

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

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

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Backend-разработка

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2024

**Москва
2024**

Содержание

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

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 Backend-разработка, 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 Backend-разработка и входит в часть, формируемую участниками образовательных отношений Блока 1, является дисциплиной по выбору.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре, является дисциплиной по выбору.

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

Задачи изучения дисциплины (модуля):

- формирование знаний синтаксиса и идиом Kotlin;
- формирование знаний асинхронности через корутины;
- формирование умения разрабатывать гибкую и надёжную бизнес-логику, используя инструменты Kotlin;
- формирование умения применять асинхронное и многопоточное программирование с корутинами;
- формирование умения реализовывать асинхронный транспорт с применением WebSocket, RabbitMQ и Kafka;
- формирование навыка использовать язык Kotlin для разработки серверных приложений;
- формирование навыка использовать Kotlin вместе со Spring и Ktor.

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-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-6.	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере профессиональной деятельности	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в конкретной предметной области
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Аудиторная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Основы и идиоматический Kotlin	7	7		10	Домашнее задание
2	Серверные паттерны и асинхронность	8	8		14	Домашнее задание
3	Работа с данными и инфраструктурой	8	8		14	Домашнее задание Тест
4	Продвинутые темы	7	7		12	Домашнее задание Проект
	<i>Зачет с оценкой</i>			4		
	Итого:	30	30	4	50	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	114				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	3				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы и идиоматический Kotlin	Базовые конструкции и null-безопасность. Функциональные идиомы. Расширения и делегирование
2	Серверные паттерны и асинхронность	ООП и модели данных. Коллекции и производительность. Корутины и Flow
3	Работа с данными и инфраструктурой	Работа с БД. Веб-фреймворки (Ktor/Spring Boot). Тестирование и MockK. Безопасность и оптимизация
4	Продвинутые темы	Миграция Java → Kotlin. Kotlin в микросервисах. Финальный проект

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. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебник для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2025. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567946>.

Дополнительная литература:

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2022. — 1120 с. — ISBN 978-5-4461-1155-8.

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;

— 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 к 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	https://elibrary.ru/defaultx.asp

	библиотека	
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, тест, проект, 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.

Участие в семинаре (аудиторная работа) – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Тест – особая форма проверки знаний. Проводится после освоения одной или нескольких тем и свидетельствует о качестве понимания основных понятий изучаемого материала. Тестовые задания составлены к ключевым понятиям, основным разделам, важным терминологическим категориям изучаемой дисциплины (модуля).

Для подготовки к тесту необходимо знать терминологический аппарат дисциплины (модуля), понимать смысл научных категорий и уметь их использовать в профессиональной лексике. Владение понятийным аппаратом, включённым в тестовые задания, позволяет преподавателю быстро проверить уровень понимания студентами важных методологических категорий.

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) «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 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:

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
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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.
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	50%	За каждое из заданий можно набрать 10 баллов
Тест	10%	Ответы на вопросы, по изученным темам
Проект	40%	Исследовательская работа по дисциплине (модулю) и презентация результатов

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на языке программирования Kotlin»: « $0,5 \times \text{среднее за домашние задания} + 0,1 \times \text{тест} + 0,4 \times \text{проект}$ ».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание: Основы и идиоматический Kotlin

1. Напишите функцию, которая принимает строку и возвращает ее длину. Используйте null-безопасность, чтобы обработать случай, когда строка может быть null.

- **Пример:** `fun getStringLength(input: String?): Int`

2. Создайте класс **Person** с полями **name** и **age**. Реализуйте метод, который возвращает строку в формате "Имя: [name], Возраст: [age]".

- **Пример:** `class Person(val name: String, val age: Int)`

3. Используйте функциональные идиомы Kotlin, чтобы написать функцию, которая принимает список чисел и возвращает их сумму, используя **fold**.

- **Пример:** `fun sumNumbers(numbers: List<Int>): Int`

4. Реализуйте расширение для класса **String**, которое будет возвращать строку в верхнем регистре, если она не пустая, и "Пустая строка" в противном случае.

- **Пример:** `fun String.toUpperOrEmpty(): String`

5. Создайте класс **DelegateExample**, который будет использовать делегирование для хранения значения. Реализуйте делегат, который будет выводить значение при его изменении.

- **Пример:** `var delegatedProperty: String by Delegates.observable("initial") { _, old, new -> println("Changed from $old to $new") }`

Домашнее задание: Серверные паттерны и асинхронность

1. Определите интерфейс **DataModel**, который будет содержать методы для получения и сохранения данных. Реализуйте класс **User**, который будет реализовывать этот интерфейс.

- **Пример:** `interface DataModel { fun save(): Boolean fun load(): DataModel }`

2. Создайте класс **User Repository**, который будет использовать коллекцию для хранения объектов **User**. Реализуйте методы для добавления и получения пользователей.

- **Пример:** `class UserRepository { private val users = mutableListOf<User>() }`

3. Напишите функцию, которая использует корутины для асинхронного получения данных из сети. Используйте **withContext** для переключения контекста.

- **Пример:** `suspend fun fetchData(): String`

4. Создайте класс **FlowExample**, который будет использовать **Flow** для генерации последовательности чисел от 1 до 10. Реализуйте метод, который будет выводить эти числа.

- **Пример:** `fun numberFlow(): Flow<Int>`

5. Опишите, как реализовать паттерн "Singleton" в Kotlin, и создайте класс **DatabaseConnection**, который будет являться синглтоном.

- **Пример:** `object DatabaseConnection { /* реализация */ }`

Домашнее задание: Работа с данными и инфраструктурой

1. Создайте класс **Database**, который будет содержать методы для подключения к базе данных и выполнения SQL-запросов. Реализуйте метод для получения всех пользователей.

- **Пример:** `class Database { fun connect() { /* реализация */ } }`

2. Напишите простое приложение на Ktor, которое будет обрабатывать HTTP GET запросы и возвращать список пользователей в формате JSON.

- **Пример:** `fun Application.module() { routing { get("/users") { /* реализация */ } } }`

3. Создайте тест для класса **Database**, используя MockK для имитации работы с базой данных. Реализуйте тест на получение данных.

- **Пример:** `@Test fun testGetUsers() { /* реализация */ }`

4. Опишите, как реализовать безопасность в приложении на Spring Boot, используя аннотации для защиты эндпоинтов.

○ **Пример:** `@PreAuthorize("hasRole('ADMIN')")`

5. Напишите функцию, которая будет оптимизировать запросы к базе данных, используя индексы. Объясните, как индексы влияют на производительность.

○ **Пример:** `fun optimizeQuery(query: String) { /* реализация */ }`

Примерные тестовые задания

Тест: Работа с БД и веб-фреймворками

1. Какой из следующих фреймворков используется для создания веб-приложений на Kotlin?

- A) Django
- B) Spring Boot
- C) Laravel
- D) Express

2. Какой аннотацией в Spring Boot можно защитить метод, чтобы только пользователи с ролью "ADMIN" могли его вызывать?

- A) @Secured
- B) @RolesAllowed
- C) @PreAuthorize
- D) @PermitAll

3. Что такое ORM?

- A) Object-Relational Mapping
- B) Online Resource Management
- C) Object-Resource Management
- D) Open Resource Model

4. Какой из следующих методов используется для выполнения SQL-запросов в Spring Data JPA?

- A) executeQuery()
- B) findAll()
- C) getResult()
- D) selectAll()

5. Какой из следующих компонентов Ktor отвечает за маршрутизацию запросов?

- A) Application
- B) Routing
- C) Pipeline
- D) Engine

6. Какой метод используется для создания теста с использованием MockK?

- A) mock()
- B) createMock()
- C) newMock()
- D) buildMock()

7. Что такое "Dependency Injection"?

- A) Метод управления состоянием приложения
- B) Метод для создания тестов

- C) Метод, позволяющий внедрять зависимости в классы
- D) Метод для работы с базами данных

8. Какой из следующих подходов используется для тестирования контроллеров в Spring Boot?

- A) Unit Testing
- B) Integration Testing
- C) Functional Testing
- D) All of the above

9. Какой из следующих методов используется для настройки соединения с базой данных в Spring Boot?

- A) application.properties
- B) config.xml
- C) database.yml
- D) settings.json

10. Какой из следующих инструментов используется для миграции схемы базы данных в Spring Boot?

- A) Flyway
- B) Liquibase
- C) Hibernate
- D) А и В

11. Что такое "SQL Injection"?

- A) Метод оптимизации запросов
- B) Уязвимость, позволяющая злоумышленнику выполнять произвольные SQL-запросы
- C) Способ защиты базы данных
- D) Метод работы с транзакциями

12. Какой из следующих методов Ktor используется для обработки HTTP POST запросов?

- A) post()
- B) get()
- C) request()
- D) handle()

13. Какой аннотацией в Spring Boot можно пометить метод, который будет обрабатывать все исключения?

- A) @ControllerAdvice
- B) @ExceptionHandler
- C) @RestController
- D) @GlobalExceptionHandler

14. Какой из следующих инструментов используется для создания моков в тестах на Kotlin?

- A) Mockito
- B) JUnit
- C) MockK
- D) EasyMock

15. Что такое "Rate Limiting"?

- A) Метод оптимизации производительности базы данных

- B) Метод ограничения числа запросов к API в единицу времени
- C) Метод защиты от SQL Injection
- D) Метод управления сессиями пользователей

Примерное задание для проекта

Тема проекта:

Миграция существующего Java-приложения на Kotlin с использованием микросервисной архитектуры.

Требования к проекту:

1. **Исходный код:** Необходимо выбрать существующее Java-приложение (например, простую систему управления задачами или интернет-магазин) и предоставить его исходный код.
2. **Микросервисы:** Приложение должно быть переработано в микросервисную архитектуру, где каждый сервис отвечает за отдельную функциональность (например, пользовательский сервис, сервис задач, сервис оплаты).
3. **Kotlin:** Все микросервисы должны быть переписаны на Kotlin, включая использование Kotlin Coroutines для асинхронного программирования.
4. **Документация:** Проект должен содержать документацию, описывающую архитектуру, использованные технологии и процесс миграции.
5. **Тестирование:** Каждый микросервис должен иметь набор юнит-тестов и интеграционных тестов, написанных с использованием Kotlin и подходящих библиотек (например, JUnit, MockK).
6. **CI/CD:** Настройка непрерывной интеграции и развертывания (CI/CD) для автоматизации тестирования и развертывания микросервисов.

Задание:

1. Выберите существующее Java-приложение, которое вы будете мигрировать на Kotlin.
2. Проанализируйте архитектуру приложения и определите, как его можно разбить на микросервисы.
3. Перепишите каждый сервис на Kotlin, используя соответствующие библиотеки и фреймворки (например, Ktor, Spring Boot).
4. Реализуйте асинхронное программирование с использованием Kotlin Coroutines.
5. Напишите тесты для каждого микросервиса и настройте CI/CD для автоматизации процессов.
6. Подготовьте документацию о процессе миграции и архитектуре нового приложения.

Этапы выполнения:

1. **Этап 1: Исследование и планирование**
 - Выбор приложения для миграции.
 - Анализ текущей архитектуры и определение микросервисов.
2. **Этап 2: Миграция первого микросервиса**
 - Переписывание первого микросервиса на Kotlin.

- Реализация тестов и настройка CI/CD.
3. **Этап 3: Миграция остальных микросервисов**
 - Переписывание оставшихся микросервисов.
 - Написание тестов и настройка CI/CD для каждого сервиса.
 4. **Этап 4: Документация и финальная проверка**
 - Подготовка документации.
 - Защита проекта перед аудиторией.

Критерии оценивания:

1. **Качество кода (30%)**
 - Чистота и читаемость кода.
 - Следование стандартам Kotlin и принципам SOLID.
2. **Функциональность (30%)**
 - Полнота функционала, реализованного в микросервисах.
 - Корректность работы всех сервисов.
3. **Тестирование (20%)**
 - Наличие и полнота тестов.
 - Уровень покрытия кода тестами.
4. **Документация (10%)**
 - Полнота и ясность документации.
 - Описание архитектуры и процесса миграции.
5. **CI/CD (10%)**
 - Настройка автоматизации тестирования и развертывания.
 - Корректность работы CI/CD пайплайна.

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) !! B) ?. C) ?: D) !!?	В	ПК-3
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.	Какой из следующих паттернов используется для работы с асинхронными данными в Kotlin? A) Singleton B) Observer C) Flow D) Factory	С	ПК-6
4.	Какой из следующих фреймворков является веб-фреймворком для Kotlin? A) Django B) Spring Boot C) Ktor D) Express	С	ПК-6
5.	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
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.	Что такое MockK?	Библиотека для мокирования в тестах	ПК-6
11.	Какой тип коллекции в Kotlin используется для хранения уникальных значений?	Set	ПК-3
12.	Какой оператор используется для обработки null-значений с помощью значения по умолчанию?	?:	ПК-6
13.	Что такое data class в Kotlin?	Класс для хранения данных	ПК-6
14.	Какой метод используется для запуска корутины?	launch	ПК-3
15.	Какой подход используется для миграции Java-кода на Kotlin?	Пошаговая рефакторинг	ПК-6