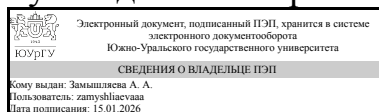


УТВЕРЖДАЮ:
Руководитель направления



А. А. Замышляева

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.18 Современные языки программирования высокого уровня для направления 02.03.02 Фундаментальная информатика и информационные технологии

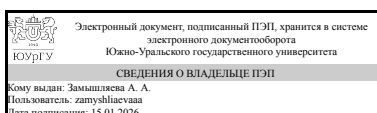
уровень Бакалавриат

форма обучения очная

кафедра-разработчик Центр ОП топ-уровня в сфере ИИ "ВиртУм"

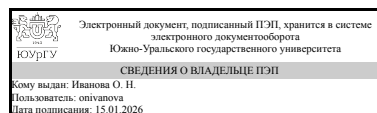
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии, утверждённым приказом Минобрнауки от 23.08.2017 № 808

Зав.кафедрой разработчика,
д.физ.-мат.н., проф.



А. А. Замышляева

Разработчик программы,
к.пед.н., доцент



О. Н. Иванова

1. Цели и задачи дисциплины

Дать студентам системное представление о современных языках программирования (Rust, Go, Kotlin, TypeScript) и их экосистемах, сформировать практические навыки разработки надёжных, производительных и масштабируемых приложений для серверной обработки, распределённых систем, аналитики и фронтенд-интерфейсов для ML-приложений; научить осознанно выбирать язык и инструменты под конкретные инженерные и исследовательские задачи. Задачи дисциплины Обеспечить понимание ключевых концепций и парадигм каждого языка (Rust: безопасность памяти и ownership; Go: простота и конкурентность; Kotlin: JVM-экосистема и функциональные приёмы; TypeScript: типизация и фронтенд-интеграция). Научить проектировать и реализовывать многопоточные и асинхронные системы без ошибок состояния (гонок), использовать корутины и асинхронные конструкции. Ознакомить с экосистемой, инструментами сборки и управления зависимостями, профилирования и отладки для каждого языка. Обучить интеграции с платформами обработки данных и ML (Spark, MLib), созданию API и сетевых сервисов, а также разработке дашбордов и визуализаций для интерпретируемости моделей. Формировать навыки профилирования, оптимизации производительности и мониторинга приложений. Развить умение выбирать язык и архитектурные решения с учётом требований к производительности, безопасности, масштабируемости и скорости разработки.

Краткое содержание дисциплины

Rust: ownership, многопоточность без гонок, экосистема, асинхронизация, профилирование Go: скорость, feature stones, model service, корутины, работа с сетью, управление зависимостями Kotlin: экосистема JVM, функциональное программирование, работа по Spark, интеграция с MLib TypeScript: фронтенд для ML-приложений, дашборды, визуализация данных, интерпретируемость моделей

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	Знает: функциональные возможности интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня, особенности работы компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ, средства мониторинга вычислительных ресурсов компьютерных программ Умеет: использовать возможности современных интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня для разработки алгоритмов и программ, использовать утилиты автоматической сборки и развертывания программ в

	<p>операционных системах Имеет практический опыт: работы с современными интегрированными средами разработки прикладного и системного программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Unix и Windows</p>
<p>ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям</p>	<p>Знает: основные подходы к разработке прикладных алгоритмов в рамках парадигмы структурного программирования на языке высокого уровня, базовые синтаксические конструкции языка программирования высокого уровня: операторы, выражения, блоки, ветвления, циклы; методы оценки сложности алгоритмов, функциональные возможности стандартной библиотеки языка высокого уровня, общие сведения об аппаратных и системных возможностях вычислительной техники для оптимизации программного обеспечения Умеет: разрабатывать прикладное программное обеспечение в рамках парадигмы структурного программирования на языке программирования высокого уровня с применением основных синтаксических конструкций и функциональных возможностей стандартной библиотеки языка высокого уровня Имеет практический опыт: разработки прикладного программного обеспечения, отладки, поиска и устранения ошибок программного кода, оценки сложности алгоритмов, использования возможностей стандартной библиотеки, сторонних библиотек программного кода</p>
<p>ОПК-9 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</p>	<p>Знает: базовые понятия и парадигмы современных языков программирования высокого уровня Умеет: разрабатывать программы с применением различных языков программирования, выбирать подходящие инструменты для конкретной задачи</p>
<p>ОПК-11 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности</p>	<p>Умеет: проектировать архитектуру программного обеспечения, использовать современные языки программирования для разработки программного обеспечения Имеет практический опыт: разработки, тестирования и отладки программ с использованием современных языков программирования, инструментов и технологий</p>
<p>ОПК-14 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем</p>	<p>Умеет: устанавливать и настраивать среду разработки для выбранного языка программирования Имеет практический опыт: настройки и интеграции программных решений с аппаратным обеспечением и внешними устройствами</p>

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
<p>1.О.08 Программирование на Python, 1.О.10 Архитектура вычислительных систем, 1.О.12 Операционные системы, 1.О.11 Программирование на C++, Учебная практика (технологическая, проектно-технологическая) (2 семестр)</p>	<p>1.О.34 Проектирование человеко-машинного интерфейса, 1.О.23 Web-программирование для систем искусственного интеллекта, ФД.02 Визуальное программирование для систем искусственного интеллекта, 1.О.27 Тестирование программного обеспечения, 1.О.32 Высокопроизводительные параллельные вычисления, 1.О.21 Базы данных, ФД.01 Мобильная и веб-разработка систем искусственного интеллекта, Производственная практика (проектно-технологическая, стажировка) (4 семестр)</p>

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
1.О.11 Программирование на C++	<p>Знает: синтаксис языка C++ и технологии разработки прикладного ПО на языке C++, -[И-1, ПУ] средства разработки языка C++ для создания прикладной системы ИИ; библиотеки OpenCV для C++, TensorFlow C++, основные концепции и синтаксис языка программирования C++ Умеет: разрабатывать прикладные программные решения на языке C++, -[И-1, ПУ] использовать средства разработки языка C++ для создания прикладной системы ИИ, стандартные библиотеки C++, реализовывать эффективные алгоритмы и решать практические задачи средствами C++ Имеет практический опыт: создания приложений на языке C++ с соблюдением принципов ООП и code style, -[И-2, ПУ] разработки и отладки прикладных решений на языке программирования C++ с учетом контроля памяти, многопоточности, профилирования кода, высокой производительности, написания программного кода на C++, его отладки и профилирования производительности</p>
1.О.08 Программирование на Python	<p>Знает: -[И-1, БУ] основы синтаксиса языка Python пишет небольшие скрипты для автоматизации ручной работы по обработке небольших объемов данных с помощью встроенных модулей и внешних библиотек (csv, json, requests)[И-2, ПУ] библиотеки машинного обучения, такие как scikit-learn, библиотеки и модули Python для обработки данных, работы с</p>

	<p>файлами, сетевыми взаимодействиями и базами данных Умеет: -[И-2, ПУ] оптимизировать код с использованием библиотек для научных вычислений [И-3, ПУ] применять основные функции фреймворка Pandas, самостоятельно построить процесс обработки больших данных с использованием Airflow, использовать стандартные библиотеки и фреймворки Python для реализации алгоритмов решения прикладных задач Имеет практический опыт: -[И-1, БУ] написания небольших скриптов для автоматизации ручной работы по обработке небольших объемов данных с помощью встроенных модулей и внешних библиотек (csv, json, requests) [И-2, ПУ] использования библиотек машинного обучения, такие как scikit-learn, написания программного кода на Python, отладки и тестирования разработанного программного обеспечения</p>
1.О.10 Архитектура вычислительных систем	<p>Знает: основные понятия и принципы построения вычислительных систем, -[И-1, БУ] архитектуры вычислительных систем, необходимых для проектирования и реализации высокопроизводительных решений в сфере промышленного внедрения систем искусственного интеллекта Умеет: анализировать и выбирать подходящее аппаратное обеспечение для конкретной задачи, -[И-1, БУ] подбирать инструменты и технологии для ресурсного обеспечения систем искусственного интеллекта различных масштабов согласно требованиям проекта Имеет практический опыт: монтажа и настройки серверного оборудования</p>
1.О.12 Операционные системы	<p>Знает: основные концепции современных операционных систем, основные средства, предоставляемые современными операционными системами прикладным программам для решения системных и пользовательских задач, - [И-3, ПУ] основные средства мониторинга и диагностики ОС, основные концепции, принципы, теории и факты, связанные с построением современных операционных систем, структуру современных операционных систем, принципы работы их основных компонентов: ядра, менеджера памяти, подсистемы ввода-вывода, файловой системы Умеет: использовать стандартные инструменты современных ОС при решении задач профессиональной деятельности, устанавливать и настраивать операционную систему, создавать прикладные программы в терминах API ОС, использовать интерфейсы прикладного программирования, предоставляемые современными операционными системами, -[И-1, ПУ] выбирать операционную систему и ее параметры с учетом требований к</p>

	<p>развертыванию и сопровождению моделей искусственного интеллекта в среде эксплуатации, использовать стандартные инструменты современных операционных систем при решении практических задач, использовать стандартные интерфейсы современных операционных систем для решения задач профессиональной деятельности Имеет практический опыт: работы с основными видами интерфейсов ОС - командным и API, использования основных видов интерфейсов операционной системы Windows, создания прикладных программ с использованием API Windows, -[И-2, БУ] реализации скриптов и настройки операционной системы для автоматизации запуска, мониторинга и устойчивой работы сервисов искусственного интеллекта в среде эксплуатации[И-3, ПУ] использования средств мониторинга и диагностики ОС для анализа стабильности и производительности сервисов искусственного интеллекта в среде эксплуатации, работы с основными компонентами современных операционных систем, создания командных файлов, использования API операционных систем при разработке прикладных программ для решения задач профессиональной деятельности</p>
<p>Учебная практика (технологическая, проектно-технологическая) (2 семестр)</p>	<p>Знает: -[И-1, ПУ] цели и задачи командной работы в сфере ИИ, виды коммуникаций и их роль в достижении поставленных целей, этические нормы и установленные правила командной работы, способы первичной обработки информации Умеет: критически оценить эффективность использования времени при решении поставленных задач, а также, относительно полученного результата, -[И-1, ПУ] включаться в состав рабочей группы и активно участвовать в коллективных обсуждениях, использовать математический аппарат в решении профессиональных задач, находить и критически анализировать информацию, необходимую для решения поставленной задачи, разрабатывать математические модели, алгоритмы и компьютерные программы для предложенных задач Имеет практический опыт: оценки личностных ресурсов по достижению целей управления своим временем для успешного выполнения порученной работы и саморазвития, -[И-1, ПУ] публичного выступления, подготовки и демонстрации презентации в рамках своей роли, согласовано с выступлениями других участников команды, программной реализации алгоритмов задач профессиональной деятельности, участия в обмене информацией,</p>

	знаниями и опытом в интересах выполнения командной задачи, декомпозиции поставленной задачи, выделяя её базовые составляющие, разработки компьютерных программ, пригодных для практического использования
--	---

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 3 з.е., 108 ч., 70,25 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах	
		Номер семестра	
		3	
Общая трудоёмкость дисциплины	108	108	
<i>Аудиторные занятия:</i>	64	64	
Лекции (Л)	32	32	
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	16	16	
Лабораторные работы (ЛР)	16	16	
<i>Самостоятельная работа (СРС)</i>	37,75	37,75	
Подготовка к зачету	5	5	
Изучение научных статей	17,75	17,75	
Изучение дополнительных разделов изучаемых тем	15	15	
Консультации и промежуточная аттестация	6,25	6,25	
Вид контроля (зачет, диф.зачет, экзамен)	-	зачет	

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Rust: ownership, многопоточность без гонок, экосистема, асинхронизация, профилирование	16	8	4	4
2	Go: скорость, feature stones, model service, корутины, работа с сетью, управление зависимостями	16	8	4	4
3	Kotlin: экосистема JVM, функциональное программирование, работа по Spark, интеграция с MLib	16	8	4	4
4	TypeScript: фронтенд для ML-приложений, дашборды, визуализация данных, интерпретируемость моделей	16	8	4	4

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Владелец, заимствования и жизненные циклы: основы системы ownership.	2
2	1	Безопасная многопоточность в Rust: Send, Sync и каналы.	2
3	1	Асинхронность в Rust: futures, async/await и runtime.	2

4	1	Инструменты экосистемы и профилирование: Cargo, crates.io, perf, flamegraphs.	2
5	2	Производительность в Go: профилирование, оптимизация сборщика и аллокаций.	2
6	2	Парадигма goroutine и каналов: дизайн concurrent-систем.	2
7	2	Построение model service: сериализация, gRPC/HTTP, контейнеризация.	2
8	2	Управление зависимостями и релизами: modules, vendoring и CI-пайплайны.	2
9	3	Kotlin на JVM: межоперабельность с Java и особенности экосистемы.	2
10	3	Функциональные возможности Kotlin: иммутабельность, коллекции, sequence и inline-функции.	2
11	3	Kotlin + Apache Spark: архитектура, API и практики использования.	2
12	3	Интеграция с MLlib и пайплайнами: от подготовки данных до распределённого обучения.	2
13	4	Архитектура фронтенда для ML-приложений: компоненты, state management и API-контракты.	2
14	4	Визуализация данных: библиотеки (D3, Vega-Lite, Plotly) и проектирование эффективных визуализаций.	2
15	4	Дашборды и интерактивность: реактивные компоненты, streaming data и перформанс.	2
16	4	Интерпретируемость моделей в UI: SHAP/LIME визуализации, объясняющие панели и пользовательские исследования.	2

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Rust: Ownership и безопасная память	4
2	2	Go: Профилирование и оптимизация	4
3	3	Kotlin на JVM и функциональный стиль	4
4	4	TypeScript: Построение ML дашбордов	2
5	4	Решение кейсов от индустриальных партнеров	2

5.3. Лабораторные работы

№ занятия	№ раздела	Наименование или краткое содержание лабораторной работы	Кол-во часов
1	1	Rust; Многопоточность и асинхронность + профилирование	4
2	2	Go: Model service и сеть	4
3	3	Kotlin: Spark + MLlib	4
4	4	TypeScript: Интерпретируемость ML-моделей	2
5	4	Решение кейсов от индустриальных партнеров	2

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Подготовка к зачету	Основные и дополнительная литература	3	5
Изучение научных статей	См. методические указания для СРС	3	17,75
Изучение дополнительных разделов	См. основную литературу, №№ 1-4	3	15

изучаемых тем			
---------------	--	--	--

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-местр	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учитывается в ПА
1	3	Текущий контроль	Практическая работа 01	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
2	3	Текущий контроль	Практическая работа 02	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
3	3	Текущий контроль	Практическая работа 03	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
4	3	Текущий контроль	Практическая работа 04	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
5	3	Текущий контроль	Лабораторная работа 01	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на	зачет

						25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	
6	3	Текущий контроль	Лабораторная работа 02	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
7	3	Текущий контроль	Лабораторная работа 03	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
8	3	Текущий контроль	Лабораторная работа 04	4	4	0 баллов: работа не выполнена 1 балл: работа выполнена не менее чем на 25%, имеются неточности 2 балла: работа выполнена не менее чем на 50%, имеются неточности 3 балла: работа выполнена не менее чем на 75%, имеются неточности 4 балла: работа выполнена полностью, без ошибок и неточностей	зачет
9	3	Текущий контроль	Практическая работа 05 + Лабораторная работа 05	4	4	Работа оценивается представителями индустриального партнера. Баллы: от 0 до 4	зачет
10	3	Текущий контроль	Тест 01	4	4	Компьютерный тест проводится на лекционном занятии. Система выбирает 3 вопроса случайных из базы вопросов по этой теме.	зачет
11	3	Текущий контроль	Тест 02	4	4	Компьютерный тест проводится на лекционном занятии. Система выбирает 3 вопроса случайных из базы вопросов по этой теме.	зачет
12	3	Текущий контроль	Тест 03	4	4	Компьютерный тест проводится на лекционном занятии. Система выбирает 3 вопроса случайных из базы вопросов по этой теме. Лекция проводится в формате класса от представителя индустриального партнера, который рассказывает разработанные преподавателем материалы, но с конкретными примерами из опыта	зачет

						компаний.	
13	3	Текущий контроль	Тест 04	4	4	Компьютерный тест проводится на лекционном занятии. Система выбирает 3 вопроса случайных из базы вопросов по этой теме. Лекция проводится в формате "Лекция вдвоем", два лектора по очереди рассказывают материал.	зачет
14	3	Промежуточная аттестация	Финальный тест	-	100	Финальный тест состоит из 20 вопросов по 5 вопросов из каждого модуля. Максимальная оценка за тест - 100 баллов, которая далее пересчитывается в соответствии с БРС.	зачет
15	3	Бонус	Бонусные баллы	-	15	Бонусы начисляются за посещение занятий и выполнение дополнительных видов работ, предусмотренных рабочей программой (см. методические указания).	зачет

6.2. Процедура проведения, критерии оценивания

Не предусмотрены

6.3. Паспорт фонда оценочных средств

Компетенции	Результаты обучения	№ КМ														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ОПК-2	Знает: функциональные возможности интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня, особенности работы компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ, средства мониторинга вычислительных ресурсов компьютерных программ	++			+++				+++		+		+	+	+	
ОПК-2	Умеет: использовать возможности современных интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня для разработки алгоритмов и программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах	++			+++			++						+		
ОПК-2	Имеет практический опыт: работы с современными интегрированными средами разработки прикладного и системного программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Unix и Windows	++			+++			++						+		
ОПК-3	Знает: основные подходы к разработке прикладных алгоритмов в рамках парадигмы структурного программирования на языке высокого уровня, базовые синтаксические конструкции языка программирования высокого уровня: операторы, выражения, блоки, ветвления, циклы; методы оценки сложности	+		++	++	++	++	++	++			+	+	+	+	

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Методические указания для студентов и преподавателей

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Дополнительная литература	ЭБС издательства Лань	Сомон., П. И. Волшебство Kotlin : руководство / П. И. Сомон. ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2020. — 536 с. — ISBN 978-5-97060-801-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140599 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
2	Основная литература	ЭБС издательства Лань	Петросян, Л. Э. Разработка мобильных приложений на языке Kotlin : учебное пособие для вузов / Л. Э. Петросян, К. В. Гусев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 104 с. — ISBN 978-5-507-52328-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/448577 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
3	Основная литература	ЭБС издательства Лань	Титмус, М. А. Облачный Go / М. А. Титмус ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2021. — 418 с. — ISBN 978-5-97060-965-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/241106 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
4	Дополнительная литература	ЭБС издательства Лань	Батчер, М. Go на практике : руководство / М. Батчер, М. Фарина ; научный редактор А. Н. Киселев ; перевод с английского Р. Н. Рагимова. — Москва : ДМК Пресс, 2017. — 374 с. — ISBN 978-5-97060-477-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/97351 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
5	Дополнительная литература	ЭБС издательства Лань	Марк, С. Программирование на Go. Разработка приложений XXI века : учебное пособие / С. Марк ; перевод с английского А. Н. Киселёв. — Москва : ДМК Пресс, 2013. — 580 с. — ISBN 978-5-94074-854-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/69944 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
6	Основная литература	ЭБС издательства Лань	Сейерс, Э. X. Docker на практике / Э. X. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131719 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
7	Основная литература	ЭБС издательства Лань	Розенталс, Н. Изучаем Typescript 3 / Н. Розенталс ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2019. — 608 с. — ISBN 978-5-97060-757-2. — Текст : электронный // Лань : электронно-библиотечная система. —

			URL: https://e.lanbook.com/book/131712 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
8	Дополнительная литература	ЭБС издательства Лань	Розенталс, Н. Изучаем Typescript 3 / Н. Розенталс ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2019. — 608 с. — ISBN 978-5-97060-757-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131712 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
9	Основная литература	ЭБС издательства Лань	Макиевский, С. Е. Программирование на языке Rust : учебное пособие / С. Е. Макиевский, А. Г. Дворецкий. — Москва : РТУ МИРЭА, 2025 — Часть I — 2025. — 123 с. — ISBN 978-5-7339-2613-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/504872 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
10	Дополнительная литература	ЭБС издательства Лань	Блэнди, Д. Программирование на языке Rust. Быстрое и безопасное системное программирование / Д. Блэнди, Д. Орендорф ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2018. — 550 с. — ISBN 978-5-97060-236-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/112925 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.
11	Дополнительная литература	ЭБС издательства Лань	Дилеман, П. Изучаем Angular 2 / П. Дилеман ; под редакцией А. Н. Киселева ; перевод с английского Р. Н. Рагимова. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-461-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100906 (дата обращения: 24.11.2025). — Режим доступа: для авториз. пользователей.

Перечень используемого программного обеспечения:

Нет

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Не предусмотрено