
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Язык программирования Kotlin»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Программная инженерия в
ритейле

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	3
3. Тематический план.....	7
4. Содержание дисциплины (модуля).....	7
5. Учебно-методическое обеспечение	9
6. Материально-техническое обеспечение	9
7. Методические и оценочные материалы	11

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Язык программирования Kotlin» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Программная инженерия в ретейле, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Язык программирования Kotlin» позволяет сформировать способность создавать комплексные приложения на Kotlin (backend, Android, KMP), выстраивать масштабируемую архитектуру и модульную структуру проектов.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Программная инженерия в ретейле и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений как дисциплина по выбору. Дисциплина (модуль) доступна к изучению на 2 курсе в 4 семестре.

Цель изучения дисциплины (модуля): формирование у студентов комплексных знаний и практических навыков разработки приложений на языке Kotlin, включая освоение синтаксиса, принципов асинхронного и функционального программирования, построения масштабируемой архитектуры, создания сервисов, мобильных и кроссплатформенных решений с применением современных инструментов и инженерных практик.

Задачи изучения дисциплины (модуля):

- изучить синтаксис Kotlin, систему типов и механизмы безопасности (null safety, data-классы);
- освоить инструменты сборки Gradle Kotlin DSL и принципы модульной архитектуры проектов;
- научиться применять корутины, диспатчеры и каналы для асинхронного программирования;
- сформировать навыки разработки REST-сервисов на Spring Boot с использованием JPA и асинхронных клиентов;
- овладеть основами Jetpack Compose, Kotlin Multiplatform и практиками тестирования и логирования.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- синтаксис Kotlin, отличие от Java, защита от null значений;
- data-классы, extension-функции, коллекции, функциональный стиль;
- sealed-классы, generics, inline/reified, основы DSL;
- gradle Kotlin DSL, многомодульность;
- основы корутин, диспатчеры, каналы;
- принципы Spring Boot, REST, JPA, асинхронность в Spring;
- основы Compose, Kotlin Multiplatform;
- основы логирования, метрик и тестирования.

уметь:

- писать приложения на Kotlin, используя функциональный и объектный стиль;
- работать с Gradle Kotlin DSL и модульной структурой;
- применять корутины, механизмы синхронизации;
- разрабатывать REST-сервисы и работать с БД;
- делать асинхронные сетевые вызовы через асинхронные клиенты + корутины;
- создавать UI в Jetpack Compose и использовать MVVM;

- настраивать KMP-проекты.

владеть:

- навыком создавать комплексные Kotlin-приложения (backend, Android, KMP);
- навыком проектировать архитектуру и модульную структуру;
- навыком разрабатывать производительные и асинхронные сервисы;
- навыком интегрировать приложения с внешними API и БД;
- навыком обеспечивать наблюдаемость и качество кода, проводить ревью кода.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает современные методы прикладной математики, а также алгоритмические основы решения задач в условиях реальных данных ретейла; владеет инструментами и технологиями, применяемыми для анализа и обработки больших объемов информации (включая отечественные программные средства)
		ОПК-1.2.	Умеет формализовать бизнес-задачи ретейла в виде математических моделей, выбирать и применять алгоритмы и методы решения, интерпретировать результаты и представлять их в форме, доступной для технических и управленческих стейкхолдеров
		ОПК-1.3.	Имеет практический опыт решения прикладных задач в сфере ретейла в рамках учебных, научных или проектных работ, включая обработку реальных данных, разработку аналитических моделей и интеграцию решений в программные системы, а также участие в командных проектах, направленных на повышение эффективности бизнес-процессов
ОПК-3.	Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства	ОПК-3.1.	Знает принципы проектирования и архитектуры программного обеспечения, современные языки программирования, фреймворки и инструменты разработки, а также особенности применения отечественных программных продуктов и платформ в условиях цифровой экономики
		ОПК-3.2.	Умеет разрабатывать,

			тестировать и сопровождать прикладные программные средства, интегрировать их с базами данных, аналитическими системами и внешними сервисами, а также адаптировать программные решения под требования бизнеса в сфере ретейла
		ОПК-3.3.	Имеет практический опыт разработки программных модулей или систем, используемых для реализации математических моделей и анализа данных в ретейле, включая участие в командных проектах, сопровождение кодовой базы, документирование решений и внедрение в производственную среду
ПК-6.	Способен представлять результаты профессиональной деятельности и формулировать практические рекомендации по развитию цифровых решений в ретейле	ПК-6.1.	Знает требования к подготовке технической и аналитической документации, принципы визуализации данных и стандарты профессиональной коммуникации
		ПК-6.2.	Умеет формулировать обоснованные выводы и рекомендации по результатам разработки и анализа, адаптируя их для технической и управленческой аудитории
		ПК-6.3.	Имеет практический опыт подготовки отчетов, презентаций и публичной защиты результатов проектов в области программной инженерии в ретейле

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Основы Kotlin и отличие от Java	4	4		17	Домашние задания
2	Функциональный и идиоматичный Kotlin	2	2		15	Домашние задания
3	Построение DSL и работа с инфраструктурой	5	5		20	Домашние задания
4	Асинхронность и конкурентность: Coroutines	4	4		16	Домашние задания
5	Серверный Kotlin на Spring Boot	5	5		20	Домашние задания
6	Kotlin в мобильной и мультиплатформенно й разработке	5	5		20	Домашние задания
7	Эксплуатация и качество кода	5	5		20	Домашние задания
	Зачет с оценкой			2		
Итого:		30	30	2	128	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы Kotlin и отличие от Java	Kotlin vs Java: синтаксис и защита от null значений Data-классы, extension-функции, коллекции
2	Функциональный и идиоматичный Kotlin	Функциональный стиль
3	Построение DSL и работа с инфраструктурой	Продвинутый Kotlin: Sealed-классы, generics, inline/reified, основы DSL Продвинутый Kotlin DSL и работа с Gradle
4	Асинхронность и конкурентность: Coroutines	Coroutines I: основы Coroutines II: context, отмена, каналы, общее состояние
5	Серверный Kotlin на Spring Boot	Работа с REST в Kotlin с помощью Spring Boot. Работа с базами данных в Kotlin с помощью Spring Boot. Производительный Spring: асинхронность и вызовы внешних API
6	Kotlin в мобильной и мультиплатформенной разработке	Kotlin в Android: Jetpack Compose Kotlin Multiplatform (KMP)
7	Эксплуатация и качество кода	Логирование и мониторинг в Kotlin

		Тестирование: юнит-тесты, мокирование, интеграционные тесты Идиоматичный Kotlin и чистый код
--	--	---

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Аделекан И. Kotlin: программирование на примерах. — СПб. : БХВ-Петербург, 2020. <https://znanium.ru/catalog/document?id=441978>.
2. Сомон П.-И. Волшебство Kotlin / пер. с англ. А. Н. Киселева. — Москва : ДМК Пресс, 2020. <https://znanium.ru/catalog/document?id=358824>.

Дополнительная литература:

1. Жемеров Д., Исакова С. Kotlin в действии / пер. с англ. А. Н. Киселева. — Москва : ДМК Пресс, 2018. <https://znanium.ru/catalog/document?id=483535>.
2. Функциональное программирование. Теоретические и практические основы для разных языков : учебник для вузов / под общ. ред. А. Ю. Анисимова, А. Е. Трубина, Ф. А. Мастяева.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том

числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Язык программирования Kotlin:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое

Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Язык программирования Kotlin» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал, использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Язык программирования Kotlin»

Оценивание уровня учебных достижений обучающихся по дисциплине (модулю) осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в syllabus дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину (модуль). Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если
2	Не сдан	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
1	Не сдан	преподаватель задает дополнительные наводящие вопросы.

Дисциплина (модуль) «Язык программирования Kotlin» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	40%	7	Набор задач по темам недели
Зачет с оценкой	60%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Язык программирования Kotlin»: $0,4 \times \text{среднее за домашние задания} + 0,6 \times \text{зачет с оценкой}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1: Основы Kotlin и функциональный стиль

1. Создайте data-класс Product с полями id, name, price, category и реализуйте copy, equals, hashCode.
2. Напишите extension-функции для String: isValidEmail(), toSlug(), truncate(maxLength: Int).
3. Реализуйте функцию обработки списка продуктов: фильтрация по категории, сортировка по цене, группировка через groupBy.
4. Используйте sealed-классы для моделирования результата операции: Result.Success, Result.Error, Result.Loading.
5. Напишите функцию с использованием inline и reified для безопасного каста и фильтрации коллекции по типу.

Домашнее задание 2: Корутины и асинхронность

1. Реализуйте функцию параллельной загрузки данных из трёх источников с помощью async / await.
2. Напишите сервис с таймаутом на операцию: используйте withTimeout и корректную обработку CancellationException.
3. Создайте Channel-based очередь задач: производитель отправляет задачи, потребители обрабатывают параллельно.
4. Продемонстрируйте разницу между Dispatchers.IO, Dispatchers.Default и Dispatchers.Main на примерах.
5. Реализуйте общее состояние между корутинами через StateFlow и защиту от гонок через Mutex.

Домашнее задание 3: Серверный Kotlin и качество кода

1. Создайте REST API на Spring Boot с эндпоинтами GET /products, POST /products, DELETE /products/{id}.
2. Подключите базу данных через Spring Data JPA: реализуйте репозиторий с методами findByCategory и findByPriceLessThan.
3. Реализуйте асинхронный вызов внешнего API через WebClient с корутинами (suspend fun).
4. Напишите юнит-тесты для сервисного слоя с мокированием зависимостей через

MockK.

5. Проведите рефакторинг кода согласно принципам идиоматичного Kotlin: замените if-else на when, используйте let, run, apply.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Как называется механизм Kotlin, предотвращающий обращение к отсутствующему значению на этапе компиляции?	безопасность типов по null	ОПК-1
2.	Какой оператор используется для возврата значения по умолчанию при отсутствии объекта?	оператор Элвиса	ОПК-1
3.	Как называется класс, автоматически генерирующий методы equals, hashCode и copy?	data-класс	ОПК-3
4.	Какой механизм позволяет расширить функциональность существующего класса без наследования?	функция-расширение	ОПК-3
5.	Как называется ограниченная иерархия типов, все подклассы которой известны на этапе компиляции?	sealed-класс	ОПК-1
6.	Какой подход предполагает передачу функций как аргументов и использование лямбда-выражений?	функциональный стиль	ПК-6
7.	Как называется блок кода, который гарантированно выполнится при работе с ресурсом?	use-блок (автозакрывание ресурса)	ОПК-3
8.	Какой механизм позволяет указать конкретный тип внутри обобщённой функции во время выполнения?	reified-тип	ПК-6
9.	Как называется файл конфигурации сборки проекта на Kotlin DSL?	build.gradle.kts	ОПК-1
10.	Какой принцип проектирования предполагает разделение приложения на независимые модули?	модульная архитектура	ОПК-3
11.	Как называется единица конкурентного выполнения в Kotlin, работающая поверх потоков?	корутина	ПК-6
12.	Какой механизм используется для отмены выполнения корутины?	отмена через контекст	ОПК-3
13.	Как называется структура для передачи данных между параллельными корутинами?	канал обмена	ПК-6
14.	Какой тип потока данных хранит последнее состояние и уведомляет подписчиков?	поток состояния	ОПК-3
15.	Как называется архитектурный стиль взаимодействия клиента и сервера через HTTP?	REST	ОПК-1
16.	Какой компонент Spring отвечает за обработку HTTP-запросов?	REST-контроллер	ОПК-3
17.	Как называется слой приложения, отвечающий за доступ к данным через JPA?	репозиторий	ПК-6
18.	Какой принцип позволяет выполнять неблокирующие сетевые вызовы?	асинхронное программирование	ПК-6

19.	Как называется декларативный инструмент построения пользовательского интерфейса в Android?	Compose	ОПК-1
20.	Какой подход к организации кода предполагает разделение на модель, представление и модель представления?	архитектура MVVM	ОПК-3