

---

**УТВЕРЖДЕНА**

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

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

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

### **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 навыков проектирования, разработки и оптимизации серверных приложений и микросервисов с использованием языка Go, а также освоение современных подходов к программированию и управления проектами.

### **Задачи изучения дисциплины (модуля):**

- формирование знаний основ Go: синтаксис, типы данных, структуры;
- формирование знаний принципов конкурентности и работы с сетью;
- формирование знаний основ тестирования и работы с базами данных;
- формирование умения писать, отлаживать и структурировать код;
- формирование умения разрабатывать и тестировать веб-сервисы;
- формирование навыка создания эффективных и надежных приложений;
- формирование навыка интеграции сервисов и баз данных.

## 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

| №<br>п/п | Наименование<br>раздела дисциплины<br>(модуля)          | Трудоемкость, академические часы |           |          |                        | ТКУ<br>(текущий<br>контроль<br>успеваемости) |
|----------|---|----------------------------------|-----------|----------|------------------------|--|
|          |   | <i>Очная форма</i>               |           |          |                        |  |
|          |   | Аудиторная работа                |           | Контроль | Самостоятельная работа |  |
| Лекции   | Семинары<br>(практические занятия)                      |                                  |           |          |                        |  |
| 1        | Основы программирования на Go                           | 10                               | 10        |          | 16                     | Домашнее задание                             |
| 2        | Практическое применение Go для решения прикладных задач | 10                               | 10        |          | 18                     | Домашнее задание<br>Тест                     |
| 3        | Реализация сервисов на Go                               | 10                               | 10        |          | 16                     | Домашнее задание<br>Проект                   |
|          | <i>Зачет с оценкой</i>                                  |                                  |           | 4        |                        |  |
|          | <b>Итого:</b>   | <b>30</b>                        | <b>30</b> | <b>4</b> | <b>50</b>              |  |
|          | <b>Объем дисциплины (модуля) (в ак. ч.)</b>             | <b>114</b>                       |           |          |                        |  |
|          | <b>Объем дисциплины (модуля) (в зач. ед.)</b>           | <b>3</b>                         |           |          |                        |  |

### 4. 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<br>Концепция указателей, мапы. 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 |

## 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<br>(издания, курса, документа) | Ссылка  |
|----|---|---|
| 1. | Научная электронная библиотека elibrary.ru          | <a href="https://elibrary.ru/defaultx.asp">https://elibrary.ru/defaultx.asp</a> |

|    |  |   |
|----|--|---|
|    | библиотека   |   |
| 2. | База данных для IT-специалистов  | <a href="https://habr.com">https://habr.com</a>                                 |
| 3. | База данных ScienceDirect  | <a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>       |
| 4. | Официальный сайт Министерства науки и высшего образования Российской Федерации | <a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>           |
| 5. | Федеральный портал «Российское образование»                                    | <a href="https://www.edu.ru/">https://www.edu.ru/</a>                           |
| 6. | Информационная система "Единое окно доступа к образовательным ресурсам"        | <a href="http://window.edu.ru/">http://window.edu.ru/</a>                       |
| 7. | Единая коллекция цифровых образовательных ресурсов                             | <a href="http://school-collection.edu.ru/">http://school-collection.edu.ru/</a> |
| 8. | Федеральный центр информационно - образовательных ресурсов                     | <a href="http://fcior.edu.ru/">http://fcior.edu.ru/</a>                         |

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 |
|---|---------------|--|
| <b>Операционные системы:</b>  |               |  |
| Microsoft Imagine (Windows Client, Server)                          | зарубежное    | лицензионное                             |
| <b>Браузеры:</b>  |               |  |
| Яндекс.Браузер  | отечественное | свободно распространяемое                |
| Google Chrome   | зарубежное    | свободно распространяемое                |
| <b>Офисные приложения:</b>  |               |  |
| Microsoft Imagine (Visio, OneNote)                                  | зарубежное    | лицензионное                             |
| TeXstudio   | зарубежное    | свободно распространяемое                |
| Adobe Acrobat Reader  | зарубежное    | свободно распространяемое                |
| <b>Программное обеспечение для планирования и учета времени:</b>    |               |  |
| Toggle app  | зарубежное    | свободно распространяемое                |
| <b>Системы управления проектами:</b>                                |               |  |
| Microsoft Imagine (Project)   | зарубежное    | лицензионное                             |
| <b>Системы управления базами данных:</b>                            |               |  |
| Microsoft Imagine (SQL Server)                                      | зарубежное    | лицензионное                             |
| <b>Системы резервного копирования (backup):</b>                     |               |  |
| Acronis Backup Advanced for HyperV                                  | зарубежное    | лицензионное                             |
| <b>Справочно-правовые системы:</b>                                  |               |  |
| КонсультантПлюс: справочно-правовая система                         | отечественное | лицензионное                             |
| <b>Средства антивирусной защиты:</b>                                |               |  |
| Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition | отечественное | лицензионное                             |
| <b>Среды разработки:</b>  |               |  |
| 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                |
| <b>Пакеты программных средств и библиотек:</b>                      |               |  |
| AutoPsy   | зарубежное    | свободно распространяемое                |

|  |            |                           |
|--|------------|---------------------------|
| Interactive Disassembler (IDA)                           | зарубежное | свободно распространяемое |
| <b>Системы управления библиографической информацией:</b> |            |                           |
| Zotero   | зарубежное | свободно распространяемое |
| <b>Сервисы и службы:</b>                                 |            |                           |
| 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 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 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 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.

*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 Go»**

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   |

| Десятибалльная оценка | Пятибалльная оценка | Оценка за зачет | Общая характеристика результата обучения по дисциплине (модулю)   |
|-----------------------|---------------------|-----------------|---|
| 6                     | Хорошо              | Зачтено         | 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. |
| 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 Go» 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 Go»:** « $0,5 \times$  среднее за домашние задания +  $0,1 \times$  тест +  $0,4 \times$  проект».

## Текущий контроль успеваемости обучающихся по дисциплине (модулю)

### Примерные домашние задания

#### Домашнее задание: Основы языка, базовые конструкции и типы данных

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: Массивы и слайсы, концепция указателей, мапы, структуры и интерфейсы

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 мапу, 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 мапе.
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: Юнит-тестирование, Graceful Shutdown, протокол HTTP и реализация REST-сервера

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 тестовые задания

#### Тест: Юнит-тестирование и стандартные библиотеки в Go

1. Что из следующего является правильным способом написания юнит-теста в Go?

- A) `func TestFunctionName() { ... }`
- B) `func Test_function_name() { ... }`
- C) `func testFunctionName() { ... }`
- D) `func TestFunctionName() -> { ... }`

2. Какой пакет используется для написания тестов в Go?

- A) `fmt`
- B) `testing`
- C) `test`
- D) `assert`

3. Какой метод используется для проверки ошибок в юнит-тестах?

- A) `CheckError()`
- B) `t.Error()`
- C) `t.Fail()`
- D) `t.Assert()`

4. Какой тег используется для указания того, что поле структуры должно быть сериализовано в JSON?

- A) `json:"fieldName"`
- B) `json.fieldName`
- C) `json:"fieldName" optional`
- D) `fieldName:"json"`

5. Какой метод стандартной библиотеки Go используется для чтения JSON-данных из файла?

- A) `json.ReadFile()`
- B) `json.Unmarshal()`
- C) `json.Decode()`
- D) `json.Load()`

6. Какой из следующих пакетов можно использовать для работы с HTTP-запросами в Go?

- A) `http`
- B) `net/http`
- C) `http/client`
- D) `http/request`

7. Что такое "теги" в структуре Go?

- A) Специальные метаданные, которые описывают поведение поля
- B) Ключевые слова, используемые для импорта пакетов
- C) Комментарии к полям структуры
- D) Ссылки на другие структуры

**8. Какой метод используется для сериализации структуры в JSON?**

- A) json.Marshal()
- B) json.Encode()
- C) json.Serialize()
- D) json.ToJSON()

**9. Какой пакет можно использовать для создания моков в юнит-тестах?**

- A) mock
- B) gomock
- C) testify
- D) mockery

**10. Какой из следующих вариантов является правильной структурой проекта Go?**

- A) src/project\_name/
- B) project\_name/src/
- C) project\_name/
- D) src/

**11. Какой метод используется для работы с каналами в Go?**

- A) channel.Send()
- B) channel.Receive()
- C) channel <- value
- D) channel(value)

**12. Какой из следующих пакетов является сторонней библиотекой для работы с HTTP в Go?**

- A) http
- B) gin
- C) net/http
- D) httpclient

**13. Какой способ позволяет импортировать сторонние библиотеки в проект Go?**

- A) import "library\_name"
- B) require "library\_name"
- C) use "library\_name"
- D) include "library\_name"

**14. Какой метод используется для десериализации JSON в структуру?**

- A) json.Unmarshal()
- B) json.Decode()
- C) json.Deserialize()

D) json.Load()

**15. Что происходит, если в тесте используется t.Fatal()?**

- A) Тест продолжает выполнение
- B) Тест завершается и выводит сообщение об ошибке
- C) Тест игнорируется
- D) Тест переходит к следующему случаю

### 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  | Компетенция |
|-------|---|--|-------------|
| 1.    | Какой из следующих типов данных является ссылочным в Go?<br><br>A) int<br>B) string<br>C) map<br>D) float64   | С  | ОПК-2       |
| 2.    | Какой из следующих типов данных является ссылочным в Go?<br><br>A) &<br>B) *<br>C) @<br>D) #  | А  | ОПК-2       |
| 3.    | Какой из следующих типов данных позволяет хранить набор значений одного типа?<br><br>A) array<br>B) struct<br>C) interface<br>D) map  | А  | ОПК-2       |
| 4.    | Что такое теги структур в Go?<br><br>A) Синтаксические конструкции для определения типов<br>B) Метаданные, добавляемые к полям структур<br>C) Специальные функции для работы с массивами<br>D) Комментарии к коду | В  | ПК-3        |
| 5.    | Какой из следующих пакетов используется для юнит-тестирования в Go?<br><br>A) testing<br>B) assert<br>C) mock<br>D) testify   | А  | ПК-3        |
| 6.    | Назовите ключевое слово, используемое для определения структуры в Go.   | struct                                       | ОПК-2       |
| 7.    | Какой тип данных используется для создания ассоциативных массивов в Go?   | map  | ОПК-2       |
| 8.    | Что такое указатель в Go?   | Переменная, хранящая адрес другой переменной | ОПК-2       |

|     |   |                     |       |
|-----|---|---------------------|-------|
| 9.  | Какой метод используется для кодирования данных в формате JSON?                         | json.Marshal        | ОПК-2 |
| 10. | Назовите стандартный пакет, используемый для работы с HTTP-запросами.                   | net/http            | ПК-6  |
| 11. | Какой тип данных используется для представления функции в Go?                           | Функция/function    | ПК-3  |
| 12. | Как называется механизм, позволяющий выполнять функции параллельно в Go?                | Горутины/goroutines | ПК-3  |
| 13. | Назовите метод, который используется для создания нового HTTP-сервера.                  | ListenAndServe      | ПК-3  |
| 14. | Структура, используемая для передачи дедлайнов, отмены и значений между функциями в Go? | контекст            | ПК-6  |
| 15. | Какой SQL-запрос используется для выборки данных из таблицы?                            | SELECT              | ПК-6  |