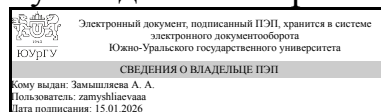


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Руководитель направления



А. А. Замышляева

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.18 Современные языки программирования высокого уровня
для направления 09.03.03 Прикладная информатика

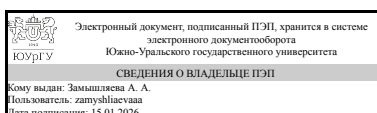
уровень Бакалавриат

форма обучения очная

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

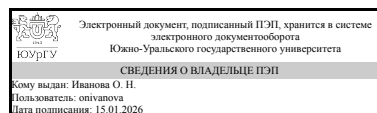
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утверждённым приказом Минобрнауки от 19.09.2017 № 922

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 понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	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
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	Умеет: устанавливать и настраивать среду разработки для выбранного языка программирования Имеет практический опыт: настройки и интеграции программных решений с аппаратным обеспечением и внешними устройствами
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического	Знает: базовые понятия и парадигмы современных языков программирования

применения	высокого уровня Умеет: разрабатывать программы с применением различных языков программирования, выбирать подходящие инструменты для конкретной задачи
------------	--

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
1.О.10 Архитектура вычислительных систем, 1.О.12 Операционные системы, ФД.03 Основы разработки и анализа алгоритмов, Учебная практика (технологическая, проектно-технологическая) (2 семестр)	1.О.27 Тестирование программного обеспечения, 1.О.21 Базы данных, ФД.01 Мобильная и веб-разработка систем искусственного интеллекта, ФД.04 Функциональное и логическое программирование, ФД.02 Визуальное программирование для систем искусственного интеллекта, 1.О.34 Проектирование человеко-машинного интерфейса, Производственная практика (проектно-технологическая, стажировка) (4 семестр)

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

Дисциплина	Требования
1.О.12 Операционные системы	Знает: основные концепции современных операционных систем, -[И-3, ПУ] основные средства мониторинга и диагностики ОС Умеет: использовать стандартные инструменты современных ОС при решении задач профессиональной деятельности, -[И-1, ПУ] выбирать операционную систему и ее параметры с учетом требований к развертыванию и сопровождению моделей искусственного интеллекта в среде эксплуатации, устанавливать и настраивать операционную систему, создавать прикладные программы в терминах API ОС Имеет практический опыт: работы с основными видами интерфейсов ОС - командным и API, -[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, использования основных видов интерфейсов операционной системы Windows
1.О.10 Архитектура вычислительных систем	Знает: -[И-1, БУ] архитектуры вычислительных систем, необходимых для проектирования и реализации высокопроизводительных решений в

	сфере промышленного внедрения систем искусственного интеллекта, основные понятия и принципы построения вычислительных систем Умеет: -[И-1, БУ] подбирать инструменты и технологии для ресурсного обеспечения систем искусственного интеллекта различных масштабов согласно требованиям проекта, анализировать и выбирать подходящее аппаратное обеспечение для конкретной задачи Имеет практический опыт: монтажа и настройки серверного оборудования
ФД.03 Основы разработки и анализа алгоритмов	Знает: основные методы анализа эффективности алгоритмов Умеет: Имеет практический опыт: разработки, анализа и реализации алгоритмов решения прикладных задач
Учебная практика (технологическая, проектно-технологическая) (2 семестр)	Знает: способы первичной обработки информации, этические нормы и установленные правила командной работы Умеет: критически оценить эффективность использования времени при решении поставленных задач, а также, относительно полученного результата, 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, декомпозиции поставленной задачи, выделяя её базовые составляющие, участия в обмене информацией, знаниями и опытом в интересах выполнения командной задачи, 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

Изучение научных статей	17,75	17.75
Подготовка к зачету	5	5
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 MLib 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
Изучение научных статей	См. методические указания для СРС	3	17,75
Подготовка к зачету	Основные и дополнительная литература	3	5
Изучение дополнительных разделов изучаемых тем	См. основную литературу, №№ 1-4	3	15

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

[illegible]

		контроль	работа 04			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	
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 компании.	зачет
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	Умеет: проектировать архитектуру программного обеспечения, использовать современные языки программирования для разработки программного обеспечения	+	+	+	+			+		+			+		+	+
ОПК-2	Имеет практический опыт: разработки, тестирования и отладки программ с использованием современных языков программирования, инструментов и технологий		+	+	+	+		+	+	+					+	+
ОПК-5	Умеет: устанавливать и настраивать среду разработки для выбранного языка программирования	+		+				+		+			+		+	+
ОПК-5	Имеет практический опыт: настройки и интеграции программных решений с аппаратным обеспечением и внешними устройствами		+	+	+	+		+	+	+					+	+
ОПК-7	Знает: базовые понятия и парадигмы современных языков программирования высокого уровня	+								+	+	+		+	+	
ОПК-7	Умеет: разрабатывать программы с применением различных языков программирования, выбирать подходящие инструменты для конкретной задачи		+		+	+			+	+					+	

Типовые контрольные задания по каждому мероприятию находятся в приложениях.

7. Учебно-методическое и информационное обеспечение дисциплины

Печатная учебно-методическая документация

а) *основная литература:*

Не предусмотрена

б) *дополнительная литература:*

Не предусмотрена

в) *отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:*

Не предусмотрены

г) *методические указания для студентов по освоению дисциплины:*

1. Методические указания для студентов и преподавателей

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. Х. Docker на практике / Э. Х. 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