

УТВЕРЖДЕНА

Приказом Ректора АНО ВО
«Центральный университет»
Е.В. Ивашкевич
от «26» июня 2025 г. № 0626.32

**Рабочая программа дисциплины (модуля)
«Разработка на языке программирования Go»
дополнительной профессиональной программы – программы
профессиональной переподготовки «Академия data science»**

Траектория: Backend-разработка

**Москва
2025**

Содержание

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

1. Краткая характеристика дисциплины (модуля)

Изучение языка Go позволяет слушателям эффективно разрабатывать высокопроизводительные и масштабируемые приложения, что особенно актуально в условиях растущей популярности облачных технологий и микросервисной архитектуры.

Цель изучения дисциплины (модуля): формирование у слушателей навыков проектирования, разработки и оптимизации серверных приложений и микросервисов с использованием языка Go, а также освоение современных подходов к программированию и управлению проектами.

Задачи изучения дисциплины (модуля):

- изучить фундаментальные конструкции языка Go, такие как синтаксис, типы данных и структуры, для формирования основы программирования;
- освоить механизмы конкурентного выполнения и сетевого взаимодействия в Go для построения эффективных многопоточных приложений;
- научиться применять методы тестирования и отладки кода при разработке веб-сервисов и решении прикладных задач.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- основы Go: синтаксис, типы данных, структуры;
- принципы конкурентности и работы с сетью;
- основы тестирования.

уметь:

- писать, отлаживать и структурировать код;
- разрабатывать и тестировать веб-сервисы.

владеть:

- навыками использования языка Go как инструмента для решения прикладных задач;
- навыками создания эффективных и надежных сервисов.

2. 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 программирования на Go	3	9		48	Домашние задания Контрольная работа	
2	Практическое применение Go для решения прикладных задач	3	10		50	Домашние задания Контрольная работа	
3	Реализация сервисов на Go	3	10		50	Домашние задания Контрольная работа Итоговый проект	
	<i>Зачет с оценкой</i>			4			
	Итого:	9	29	4	148		
	Объем дисциплины (модуля) (в ак. ч.)	190					

3. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы программирования на Go	Основы языка, базовые конструкции и типы данных. Массивы и слайсы Концепция указателей, карты. Структуры и интерфейсы
2	Практическое применение Go для решения прикладных задач	Юнит-тестирование. Функции стандартной библиотеки. Теги структур и JSON. Структура проекта и использование сторонних библиотек
3	Реализация сервисов на Go	Горутины и каналы. Контексты и Graceful Shutdown. Протокол HTTP и реализация REST-сервера. Интеграционное тестирование сервисов. Использование шаблонов и обработка Web-форм. Введение в SQL и хранение данных в БД

4. Учебно-методическое обеспечение

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

Каждый слушатель 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.

Слушателям 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.

5. 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

6. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка на языке программирования Go» 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 слушателя на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре слушателям 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.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы – получить специальные знания по одной или нескольким темам дисциплины и продемонстрировать навыки их практического применения.

Электронный документ

Проект – исследовательская работа по дисциплине (модулю) 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.

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 слушателей, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы слушатели взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи слушателя 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)

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*.

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	Слушатель 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. Слушатель хорошо знаком с методами исследования, необходимыми
9	Отлично	Зачтено	
8	Отлично	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Зачтено	Слушатель 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. Слушатель хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	Зачтено	
5	Удовлетворительно	Зачтено	Слушатель 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. Слушатель способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Слушатель 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 Go» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	40%	За каждое из заданий можно набрать 10 баллов
Контрольные работы	60%	Письменная работа с набором задач, которые нужно решить за ограниченное время

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

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание

- 1. Напишите программу на Go, которая принимает на вход два числа и выводит их сумму, разность, произведение и частное.**
 - Убедитесь, что программа обрабатывает случаи деления на ноль.
- 2. Создайте функцию, которая принимает строку и возвращает количество гласных и согласных букв в этой строке.**
 - Используйте базовые конструкции языка для подсчета.
- 3. Реализуйте программу, которая принимает целое число и определяет, является ли оно четным или нечетным.**
 - Используйте условные операторы для выполнения проверки.
- 4. Создайте программу, которая использует switch для определения дня недели по номеру (1-7).**
 - Выведите название дня недели для введенного номера.
- 5. Напишите программу, которая создает переменные разных типов (int, float64, string) и выводит их значения и типы на экран.**
 - Используйте встроенные функции для определения типа переменной.

Домашнее задание

- 1. Создайте массив из 10 целых чисел и напишите функцию, которая находит и возвращает максимальное значение в этом массиве.**
 - Используйте цикл для итерации по элементам массива.
- 2. Реализуйте программу, которая создает слайс строк и добавляет в него элементы, введенные пользователем.**
 - Позвольте пользователю вводить строки до тех пор, пока он не введет пустую строку.
- 3. Создайте структуру Person с полями Name и Age. Напишите функцию, которая принимает указатель на Person и увеличивает возраст на 1.**
 - Выведите информацию о человеке до и после изменения возраста.
- 4. Реализуйте map, где ключом будет строка (имя), а значением — структура Person. Напишите функцию, которая принимает имя и возвращает информацию о человеке.**
 - Обработайте случай, когда имя отсутствует в карте.
- 5. Создайте интерфейс Shape с методом Area(), и реализуйте две структуры: Circle и Rectangle, которые реализуют этот интерфейс.**
 - Напишите функцию, которая принимает Shape и выводит площадь фигуры.

Домашнее задание

- 1. Напишите юнит-тест для функции, которая вычисляет факториал числа.**
 - Используйте пакет testing для написания тестов.
- 2. Создайте структуру с тегами для JSON и напишите функцию, которая сериализует и десериализует эту структуру.**
 - Проверьте корректность работы с JSON на примере.
- 3. Реализуйте простую горутину, которая выводит числа от 1 до 5 с задержкой в 1 секунду.**
 - Используйте каналы для синхронизации завершения работы горутин.

4. **Создайте HTTP-сервер, который обрабатывает GET-запросы и возвращает JSON-ответ с приветствием.**

- Реализуйте Graceful Shutdown для корректного завершения сервера.

5. **Используйте стороннюю библиотеку (например, gorilla/mux) для создания RESTful API с одним маршрутом, который возвращает информацию о пользователе.**

- Обработайте запросы и верните данные в формате JSON.

Примерные задания для контрольных работ

Контрольная работа 1

1. Какие ключевые слова используются для объявления переменных в Go? Приведите пример объявления переменной типа int.

2. Объясните разницу между коротким объявлением переменной (:=) и полным (var). Приведите пример для каждого.

3. Какие базовые типы данных существуют в Go? Назовите три примера и их назначение.

4. Напишите программу на Go, которая выводит "Hello, World!" с использованием функции fmt.Println().

5. Чем отличается массив от слайса в Go? Приведите пример создания массива и слайса.

6. Напишите код, который создает слайс целых чисел, добавляет к нему элементы и выводит его длину.

7. Как работает функция append() для слайсов? Приведите пример добавления элемента в слайс.

8. Объясните, что такое capacity слайса. Как ее можно увеличить?

9. Что такое указатель в Go? Приведите пример создания указателя на переменную типа int.

10. Напишите код, который использует mapu для хранения пар "ключ-значение" (строка-целое число) и выводит значение по ключу.

11. Чем мапа отличается от слайса? Приведите пример инициализации мапы.

12. Как проверить наличие ключа в мапе? Напишите соответствующий код.

13. Что такое структура в Go? Приведите пример определения и инициализации структуры с двумя полями.

14. Объясните понятие интерфейса. Приведите пример интерфейса и его реализации.

15. Напишите код, который определяет структуру "Person" с полями Name и Age, создает экземпляр и выводит его.

Контрольная работа 2

1. Что такое юнит-тестирование в Go? Приведите пример простой тестовой функции.

2. Как запустить тесты в Go с помощью команды go test? Опишите основные флаги.

3. Напишите тест для функции, которая складывает два числа, используя пакет testing.

4. Объясните, что такое table-driven tests в Go. Приведите краткий пример.

5. Назовите три функции из пакета strings и опишите их назначение.

6. Как работает функция `fmt.Sprintf()`? Приведите пример ее использования.
7. Напишите код, который использует пакет `time` для получения текущего времени и форматирования его в строку.
8. Объясните разницу между функциями `os.Open()` и `ioutil.ReadFile()`. Когда использовать каждую?
9. Что такое теги структур в Go? Приведите пример структуры с JSON-тегами.
10. Как сериализовать структуру в JSON с помощью пакета `encoding/json`? Напишите код.
11. Напишите код для десериализации JSON-строки в структуру с полями `Name` и `Age`.
12. Объясните, зачем нужны теги в структурах при работе с JSON.
13. Как организовать структуру проекта в Go (например, с использованием модулей)? Опишите основные файлы.
14. Как установить стороннюю библиотеку в Go-проекте с помощью `go get`? Приведите команду.
15. Напишите код, который импортирует и использует функцию из сторонней библиотеки (например, для HTTP-запросов).

Контрольная работа 3

1. Что такое горутина в Go? Приведите пример запуска горютины.
2. Как работают каналы для коммуникации между горютинами? Приведите простой пример с `chan int`.
3. Напишите код, который запускает две горютины, отправляющие данные в канал, и читает из него в главной функции.
4. Объясните разницу между буферизованным и небуферизованным каналами.
5. Что такое контекст в Go и зачем он нужен? Приведите пример создания контекста с таймаутом.
6. Как реализовать Graceful Shutdown сервера в Go? Опишите шаги с использованием `signal.Notify()`.
7. Напишите код для обработки сигнала `SIGINT` и graceful завершения работы сервера.
8. Объясните роль контекста в отмене операций в горютинах.
9. Как создать простой HTTP-сервер в Go с помощью пакета `net/http`? Приведите пример маршрута.
10. Что такое REST API? Напишите код для обработки GET-запроса, возвращающего JSON.
11. Объясните, как обрабатывать параметры пути в HTTP-запросах (например, `/users/{id}`).
12. Напишите код для POST-запроса, который принимает JSON и сохраняет данные в памяти.
13. Чем отличается интеграционное тестирование от юнит-тестирования? Приведите пример сценария.
14. Как протестировать HTTP-сервер в Go с использованием `httptest`? Напишите простой тест.
15. Напишите тест, который отправляет запрос к REST-серверу и проверяет ответ.
16. Объясните, зачем использовать мок-объекты в интеграционных тестах.
17. Как работать с HTML-шаблонами в Go с помощью пакета `html/template`? Приведите пример рендеринга шаблона.

18. Напишите код для обработки формы, отправленной методом POST, и извлечения данных.
19. Объясните безопасность при работе с шаблонами (например, защита от XSS).
20. Как передать данные из Go в HTML-шаблон? Приведите пример с `map[string]interface{}`.

Примерное задание для проекта

Описание проекта:

Создайте веб-приложение для управления задачами, которое позволит пользователям добавлять, обновлять, удалять и просматривать свои задачи. Приложение должно использовать горутины и каналы для обработки запросов, контексты для управления временем выполнения и Graceful Shutdown для безопасного завершения работы сервера. Данные о задачах должны храниться в базе данных SQL, а интерфейс должен поддерживать обработку веб-форм и использование шаблонов.

Требования к проекту:

1. Функциональность:

- Пользователь должен иметь возможность:
 - Создавать новую задачу (с заголовком и описанием).
 - Просматривать список всех задач.
 - Обновлять существующую задачу (изменять заголовок и описание).
 - Удалять задачу.
- Все операции должны быть асинхронными и обрабатываться с использованием горутин и каналов.

2. HTTP-сервер:

- Реализуйте RESTful API для работы с задачами.
- Используйте стандартный пакет `net/http` для обработки запросов.
- Все запросы должны возвращать JSON-ответы.

3. Контексты и Graceful Shutdown:

- Используйте контексты для управления временем выполнения запросов.
- Реализуйте Graceful Shutdown, чтобы сервер корректно завершал работу при получении сигнала остановки.

4. Интеграционное тестирование:

- Напишите интеграционные тесты для проверки работы вашего REST API.
- Убедитесь, что все тесты проходят успешно.

5. Шаблоны и обработка веб-форм:

- Реализуйте веб-формы для создания и обновления задач.
- Используйте HTML-шаблоны для отображения данных.

6. Хранение данных:

- Используйте SQL (например, SQLite или PostgreSQL) для хранения информации о задачах.

- Реализуйте CRUD-операции для взаимодействия с базой данных.

Этапы выполнения:

1. Планирование и проектирование:

- Определите структуру данных для задач.
- Спроектируйте API и маршруты для обработки запросов.

2. Настройка окружения:

- Создайте новый проект Go и настройте зависимости.
- Настройте базу данных и создайте таблицы для хранения задач.

3. Реализация функциональности:

- Реализуйте RESTful API для задач.
- Реализуйте обработку веб-форм и шаблонов.

4. Обработка асинхронных запросов:

- Используйте горутины и каналы для обработки запросов.
- Реализуйте контексты для управления временем выполнения.

5. Тестирование:

- Напишите интеграционные тесты для проверки функциональности API.
- Проверьте работу приложения с различными сценариями.

6. Документация:

- Подготовьте документацию по API.
- Опишите, как развернуть и использовать ваше приложение.

Критерии оценивания:

1. Функциональность (40%):

- Все требования к функциональности выполнены.
- Приложение работает без ошибок.

2. Качество кода (30%):

- Код структурирован и читабелен.
- Используются лучшие практики Go (например, обработка ошибок, использование контекстов).

3. Тестирование (20%):

- Написаны интеграционные тесты.
- Все тесты проходят успешно.

4. Документация (10%):

- Документация по API ясна и полна.
- Инструкции по развертыванию приложения понятны.

Дополнительные рекомендации:

- Используйте сторонние библиотеки, если это уместно (например, gorilla/mux для маршрутизации).
- Обратите внимание на обработку ошибок и ведение логов.
- Рассмотрите возможность использования Docker для развертывания приложения.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ
1.	Какой из следующих типов данных является ссылочным в Go? A) int B) string C) map D) float64	C
2.	Какой из следующих типов данных является ссылочным в Go? A) & B) * C) @ D) #	A
3.	Какой из следующих типов данных позволяет хранить набор значений одного типа? A) array B) struct C) interface D) map	A
4.	Что такое теги структур в Go? A) Синтаксические конструкции для определения типов B) Метаданные, добавляемые к полям структур C) Специальные функции для работы с массивами D) Комментарии к коду	B
5.	Какой из следующих пакетов используется для юнит-тестирования в Go? A) testing B) assert C) mock D) testify	A
6.	Назовите ключевое слово, используемое для определения структуры в Go.	struct
7.	Какой тип данных используется для создания ассоциативных массивов в Go?	map
8.	Что такое указатель в Go?	Переменная, хранящая адрес другой переменной
9.	Какой метод используется для кодирования данных в формате JSON?	json.Marshal

10.	Назовите стандартный пакет, используемый для работы с HTTP-запросами.	net/http
11.	Какой тип данных используется для представления функции в Go?	Функция/function
12.	Как называется механизм, позволяющий выполнять функции параллельно в Go?	Горутины /goroutines
13.	Назовите метод, который используется для создания нового HTTP-сервера.	ListenAndServe
14.	Структура, используемая для передачи дедлайнов, отмены и значений между функциями в Go?	контекст
15.	Какой SQL-запрос используется для выборки данных из таблицы?	SELECT
16.	Укажите количество байт, которое занимает тип int64 в Go.	8/ Восемь/ восемь
17.	Укажите тип данных в Go, который представляет последовательность байтов.	string
18.	Укажите ключевое слово в Go для объявления функции.	func
19.	Укажите способ запуска горутины в Go с помощью ключевого слова.	go
20.	Укажите метод для сериализации структуры в JSON в пакете encoding/json.	Marshal