Общество с ограниченной ответственностью «Ребреин»
ИНН 7727409582, ОГРН 1197746106161
Адрес: 123056, город Москва, Большая Грузинская ул, д. 36а стр. 5а, офис 13

Утверждено
Приказом № ПР-1 от 17.06.2025 г.

Генеральный директор

_____ Фролкина Е.А.
«17» июня 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
– ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ GO»

Срок реализации: 1 месяц
Количество часов: 106 акад. ч.
Форма обучения: заочная форма
Формат обучения: с применением исключительно дистанционных технологий
Возраст обучающихся: для лиц старше 17 лет, имеющих или получающих среднее профессиональное и (или) высшее образование

Москва
2025 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая дополнительная профессиональная программа – программа повышения квалификации «Введение в программирование на языке Go» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Профессиональным стандартом 06.026 «Системный администратор информационно-коммуникационных систем», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 26.10.2020 года № 60580;
- ФГОС высшего образования – бакалавриат по направлению 09.03.02 Информационные системы и технологии, утв. приказом Минобрнауки России от 19.09.2017 №926;
- Локальными нормативными актами ООО «Ребреин».

В данной программе учтены основные идеи формирования универсальных учебных действий учащихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

Направленность программы: Программа имеет техническую направленность.

Адресат:

Программа предназначена для:

- начинающих разработчиков, желающих освоить язык Go с нуля;
- специалистов из смежных областей (тестировщиков, аналитиков, администраторов), планирующих перейти в разработку;
- студентов технических направлений, интересующихся современными языками программирования;
- разработчиков, владеющих другими языками (C, Java, Python), и желающих расширить стек за счёт Go.

Требования к входным знаниям обучающегося:

Для освоения программы слушателям рекомендуется:

- базовое понимание принципов алгоритмизации и структур данных;
- минимальные навыки работы с операционной системой (Linux или Windows);
- опыт использования командной строки и работы с файлами;
- умение пользоваться системами контроля версий (желательно, Git).

Актуальность реализации:

Язык Go, разработанный компанией Google, является одним из самых востребованных инструментов для создания современных веб-сервисов, облачных решений и высокопроизводительных систем. Его отличают простота синтаксиса, встроенные средства параллельных вычислений и высокая скорость работы.

С учётом активного распространения Go в компаниях, связанных с DevOps, backend-разработкой и высоконагруженными сервисами, владение данным языком становится важным конкурентным преимуществом на рынке труда. Программа даёт

систематизированное введение в Go, позволяя слушателям перейти от базового уровня к написанию асинхронных приложений, работе с тестированием и профилированием.

Отличительные особенности программы:

- обучение начинается «с нуля», что делает программу доступной для новичков;
- последовательное погружение: от базовых конструкций до асинхронности, тестирования и кодогенерации;
- акцент на практических заданиях и написании реального кода с первого модуля;
- разбор типичных ошибок начинающих разработчиков и способов их предотвращения;
- включение современных инструментов разработки (pprof, GoMock, errgroup и др.);
- гибкая структура с опциональными темами, которые можно изучать по мере готовности.

Объем и срок освоения программы: 106 академ. ч. в течение 2 мес. (8 недель).

Доступ к материалам Программы у обучающихся остаётся и после окончания периода обучения. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

Выдаваемый документ о квалификации: удостоверение о повышении квалификации и/или сертификат об успешном освоении программы.

Цели и задачи программы:

Сформировать у слушателей базовые знания и практические навыки программирования на языке Go, необходимые для разработки приложений, включающих обработку ошибок, работу с асинхронностью, тестирование и профилирование.

Программа направлена на решение следующих основных задач:

Обучающие задачи:

- освоить синтаксис языка Go и базовые конструкции;
- изучить работу с модулями, пакетами и зависимостями;
- сформировать понимание принципов асинхронного программирования на Go (goroutines, каналы, контекст);
- познакомиться с методами тестирования, бенчмарками и инструментами профилирования;
- освоить основы кодогенерации и работу с рефлексией.

Развивающие задачи:

- развить навыки алгоритмического и структурного мышления;
- научиться анализировать и оптимизировать собственный код;
- сформировать умение работать с современными инструментами Go и применять их в реальных проектах;
- стимулировать интерес к дальнейшему изучению языка и смежных технологий.

Воспитательные задачи:

- сформировать ответственное отношение к качеству кода и его тестируемости;
- развить культуру командной разработки через использование Git и модульного подхода;

- привить привычку к аккуратной обработке ошибок и безопасному программированию;
- способствовать развитию самостоятельности и профессиональной инициативы.

Планируемые результаты:

Знания:
- базовый синтаксис языка Go, основные типы данных и конструкции;
- особенности работы с указателями, слайсами, map-ами;
- принципы работы с модулями, пакетами и зависимостями (go mod, layout проекта);
- основы структур, интерфейсов и композитного наследования;
- ключевые механизмы асинхронного программирования в Go (goroutines, каналы, context, sync, atomic);
- методы тестирования, подходы к бенчмаркингу и профилированию кода;
- основы кодогенерации, использование AST, рефлексии и шаблонов.

Умения:
- писать и отлаживать простые программы на Go;
- использовать переменные, функции, конструкции управления и обработку ошибок;
- работать с модулями и зависимостями в Go, правильно организовывать структуру проекта;
- применять структуры и интерфейсы для решения практических задач;
- использовать механизмы асинхронности и параллельных вычислений;
- писать unit-тесты, использовать моки и стабы, проводить профилирование и бенчмарки;
- применять инструменты кодогенерации и решать задачи с использованием рефлексии.

Навыки:
- написание корректного, читаемого и поддерживаемого кода на Go;
- работа с системами контроля версий (GitHub/GitLab) в процессе разработки;
- обработка ошибок и предотвращение deadlocks при работе с асинхронностью;
- организация тестового покрытия и оценка производительности кода;
- использование стандартных инструментов Go для разработки и отладки;
- применение принципов структурного и модульного проектирования.

Перечень профессиональных компетенций, на получение которых направлено обучение:
На основе профстандарта 06.026 «Системный администратор информационно-коммуникационных систем»:
- B/02.5 Обеспечение работы технических и программных средств информационно-коммуникационных систем;
- C/05.6 Выполнение обновления программного обеспечения сетевых устройств информационно-коммуникационных систем;
- C/08.6 Планирование и проведение работ по распределению нагрузки между имеющимися ресурсами, снятию нагрузки на сетевые устройства информационно-коммуникационных систем перед проведением регламентных работ, восстановлению штатной схемы работы в случае сбоев.

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;
- ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;
- ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.

Организационно-педагогические условия реализации программы дополнительного профессионального образования

Язык реализации образовательной программы: обучение проводится на русском языке.

Форма обучения: заочная форма.

Особенности реализации программы: программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

Условия набора: на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

Формы проведения занятий:
- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа с литературой;
- индивидуальные вопросы.

Материально-техническое оснащение

Материальное обеспечение программы

Занятия проводятся в системе дистанционного обучения «Rebrain». Каждый обучающийся и педагог оснащены доступом к системе дистанционного обучения: https://rebrainme.com/.

У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

Методическое обеспечение программы

Программа обеспечена:
- учебно-методическими материалами (текстовые занятия, полезными материалами);
- практическими заданиями.

Кадровое обеспечение:

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

1) отвечающее одному из требований:

а) имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования «Образование и педагогические науки»;

б) имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Ребреин», и получение при необходимости дополнительного профессионального образования педагогической направленности;

в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности дополнительной общеобразовательной общеразвивающей программе;

2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;

3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

2. УЧЕБНЫЙ ПЛАН

№ п/п	Наименование модуля	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1	Модуль 1. Онбординг	2	1	1	Входное тестирование
2	Модуль 2. Введение. Подготовка окружения.	8	3	5	Практическое задание
3	Модуль 3. Основы языка Go для начинающих	18	5	13	Практическое задание
4	Модуль 4. Модули и пакеты	16	5	11	Практическое задание
5	Модуль 5. Структуры и интерфейсы	16	4	12	Практическое задание
6	Модуль 6. Асинхронность	17	3	14	Практическое задание
7	Модуль 7. Тестирование, бенчмарки и профилирование	12	4	8	Практическое задание
8	Модуль 8. Кодогенерация	9	4	5	Практическое задание

9	Итоговая аттестация	8		8	Итоговое практическое задание

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 неделя	2 неделя	3 неделя	4 неделя	5 неделя	6 неделя	7 неделя	8 неделя
1	Модуль 1. Онбординг	2							
2	Модуль 2. Введение. Подготовка окружения.	8							
3	Модуль 3. Основы языка Go для начинающих	3	15						
4	Модуль 4. Модули и пакеты			14	2				
5	Модуль 5. Структуры и интерфейсы				13	3			
6	Модуль 6. Асинхронность					10	7		
7	Модуль 7. Тестирование, бенчмарки и профилирование						6	6	
8	Модуль 8. Кодогенерация							7	2
9	Итоговая аттестация								8 | А

4. РАБОЧАЯ ПРОГРАММА

Модуль 1. Онбординг
Теория 1 академ. ч. Практика 1 академ. ч.
Модуль состоит из следующих тем:

Тема 1: Онбординг

В модуле обучающемуся предоставляется вводный конспект, содержащий общую информацию о программе, структуре курса, форматах взаимодействия с материалами и ожидаемых результатах обучения.

Предусмотрено прохождение входного тестирования, включающего 7 вопросов, направленных на закрепление информации из онбординга. В рамках темы обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы,

указывает желаемые навыки по окончании обучения, а также оценивает текущий уровень своих знаний, выбрав один из предложенных вариантов.

Модуль 2. Введение. Подготовка окружения.
Теория 3 академ. ч. Практика 5 академ. ч.
Модуль состоит из следующих тем:

Тема 1. Введение. Подготовка окружения
Содержание: Проблематика, которую решает появление Go, и пути его эволюции как языка. Области применения языка Go. Установка компилятора Go для различных ОС. Возможные среды разработки (IDE). Запуск первой программы на Go. Практическое задание.

Модуль 3. Основы языка Go для начинающих
Теория 5 академ. ч. Практика 13 академ. ч.
Модуль состоит из следующих тем:

Тема 1. Переменные. Типы данных
Содержание: Переменные: объявление и работа со значениями. Константы и их отличие от переменных. Типы переменных. Массивы для работы с однотипными объектами. Особенности области видимости переменных в Go. Практическое задание.

Тема 2. Указатели в Go
Содержание: Указатели в Go: реализация и работа. Извлечение значения по ссылке. Понятие nil. Выделение памяти под значение. Практическое задание.

Тема 3. Слайсы
Содержание: Отличия слайса от массива. Устройство слайса и связь с указателями. Особенности выделения памяти при увеличении размера слайса. Практическое задание.

Тема 4. Map-ы в Go
Содержание: Ассоциативные массивы (map) в Go. Отличия от слайсов и массивов. Возможности и применение. Практическое задание.

Тема 5. Конструкции языка и функции
Содержание: Конструкции языка Go. Функции и их вариации. Организация кода через изолированные блоки ответственности. Практическое задание.

Тема 6. defer - обработка выхода из функции
Содержание: Гарантированное выполнение действий при завершении функции. Работа с вызовом defer. Практическое задание.

Тема 7. Panic и их обработка
Содержание: Обработка критических ошибок. Использование panic. Отличия от стандартной обработки ошибок. Практическое задание.

Тема 8. Обработка ошибок
Содержание: Ошибки в Go. Правильные подходы к обработке ошибок. Практическое задание.

Модуль 4. Модули и пакеты
Теория 5 академ. ч. Практика 11 академ. ч.
Модуль состоит из следующих тем:

Тема 1. Области видимости, инициализация через init()
Содержание: Понятие пакета в Go. Импорт пакетов. Экспортируемые и неэкспортируемые сущности. Виды импорта пакетов. Практическое задание.

Тема 2. Работа с зависимостями, go mod
Содержание: Работа с зависимостями в Go. Модульный подход. Возможности инструмента go mod. Практическое задание.

Тема 3. Создание модулей и их версионирование
Содержание: Оформление собственного модуля. Правила версионирования. Публикация собственного модуля. Практическое задание.

Тема 4. layout проекта (структура проекта)
Содержание: project-layout. Практическое задание.

Модуль 5. Структуры и интерфейсы
Теория 4 академ. ч. Практика 12 академ. ч.
Модуль состоит из следующих тем:

Тема 1.Структуры в GO
Содержание: Создание составных сущностей: структур. Особенности инициализации структур. Области видимости структур. Практическое задание.

Тема 2. Методы структур
Содержание: Привязка поведения к структуре через методы. Особенности объявления методов по ссылке и по значению. Практическое задание.

Тема 3. Интерфейсы и утиная типизация
Содержание: Реализация интерфейсов в Go. Принцип утиной типизации. Объявление интерфейсов и реализация их поведения. Практическое задание.

Тема 4. Пустой интерфейс
Содержание: Понятие пустого интерфейса. Приведение пустого интерфейса к любому типу. Пустая структура и её использование. Практическое задание.

Тема 5. Композитное наследование
Содержание: Детали реализации наследования в Go. Встраивание других структур и интерфейсов. Практическое задание.

Тема 6. Продвинутая работа с ошибками
Содержание: Объект error в Go. Генерация и обработка ошибок. Приведение ошибок к нужному типу. Практическое задание.

Модуль 6. Асинхронность
Теория 3 академ. ч. Практика 14 академ. ч.
Модуль состоит из следующих тем:

Тема 1. Gorountines
Содержание: Реализация многопоточности в Go. Практическое задание.

Тема 2. Go scheduler

Содержание: Горутины с точки зрения технической реализации. Связь горутин с потоками ОС и управление ими. Зачем использовать горутины вместо обычных потоков. Практическое задание.

Тема 3. Race condition
Содержание: Состояние гонки, последствия и методы предотвращения. Инструменты Go для детектирования гонок. Практическое задание.

Тема 4. Пакеты sync и atomic
Содержание: Примитивы синхронизации в Go. Управление потоками выполнения. Практическое задание.

Тема 5. Каналы ч.1. Deadlocks.
Содержание: Реализация каналов в Go. Буферизированные и небуферизированные каналы. Работа с закрытыми каналами. Понятие взаимной блокировки (deadlock). Практическое задание.

Тема 6. Каналы ч.2. Context.
Содержание: Использование конструкции select для работы с каналами. Объект context в Go. Практическое задание.

Тема 7. sync.Pool (опциональный)
Содержание: Переиспользование аллоцированной памяти в ходе выполнения программы. Практическое задание.

Тема 8. Пакет errgroup (опциональный)
Содержание: Использование errgroup для управления группами горутин и обработки ошибок. Практическое задание.

Модуль 7. Тестирование, бенчмарки и профилирование
Теория 4 академ. ч. Практика 8 академ. ч.
Модуль состоит из следующих тем:

Тема 1. Unit тестирование в Go
Содержание: Виды тестов. Написание тестов в Go. Инструменты тестирования Go. Практическое задание.

Тема 2. Моки, стабы и генерация через GoMock
Содержание: Подмена реализаций сторонних зависимостей (БД, сетевые клиенты и др.). Различные решения для подмены и их отличия. Практическое задание.

Тема 3. Table driven test vs closure driven tests
Содержание: Сравнение подходов Table driven и Closure func для написания тестов. Практическое задание.

Тема 4. Test coverage
Содержание: Отслеживание покрываемости кода тестами с помощью механизма test coverage. Практическое задание.

Тема 5. Benchmarks
Содержание: Встроенные инструменты для исследования и сравнения производительности блоков кода в Go. Эффективное сравнение результатов замеров. Практическое задание.

Тема 6. Профилирование с pprof
Содержание: Возможности встроенного механизма профилирования Go. Практическое задание.

Модуль 8. Кодогенерация
Теория 4 академ. ч. Практика 5 академ. ч.
Модуль состоит из следующих тем:

Тема 1. Рефлексия
Содержание: Понятие рефлексии. Разбор конструкций языка и работа с ними в рантайме. Практическое задание.

Тема 2. AST
Содержание: Анализ исходного кода программы на Go с помощью Abstract Syntax Tree. Возможности управления конструкциями кода для кодогенерации. Практическое задание.

Тема 3. Templates
Содержание: Инструменты шаблонизации в Go. Формат Go templates. Практическое задание.

Тема 4. Решение проблем рефлексии
Содержание: Сравнение универсальных преобразований в рантайме и статичного подхода с кодогенерацией. Сравнение производительности. Практическое задание.

Тема 5. Враппинг
Содержание: Использование wrapping для написания декораторов и автоматизации повторяющихся действий через кодогенерацию. Практическое задание.

Каждая тема модуля включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание. Практические задания рассчитаны на 2 академических часа. Выполнение заданий предполагает отправку решения на проверку через личный кабинет обучающегося. Критерии оценки прописаны в описании к каждому заданию. В случае корректного выполнения выставляется зачёт. Если работа содержит ошибки, задание возвращается на доработку. При повторной неудачной попытке (после двух доработок) обучающийся получает «незачёт».

Итоговая аттестация.
Модуль посвящён выполнению финального практического задания без предварительного теоретического блока.

5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Программа обеспечена системой дистанционного обучения https://rebrainme.com/.

Педагогические технологии:
- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

Дидактический материал:
Калеб Докси Введение в программирование на Go [Электронный ресурс]: https://golang-book.ru/

Электронно-библиотечные ресурсы и системы, информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы: https://my.rebrainme.com/course/golang-basic
3. Официальный сайт языка программирования Go [Электронный ресурс]: https://go.dev/
4. Документация языка Go [Электронный ресурс]: https://go.dev/doc/

6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценочные материалы:
Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль, промежуточный контроль и итоговое оценивание.

Текущий контроль
Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося в СДО. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическим решиме в СДО.

Модуль 2. Введение. Подготовка окружения.
Практическое задание

1. Установите Go и настройте рабочее пространство.
2. Ознакомьтесь со стилем написания кода на Go (code style) и утилитой для форматирования gofmt.
3. Создайте файл main.go и напишите код, который выводит приветствие и текущие дату и время в виде строки ДД.ММ.ГГГГ ЧЧ:ММ.

- Для вывода строки нужно воспользоваться стандартным пакетом fmt.
- Для получения даты и форматирования потребуется функция из стандартного пакета time.

Подробно концепция пакетов и работа с ними будут рассмотрены позже. На данном этапе следует ознакомиться с функциональностью указанных пакетов.

4. Скомпилируйте ваш код и запустите полученный бинарный файл. В качестве ответа пришлите исходный код программы.

Модуль 3. Основы языка Go для начинающих
Тема 8. Обработка ошибок

Практическое задание

1. Ознакомьтесь со стандартными пакетами log, errors.
2. Создайте в проекте module02 новую ветку 08_task.
3. Создайте новую директорию и файл main.go.
4. Скачайте файл in.txt и скопируйте его в директорию data рядом с файлом main.go.
5. Напишите программу, которая построчно считывает файл.
6. Выведите в консоль количество строк в формате Total strings: %d.
7. Корректно обработайте в отложенном вызове ошибки закрытия файловых дескрипторов.
8. Корректно обработайте ошибку окончания файла EOF.
9. Зафиксируйте изменения в ветке и отправьте их в удалённый репозиторий проекта.
10. В качестве ответа пришлите ссылку на merge request в ветку master вашего проекта ветки 08_task.

Модуль 4. Модули и пакеты
Тема 2. Работа с зависимостями, go mod
Практическое задание

1. Переведите проект из предыдущего задания на Go mod (если вы этого не сделали по ходу этого задания).
2. Уберите использование библиотеки github.com/fatih/color и приведите в соответствие файлы go.mod и go.sum с помощью команды go mod tidy.
3. Измените версию библиотеки github.com/huandu/xstrings на 1.2.1.
4. Соберите проект в режиме vendor, директорию vendor добавьте в .gitignore.
5. Выполненное задание поместите в ветку task_02 вашего репозитория.
6. В ответе пришлите ссылку на merge request в ветку master своего проекта ветки 02_task.

Модуль 5. Структуры и интерфейсы
Тема 5. Композитное наследование
Задание:

1. Создайте в своём проекте module04 из ветки module04_04 ветку module04_05.
2. Создайте структуру Overduer, в которую вынесите из структуры Customer поля balance и debt.
3. Сделайте так, чтобы Customer продолжал реализовывать интерфейс Debtor, но при этом поля balance и debt из него были недоступны напрямую.

Подсказка: помните, что можно привести структуру к интерфейсу, и встраивать в другую структуру уже интерфейс, чтобы скрыть детали реализации.

Логика WrOffDebt() (второе задание модуля) меняться не должна.

4. В ответе пришлите ссылку на MP ветки module04_05 с нужными правками в ветку master своего проекта.

Модуль 6. Асинхронность
Тема 3. Race condition
Практическое задание

1. Создайте в проекте module05 ветку module05_03.
2. Код для данного задания расположен в файле cache/main.go репозитория module05.

3. Перепишите структуру Cache так, чтобы она стала thread-safe (потокобезопасной). Потокобезопасность означает, что код работает исправно при использовании как одним, так и несколькими потоками.
4. В ответе пришлите ссылку на MP в ветку master своего проекта ветки module05_03.

Модуль 7. Тестирование, бенчмарки и профилирование
Тема 2. Моки, стабы и генерация через GoMock
Практическое задание
В этом задании вам предстоит написать тесты к функции PostCount. Эта функция через http-запрос достаёт посты пользователей какого-нибудь ресурса (представим, что этот ресурс — habr). API ресурса может работать, а может и не работать, поэтому тестировать функцию напрямую нельзя. Нужно замокать зависимости этой функции и протестировать только свой код.

1. В вашем проекте module06 сделайте новую ветку module06_02.
2. Проанализируйте функцию PostCount из пакета ./internal/app/processors/counter, какие зависимости есть и как они работают.
3. Сгенерируйте моки для клиента постов при помощи GoMock. Destination-каталог должен быть ./test/gomock/mocks/postmock.
4. Создайте файл post_counter_test.go в каталоге ./internal/app/processors/counter.
5. Напишите тесты с использование сгенерированных моков.
6. В качестве ответа пришлите ссылку на merge request в ветку master вашего проекта ветки module06_02, в которой должны быть:

- Моки, сгенерированные в папку ./test/gomock/mocks/postmock.
- Тест с использованием сгенерированных моков.

Модуль 8. Кодогенерация
Тема 3. Templates
Практическое задание
В этом задании вам нужно написать генератор YAML-конфигов. На вход ему передаётся шаблонизированный YAML-файл, который будет находиться в папке module07/assets/template/config_template.yml.

Вам нужно дополнить шаблон информацией с различными вставками и сгенерировать по этому шаблону валидный YAML-конфиг. Запуск осуществляется при помощи пакета module07/cmd/app и команды make run. Перед запуском нужно раскомментировать вызов функции Task03() в функции main.

Функция генератора имеет интерфейс:
func generate(tmpl string, outfilePath string, fields interface{}) error

Где:
- tmpl — это шаблон в виде строки;
- outFilePath — это путь к файлу, в который будет сгенерирован конфиг;
- fields — значения, которые нужны шаблону для подстановки.

Это основная функция генератора, через которую дальше вы будете генерировать различный код.

Также в пакете есть функция с интерфейсом вида.
func ConfigGenerate(tmpl string, outFilePath string) error

Эта функция должна подготавливать данные для шаблона и вызывать функцию generate.

Порядок действий:
1. В вашем проекте module07 создайте новую ветку module07_03.
2. В пакете module07/internal/generator заполните логикой функцию generate и ConfigGenerate.
3. Проверьте работоспособность генератора.
4. Проверьте через yaml validator сгенерированный вами конфиг.
5. В качестве ответа пришлите ссылку на merge request в ветку master вашего проекта ветки module07_03.

Итоговое оценивание
В конце программы обучающиеся сдают итоговую аттестацию.

Практическое задание
- Требуется реализовать приложение, которое в качестве параметров получает путь к двум директориям (исходной и её копии) и выполняет их синхронизацию.
- (Опционально) Ваша задача сделать не просто копирование файла, а полную синхронизацию. Чтобы при добавлении/удалении файла из одной папки, вторая полностью его повторяла. (Что-то похожее на BT SYNC), в том числе права на файлы, если это позволено разрешениями на уровне ОС.

Требования к реализации:

К Приложению:
- Работает в фоновом режиме.
- Использует асинхронность Go.
- Обязательно используется context.
- Обязательно используются инструменты пакета sync.
- Запрещены все библиотеки, связанные с асинхронностью, кроме нативных(sync, context).

К логам:
- Логи работы приложения должны записываться в файл log.txt.
- В логах обязательно должны присутствовать основные сведения (время операции, тип операции, размер файла и т.д).
- (Опционально) Продумать уровни логирования и предоставления более развёрнутой информации об операциях.
- (Опционально) Логи должны быть читаемыми и визуально легко воспринимаемыми.

К тестам:
- Должны присутствовать тесты на основные функции.
- Постарайтесь достичь максимального test-coverage.
- Также должны присутствовать benchmark на функции копирования.

К Git репозиторию:
- Репозиторий проекта должен находиться в приватном репозитории, расположенном в gitlab.rebrainme.com/golang_user_repos/<your_gitlab_id>

В качестве ответа на задание мы ожидаем от вас:
- Ссылку на репозиторий с вашим проектом.

- Несколько абзацев о том, с какими сложностями или нюансами вы столкнулись при выполнении. Что получилось реализовать из задуманного, а что нет.
- В README репозитория должна находиться короткая инструкция с описанием использования реализованной программы.

Результаты текущего контроля и итогового оценивания отображаются в личном кабинете слушателя в системе дистанционного обучения https://rebrainme.com/.

По результатам сдачи текущего контроля и итогового оценивания педагог даёт обратную связь слушателям, отмечает их сильные стороны и обращает внимание на зоны для развития. При необходимости педагог может повторить пройденные темы со слушателями, если установлен факт плохого закрепления и усвоения темы у слушателей.