозитория, установка зависимостей.
 - Тестирование. Запуск unit-тестов и/или интеграционных тестов (если они есть в проекте).
 - Развертывание. Копирование собранного приложения на тестовый сервер или виртуальную машину.
- Настроить автоматический запуск pipeline при каждом коммите в основную ветку репозитория (например, *main* или *master*).
- Предоставить скриншоты настроек pipeline и логи успешного выполнения всех этапов.

5. Ожидаемые результаты

- Рабочий pipeline CI/CD, автоматически собирающий, тестирующий и развертывающий приложение при внесении изменений в репозиторий.
- Отчет о выполненной работе, содержащий:
 - Краткое описание процесса настройки pipeline.
 - Скриншоты настроек pipeline.
 - Логи успешного выполнения всех этапов pipeline.
 - Выводы о полученном опыте и сложностях, возникших в процессе работы.

6. Дополнительные материалы

- При необходимости.

Документация по выбранной платформе CI/CD: [Ссылка на документацию]

Примеры конфигурационных файлов для pipeline: [Ссылка на примеры]

Преподаватель: [ФИО преподавателя]

Подпись студента: _____ (подтверждает получение задания)

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен/зачет)

Методические рекомендации, определяющие процедуры оценивания выполнения лабораторных работ:

Оценивание выполнения лабораторных работ осуществляется с целью контроля освоения студентами практических навыков и теоретических знаний по дисциплине «Технологии создания и поддержки программного обеспечения». Задание считается выполненным при выполнении следующих условий:

1. Реализованный функционал соответствует требованиям, указанным в задании к лабораторной работе. Проверяется корректность работы программы / скрипта / конфигурации в соответствии с поставленной целью.

2. Код должен быть читаемым, хорошо структурированным и соответствовать общепринятым стандартам кодирования для используемого языка программирования. Конфигурационные файлы должны быть корректно оформлены и содержать необходимые параметры.

3. Наличие краткого отчета о выполненной работе, включающего:

- описание цели лабораторной работы;
- краткое описание реализованного решения;
- инструкцию по запуску и использованию разработанного продукта;
- скриншоты, демонстрирующие работу программы/скрипта/конфигурации.

4. Лабораторная работа должна быть выполнена и представлена к проверке в установленные сроки. Задержка с представлением работы влечет снижение оценки.

5. Работа должна быть выполнена студентом самостоятельно. Плагиат или использование готовых решений без понимания принципов их работы недопустимы и приводят к нулевой оценке. При использовании сторонних библиотек/инструментов необходимо указать источники.

6. Наличие тестов, покрывающих основные сценарии использования разработанного функционала (для работ по тестированию). Проверка работоспособности тестов и их соответствие требованиям задания.

Критерии оценивания лабораторных работ:

- Отлично (5 баллов): все условия выполнены в полном объеме. Работа выполнена качественно, с соблюдением всех требований и стандартов. Демонстрируется глубокое понимание материала.
- Хорошо (4 балла): большинство условий выполнены. Незначительные недостатки не влияют на общую работоспособность решения. Проявляется хорошее понимание материала.
- Удовлетворительно (3 балла): выполнены основные условия, но имеются существенные недостатки в функциональности или качестве кода/конфигурации. Понимание материала поверхностное.
- Неудовлетворительно (2 балла и ниже): не выполнено большинство условий. Работа не соответствует требованиям задания. Продemonстрировано недостаточное понимание материала.

Процедура оценивания лабораторных работ:

1. Преподаватель проверяет представленную студентом работу на соответствие вышеперечисленным условиям.

2. В случае необходимости, преподаватель может запросить у студента демонстрацию работы программы/скрипта/конфигурации и ответы на вопросы по реализованному решению.

3. Оценка выставляется на основании комплексной оценки выполненной работы с учетом всех критериев.

Методические рекомендации, определяющие процедуры оценивания на зачете:

Зачет по дисциплине «Технологии создания и поддержки программного обеспечения» проводится с целью комплексной оценки освоения студентами

теоретических знаний и практических навыков, полученных в процессе изучения курса. Оценка на зачете формируется из нескольких компонентов: выполнения лабораторных работ, решения теоретических вопросов и демонстрации проектной работы.

Компоненты оценки зачета:

1. Оценка по лабораторным работам (35% от общей оценки): учитывается средняя оценка, полученная студентом за все выполненные лабораторные работы в течение семестра. Оцениваются соответствие выполненных работ требованиям заданий, качество кода/конфигураций, документирование и соблюдение сроков выполнения.

2. Теоретический опрос (35% от общей оценки): проводится в форме устного опроса или письменного теста (по решению преподавателя). Охватывает ключевые понятия, принципы и технологии, изученные в рамках курса. Вопросы направлены на проверку понимания теоретических основ разработки программного обеспечения, процессов CI/CD, DevOps, контейнеризации и оркестрации.

3. Демонстрация проектной работы (30% от общей оценки - *опционально*): в рамках курса предусмотрена разработка мини-проекта, студент должен продемонстрировать его работоспособность и объяснить принятые архитектурные решения. Оценивается соответствие проекта требованиям задания, качество кода, документирование и способность студента ответить на вопросы по проекту.

Критерии оценивания – зачет:

«Зачтено» с оценкой «отлично» (5 баллов): студент демонстрирует глубокое понимание теоретического материала, уверенно отвечает на вопросы, успешно выполнил все лабораторные работы с высоким качеством и (при наличии) представил работоспособный проект, соответствующий требованиям.

«Зачтено» с оценкой «хорошо» (4 балла): студент хорошо знает теоретический материал, может ответить на большинство вопросов, успешно выполнил все лабораторные работы с незначительными замечаниями и (при наличии) представил работоспособный проект, в котором есть небольшие недостатки.

«Зачтено» с оценкой «удовлетворительно» (3 балла): студент владеет основными понятиями и принципами дисциплины, может ответить на базовые вопросы, выполнил все лабораторные работы с существенными замечаниями или пропустил несколько работ и (при наличии) представил проект, требующий значительной доработки.

«Не зачтено» с оценкой «неудовлетворительно» (2 балла и ниже): студент демонстрирует недостаточное понимание теоретического материала, не может ответить на большинство вопросов, имеет большое количество непроверенных или неудовлетворительно выполненных или не выполненных лабораторных работ, не представил проект или представил неработоспособный проект.

Процедура оценивания – зачет

1. Студент допускается к зачету при условии успешного выполнения всех лабораторных работ (получение оценки не ниже удовлетворительной по каждой работе).

2. Теоретический опрос проводится в индивидуальном порядке или в форме группового опроса (по решению преподавателя).

3. При демонстрации проектной работы студент должен предоставить доступ к коду проекта и продемонстрировать его работоспособность.

4. Окончательная оценка на зачете формируется путем суммирования оценок по всем компонентам с учетом их весовых коэффициентов.

5. Решение о допуске студента к зачету принимается преподавателем на основании комплексной оценки всех компонентов.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачете;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1. Основные учебники и руководства

1. Соммервилл И. *Инженерия программного обеспечения*. — 10-е изд. — М.: Вильямс, 2020.

2. Брукс Ф. *Мифический человеко-месяц, или Как создаются программные системы*. — СПб.: Символ-Плюс, 2021.

3. Макконнелл С. *Совершенный код*. — 2-е изд. — М.: Русская редакция, 2022.

5.2 Методические пособия и практикумы

4. Орлов С.А., Цилькер Б.Я. *Технологии разработки программного обеспечения*. — СПб.: Питер, 2021.

5. Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. *Проектирование и разработка программного обеспечения*. — М.: Форум, 2020.

5.3 Дополнительные материалы и стандарты

6. ISO/IEC/IEEE 12207:2017 Системная и программная инженерия. Процессы жизненного цикла программных средств.

7. Рудаков М.В. *DevOps: принципы, методики, инструменты*. — М.: ДМК Пресс, 2023.

5.4 Электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ
<http://megapro.kubsu.ru/MegaPro/Web>

2. Электронная библиотека трудов ученых КубГУ
<http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>

3. Среда модульного динамического обучения <http://moodle.kubsu.ru>

4. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
5. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>

6. Методические указания и рекомендации для обучающихся по освоению дисциплины (модуля)

Настоящие методические указания предназначены для студентов, изучающих дисциплину «Технологии создания и поддержки программного обеспечения». Они содержат рекомендации по организации учебной деятельности, освоению теоретического материала, выполнению практических работ и подготовке к зачету.

6.1. Цели и задачи дисциплины

Целью изучения дисциплины является формирование у студентов системы знаний о современных технологиях разработки, тестирования, сборки, развертывания и поддержки программного обеспечения.

Задачи дисциплины:

- Изучение основных принципов и практик DevOps.
- Освоение инструментов автоматизации сборки и развертывания (CI/CD).
- Приобретение навыков работы с системами контроля версий (Git).
- Изучение технологий контейнеризации (Docker) и оркестрации (Kubernetes).
- Формирование понимания важности тестирования программного обеспечения и освоение инструментов автоматизированного тестирования.
- Ознакомление с методами мониторинга и логирования приложений.

6.2. Организация учебной деятельности

Дисциплина включает в себя лекционные занятия, практические работы (лабораторные), самостоятельную работу студентов и выполнение мини-проектной работы.

Лекции. На лекциях рассматриваются теоретические основы дисциплины, ключевые понятия и технологии. Студентам рекомендуется вести конспекты, активно участвовать в обсуждениях и задавать вопросы преподавателю.

Лекция представляет собой систематическое изложение учебного материала, обеспечивающее целостное представление предмета на основе новейших научных данных. Ее цель – организация познавательной деятельности студентов по освоению программного материала дисциплины.

Основные задачи лекции: формирование системы знаний, развитие умения аргументированно излагать научный материал, расширение профессионального кругозора и освещение актуальных достижений науки.

Для успешной подготовки к лекциям необходимо изучить основную литературу по теме и обратить внимание на вопросы для рассмотрения, предложенные в конце каждой темы. Усвоение материала будет наиболее эффективным при следующих условиях:

1. Систематическая работа на занятиях и самостоятельное закрепление знаний.
2. Добросовестное выполнение заданий на практических занятиях.
3. Анализ предпосылок, умозаключений и выводов учебного курса, а также взаимосвязей между его разделами.
4. Сопоставление различных точек зрения по рассматриваемым проблемам и критическая оценка научной литературы.
5. Разработка предложений по совершенствованию учебного процесса."

Лабораторные работы. Лабораторные работы направлены на закрепление теоретических знаний, приобретение практических навыков работы с инструментами разработки и эксплуатации программного обеспечения, а также формирование компетенций, полученных в ходе лекционных занятий и самостоятельной работы. К каждому лабораторному занятию преподаватель разрабатывает практические задания, определяет требования к их выполнению и предоставляет методические рекомендации, содержащиеся в фонде оценочных средств учебной дисциплины. В процессе подготовки к

занятиям студент осуществляет сбор и анализ информации по тематике работы, используя открытые источники (научные публикации, аналитические материалы, ресурсы сети Интернет и др.), а также практический опыт и доступные данные об объекте исследования. Контроль за выполнением самостоятельной работы осуществляется в ходе изучения каждой темы дисциплины на лабораторных занятиях.

Самостоятельная работа. Самостоятельная работа студентов по дисциплине направлена на закрепление и систематизацию теоретических знаний, формирование практических навыков их применения при решении задач, а также углубленное изучение отдельных тем курса. Она включает в себя изучение основной и дополнительной литературы, подготовку к лекциям и лабораторным работам, выполнение домашних заданий и самостоятельный поиск информации с использованием ресурсов библиотеки, онлайн-курсов и документации по используемым инструментам.

Подготовка к каждому разделу дисциплины осуществляется поэтапно:

Этап 1. Изучение теоретического материала на основе лекций преподавателя, рекомендуемой основной литературы и научных публикаций для овладения ключевыми понятиями и формирования понимания аналитического инструментария в данной области.

Этап 2. Выполнение лабораторных работ, направленных на формирование практических умений и навыков в рамках заявленных компетенций. Этот этап предполагает самостоятельный поиск эмпирических данных, их обобщение и анализ по схеме, рекомендованной преподавателем, а также формулирование выводов.

Контролируемая самостоятельная работа (КСР) представляет собой совокупность заданий, выполняемых студентом под руководством и контролем преподавателя с целью закрепления теоретических знаний, формирования умений и навыков, а также накопления практического опыта. Студенты руководствуются методическими указаниями преподавателя и инструкциями по выполнению типовых заданий.

Текущий контроль КСР осуществляется еженедельно в соответствии с программой дисциплины. Задания для самостоятельной работы и требования к их выполнению предоставляются преподавателем на основе фонда оценочных средств по дисциплине.

Для студентов-инвалидов и лиц с ограниченными возможностями здоровья важным элементом освоения дисциплины являются индивидуальные консультации, обеспечивающие дополнительное разъяснение учебного материала и способствующие индивидуализации обучения. Консультации также поддерживают воспитательный контакт между преподавателем и обучающимся.

6.3. Рекомендации по изучению теоретического материала

- Начните с изучения основных понятий и терминов.
- Используйте различные источники информации: учебники, статьи, онлайн-документацию.
- Старайтесь понять взаимосвязь между различными концепциями и технологиями.
- Приводите примеры из реальной практики для лучшего усвоения материала.
- Регулярно повторяйте изученный материал.

6.4. Рекомендации по выполнению лабораторных работ

- Внимательно читайте условия задания перед началом выполнения работы.
- Разбейте задачу на более мелкие подзадачи и решайте их последовательно.
- Используйте систему контроля версий (Git) для отслеживания изменений в коде.
- Пишите чистый, хорошо документированный код.
- Тестируйте свою работу после каждого этапа разработки.
- Соблюдайте сроки выполнения лабораторных работ.

6.5. Рекомендации для самостоятельного изучения

- Используйте учебники из раздела 5.

- Дополнительно ознакомьтесь с содержанием онлайн-курсов на таких платформах как: Coursera, Udemу, Stepik.
- Используйте официальную документацию к используемым инструментам и технологиям.
- Читайте популярные источники такие как профильные блоги и статьи: Medium, Habr и т.д.

6.6. Подготовка к зачету

- Повторите все темы, изученные в течение семестра.
- Просмотрите конспекты лекций и отчеты по лабораторным работам.
- Решите тестовые задания и задачи для самоконтроля.
- Подготовьте ответы на возможные вопросы преподавателя.
- При необходимости обратитесь к преподавателю за консультацией.

6.7. Дополнительные рекомендации

- Активно участвуйте в обсуждениях на лекциях и семинарах.
- Задавайте вопросы преподавателю, если что-то непонятно.
- Работайте в команде с другими студентами для обмена опытом и знаниями.
- Используйте возможности онлайн-ресурсов для самостоятельного изучения материала.
- Экспериментируйте и пробуйте новые технологии.

7. Материально-техническое обеспечение по дисциплине (модулю)

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа	Мебель: учебная мебель Технические средства обучения: экран, проектор, компьютер ауд. 129, 131, А-305, А-307	MS Office Word 2016 и выше Ms Power Point 2016 и выше
Учебные аудитории для проведения текущего контроля (Ауд. 101, 102, 105/1, 106 и 106а)	Мебель: учебная мебель Технические средства обучения: Экран, компьютер Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)
Учебные аудитории для проведения промежуточной аттестации (Ауд. 129, 131, А-305, А-307)	Мебель: учебная мебель	-
Учебные аудитории для проведения лабораторных работ (Ауд. 101, 102, 105/1, 106 и 106а)	Мебель: учебная мебель Технические средства обучения: экран, компьютер Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	MS Office Word 2016 и выше Ms Power Point 2016 и выше
Помещение для самостоятельной работы обучающихся (Ауд. 101, 102, 103, 105/1, 106 и 106а)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)	Браузер Google Chrome, Jupyter Notebook 6.3.0 и выше (язык Python с библиотеками Numpy, Pandas, gensim, NLTK, PyMorphy, фреймворком PyTorch)

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.