
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Разработка на языке программирования Kotlin»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Машинное обучение

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2025

**Москва
2025**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	5
3. Тематический план.....	5
4. Содержание дисциплины (модуля).....	7
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	8
7. Методические и оценочные материалы	10

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка на языке программирования Kotlin» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по специальности 02.04.01 Математика и компьютерные науки, профиль Машинное обучение, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение Kotlin как современного языка программирования открывает перед студентами возможности создания высокопроизводительных и безопасных приложений, а также способствует улучшению качества кода благодаря лаконичности и выразительности синтаксиса. Кроме того, знание Kotlin расширяет профессиональные горизонты выпускников, так как этот язык активно используется в разработке на платформе Android и в серверной разработке, что делает их более конкурентоспособными на рынке труда.

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 Машинное обучение и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре, доступна для прохождения при условии успешного завершения дисциплин (модулей) «Java Core (Основы разработки на языке Java)», «Java Spring (Разработка веб-приложений на Java с использованием Spring)», «Многопоточное программирование», «Инструменты разработчика», «SQL и базы данных для разработчиков».

Цель изучения дисциплины (модуля): формирование у студентов глубоких знаний и практических навыков разработки серверных приложений с использованием языка Kotlin.

Задачи изучения дисциплины (модуля):

- изучить основы синтаксиса и идиом Kotlin для написания читаемого и эффективного кода;
- понять принципы асинхронного программирования через корутины для управления параллельными задачами;
- освоить разработку гибкой бизнес-логики с применением инструментов Kotlin в реальных проектах;
- научиться реализовывать многопоточные и асинхронные операции с использованием корутин для повышения производительности;
- овладеть навыками интеграции Kotlin с фреймворками, такими как Spring и Ktor, для создания масштабируемых серверных решений.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- синтаксис Kotlin, отличие от Java, null safety;
- data-классы, extension-функции, коллекции, функциональный стиль;
- sealed-классы, generics, inline/reified, основы DSL;
- gradle Kotlin DSL, многомодульность;
- основы корутин, диспатчеры, channels;
- принципы Spring Boot, REST, JPA/Exposed, асинхронность в Spring;
- основы Compose, MVVM, 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, suspend-функции, механизмы синхронизации;
- разрабатывать REST-сервисы и работать с БД;
- делать асинхронные сетевые вызовы через WebClient и корутины;
- создавать UI в Jetpack Compose и использовать MVVM;
- настраивать KMP-проекты.

владеть:

- навыками создания комплексных Kotlin-приложений (backend, Android, KMP);
- навыками построения архитектуры и модульной структуры;
- навыками разработки производительных и асинхронных сервисов;
- навыками интеграции приложений с внешними API и БД;
- навыками обеспечения наблюдаемости и качества кода, проведения code review.

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)
УК-6.	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1.	Знает основные методы самооценки и анализа своей деятельности, а также принципы управления временем и целеполагания
		УК-6.2.	Умеет ставить реалистичные и достижимые цели, определять приоритеты в своей деятельности, а также разрабатывать и внедрять планы по совершенствованию своих навыков и компетенций на основе полученной самооценки
		УК-6.3.	Имеет практический опыт применения методов самооценки в своей профессиональной деятельности, включая участие в тренингах, семинарах и проектах, направленных на развитие личной эффективности и профессионального роста
ОПК-2.	Способен создавать и исследовать новые математические модели в естественных науках, совершенствовать и разрабатывать концепции, теории и методы	ОПК-2.1.	Знает основные математические модели и методы, используемые в естественных науках, включая статистическое моделирование, дифференциальные уравнения и численные методы, а также современные подходы к исследованию и анализу данных
		ОПК-2.2.	Умеет разрабатывать и адаптировать математические модели для решения конкретных проблем в естественных науках, проводить их анализ и верификацию, а также интерпретировать полученные результаты в контексте научных исследований
		ОПК-2.3.	Имеет практический опыт создания и исследования математических моделей в рамках научных проектов или

			исследований, включая участие в публикациях, конференциях или коллаборациях, где были разработаны и апробированы новые концепции и методы
ПК-3.	Способен решать задачи профессиональной деятельности, формулировать результат, увидеть следствия полученного результата	ПК-3.1.	Знает основные принципы и методы решения задач профессиональной деятельности, а также способы формулирования и представления результатов, включая анализ последствий и их значимость в контексте проекта
		ПК-3.2.	Умеет применять математические и компьютерные методы для решения конкретных задач, формулировать четкие и обоснованные результаты, а также анализировать их последствия для дальнейших действий и решений
		ПК-3.3.	Имеет практический опыт в решении профессиональных задач, включая участие в проектах, где были получены результаты и проанализированы их следствия, что способствовало принятию обоснованных решений
ПК-4.	Способен публично представлять собственные и известные научные результаты	ПК-4.1.	Знает основные принципы эффективного публичного выступления, методы визуализации данных и основные требования к научным презентациям, включая структуру и содержание
		ПК-4.2.	Умеет четко и логично формулировать свои научные результаты, адаптируя их для различных аудиторий, а также использовать визуальные средства для улучшения восприятия информации
		ПК-4.3.	Имеет практический опыт участия в научных конференциях, семинарах или других мероприятиях, где успешно представлял свои и известные научные результаты, получая обратную связь и взаимодействуя с аудиторией

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Основы Kotlin и отличие от Java	4	4		7	Домашнее задание
2	Функциональный и идиоматичный Kotlin	2	2		4	Домашнее задание
3	Построение DSL и работа с инфраструктурой	4	4		7	Домашнее задание
4	Асинхронность и конкурентность: Coroutines	4	4		7	Домашнее задание
5	Серверный Kotlin на Spring Boot	6	6		9	Домашнее задание
6	Kotlin в мобильной и мультиплатформенной разработке	4	4		7	Домашнее задание
7	Эксплуатация и качество кода	6	6		9	Домашнее задание
	<i>Зачет</i>			4		
	Итого:	30	30	4	50	
	Объем дисциплины (модуля) (в ак. ч.)	114				
	Объем дисциплины (модуля) (в зач. ед.)	3				

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 null safety 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, launch/async, suspend Coroutines II: context, отмена, channels, shared state
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: JUnit5, MockK, 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 : практическое руководство / И. Аделекан. - Санкт-Петербург : БХВ-Петербург, 2020. - 432 с. - ISBN 978-5-9775-6673-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2139144>.

2. Макгрегор, Д. От Java к Kotlin : практическое руководство / Д. Макгрегор, Н. Прайс. - Санкт-Петербург : БХВ-Петербург, 2023. - 448 с. - ISBN 978-5-9775-6841-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2122955>.

Дополнительная литература:

1. Гриффитс, Д. Head First. Kotlin : практическое руководство / Д. Гриффитс, Д. Гриффитс. - Санкт-Петербург : Питер, 2020. - 464 с. - (Серия «Head First O'Reilly»). - ISBN 978-5-4461-1335-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1733498>.

2. Функциональное программирование. Теоретические и практические основы для разных языков : учебник для вузов / под общей редакцией А. Ю. Анисимова, А. Е. Трубина, Ф. А. Мастяева. — Москва : Издательство Юрайт, 2025. — 135 с. — (Высшее образование). — ISBN 978-5-534-20518-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/558300>.

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
Операционные системы:		
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) «Разработка на языке 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).

Электронный документ

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) «Разработка на языке 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*, 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:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	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	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	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,
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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.
5	Удовлетворительно	Зачтено	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	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Разработка на языке программирования Kotlin» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	40%	За каждое из заданий можно набрать 10 баллов
Зачет	60%	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на языке программирования Kotlin»: $\langle 0,4 \times \text{среднее за домашние задания} + 0,6 \times \text{зачет} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание

Перепишите небольшой Java-класс (например, модель пользователя с валидацией и коллекциями) на Kotlin, примените null safety, используйте data-класс для хранения данных, реализуйте не менее двух extension-функций для работы с коллекциями, продемонстрируйте различия синтаксиса и подходов Kotlin и Java и оформите краткое сравнение ключевых отличий.

Электронный документ

Домашнее задание

Разработайте модуль обработки списка объектов (например, заказов или событий), примените функции высшего порядка, map/filter/reduce, lambdas и score-функции, реализуйте цепочку преобразований данных в функциональном стиле без изменяемого состояния и обоснуйте выбор идиоматичных конструкций Kotlin.

Домашнее задание

Разработайте небольшой DSL на Kotlin (например, для конфигурации сервиса или описания HTTP-роутов), примените sealed-классы для описания состояний, используйте generics и inline/reified-функции для безопасной типизации, подключите проект к Gradle Kotlin DSL и продемонстрируйте пример использования созданного DSL в рабочем коде.

Домашнее задание

Реализуйте асинхронный сервис на Kotlin с использованием coroutines, создайте задачи через launch и async, опишите suspend-функции для имитации сетевых запросов, добавьте управление контекстом и отменой операций, реализуйте взаимодействие через channels, обеспечьте корректную работу с общим состоянием и продемонстрируйте обработку конкурентных сценариев.

Домашнее задание

Разработайте REST-сервис на Kotlin с использованием Spring Boot, реализуйте CRUD-операции для одной предметной сущности, подключите базу данных (PostgreSQL или H2) через Spring Data, добавьте валидацию входных данных, реализуйте асинхронный вызов внешнего API с использованием suspend-функций или WebClient, обеспечьте обработку ошибок и продемонстрируйте базовые показатели производительности сервиса.

Домашнее задание

Разработайте простое Android-приложение с использованием Jetpack Compose, реализуйте экран со списком данных и навигацией, вынесите бизнес-логику в общий модуль Kotlin Multiplatform, обеспечьте повторное использование кода между платформами и продемонстрируйте архитектурное разделение UI и доменной логики.

Домашнее задание

Добавьте в существующий Kotlin-проект структурированное логирование и базовые метрики, реализуйте unit-тесты с использованием JUnit5 и MockK, напишите интеграционные тесты для ключевых сценариев, проведите рефакторинг кода с применением идиоматичных конструкций Kotlin и принципов чистого кода и опишите улучшения в качестве и поддерживаемости проекта.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой из следующих операторов используется для безопасной работы с null в Kotlin? A) !!	В	ПК-3

	В) ?. С) ?: D) !!?		
2.	Что такое let в Kotlin? A) Функция для создания классов B) Функция расширения для работы с null C) Лямбда-функция для выполнения операций над объектами D) Способ создания коллекций	С	ПК-3
3.	Что такое coroutine в Kotlin? A) Способ работы с потоками B) Способ асинхронного программирования C) Функция для управления памятью D) Метод обработки ошибок	В	ПК-3
4.	Укажите элемент самооценки в работе с веб-фреймворками в Kotlin.	Логирование	УК-6
5.	Назовите технику совершенствования деятельности в разработке на Kotlin.	Рефакторинг	УК-6
6.	Что такое null-безопасность в Kotlin?	Защита от null-значений	ОПК-2
7.	Назовите ключевое слово для объявления функции-расширения в Kotlin.	fun	ПК-3
8.	Какой паттерн используется для реализации асинхронного программирования в Kotlin?	Корутины	ОПК-2
9.	Какой фреймворк часто используется для создания RESTful API на Kotlin?	Ktor	ОПК-2
10.	Укажите способ определения приоритетов в оптимизации кода на Kotlin.	Профилирование	УК-6
11.	Какой тип коллекции в Kotlin используется для хранения уникальных значений?	Set	ПК-3
12.	Назовите метод самооценки навыков в работе с корутинами в Kotlin.	Тестирование	УК-6
13.	Назовите формат презентации результатов проекта на Kotlin.	Демо	ПК-4
14.	Какой метод используется для запуска корутины?	launch	ПК-3
15.	Укажите способ представления научных результатов в разработке на Kotlin.	Доклад	ПК-4
16.	Назовите ключевую особенность синтаксиса Kotlin для безопасности типов.	null safety	ОПК-2
17.	Назовите инструмент Kotlin для разработки бизнес-логики.	data classes	ПК-3
18.	Назовите элемент эффективной презентации кода на Kotlin.	Визуализация	ПК-4
19.	Назовите механизм для моделирования потоков данных в Kotlin.	channels	ОПК-2
20.	Назовите протокол для асинхронного транспорта в Kotlin.	WebSocket	ПК-3