

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))



Рабочая программа дисциплины (модуля),
как компонент образовательной программы
высшего образования - программы магистратуры
по направлению подготовки
09.04.01 Информатика и вычислительная техника,
утвержденной первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Микросервисная архитектура

Направление подготовки: 09.04.01 Информатика и вычислительная техника

Направленность (профиль): Технологии проектирования программного обеспечения

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде
электронного документа выгружена из единой
корпоративной информационной системы управления
университетом и соответствует оригиналу

Простая электронная подпись, выданная РУТ (МИИТ)
ID подписи: 5665
Подписал: заведующий кафедрой Нутович Вероника
Евгеньевна
Дата: 03.06.2023

1. Общие сведения о дисциплине (модуле).

Целями освоения данной дисциплины являются получение базовых, теоретических знаний о принципах, подходах разработки микросервисов на языке программирования Golang.

Основной задачей изучения дисциплины является формирование базовых представлений, знаний и умений о микросервисах, разработке как REST API, так и gRPC сервисов, работе с очередями сообщений и хранилищами данных, методами и средствами идентификации, авторизации и контейнеризации.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ОПК-5 - Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;

ОПК-6 - Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования;

ОПК-7 - Способен адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий;

ПК-2 - Способен проектировать и разрабатывать распределенные высокопроизводительные программные продукты с применением методов оптимизации программного обеспечения для корпоративного рынка.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- синтаксис языка программирования Golang ;
- основные структуры данных в языке программирования Golang;
- принципы объектно-ориентированного программирования на языке программирования Golang;
- принципы конкурентного программирования на языке программирования Golang.

Уметь:

- применять принципы и приемы конкурентного программирования при проектировании и разработке микросервисов;
- работать с брокерами сообщений на языке программирования Golang;

- применять практики проектирования и разработки слоя идентификации и авторизации;
- работать с Docker для создания и развертывания контейнеров.

Владеть:

- навыками разработки REST и gRPC интерфейсов;
- навыками работы с реляционными и нереляционными хранилищами на языке программирования Golang;
- навыками проектирования и разработки микросервисов на языке программирования Golang.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 6 з.е. (216 академических часа(ов)).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №2
Контактная работа при проведении учебных занятий (всего):	48	48
В том числе:		
Занятия лекционного типа	16	16
Занятия семинарского типа	32	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 168 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	<p>Введение в Golang.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- обзор языка и идиоматика программирования на Go;- области применения;- достоинства и недостатки;- стандартные конструкции языка;- типизация в Go;- особенности Go: слайсы, горутины, каналы, интерфейсы;- особенности синтаксиса: select, ошибки, объявления переменных;- пакеты и модули;- архитектурные принципы декомпозиции системы на пакеты.
2	<p>Структуры данных в Golang.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- встроенные структуры данных;- связные списки (List);- хэш-таблицы;- бинарные деревья поиска;- стеки и очереди.
3	<p>ООП в Golang.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- терминология;- методы;- инкапсуляция и наследование;- методы и интерфейсы;- методы и композиция типов;- конструкторы и деструкторы;- использование ООП в Go;- понятие и устройство интерфейса;- пустой интерфейс;- проверка типа интерфейса;- важные интерфейсы;- интерфейсы и архитектура системы;- обобщённое программирование (Generics).
4	<p>Конкурентное программирование.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none">- конкурентное программирование и параллелизм;- инструменты конкурентного программирования;- синхронизация потоков;- общая память и условие гонки;- Workers Pool;- типовые ошибки;- пакеты «sync» и «sync/atomic»;- шаблоны проектирования;- обработка ошибок и отмена.
5	<p>Сетевое программирование и веб-приложения.</p> <p>Рассматриваемые вопросы:</p>

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - программирование на транспортном уровне. Пакет «net»; - тестирование сетевой службы; - понятие веб-приложения; - HTTP-сервер; - маршрутизатор и обработчики запросов; - традиционные веб-приложения; - файловый сервер и SPA; - тестирование веб-приложения.
6	<p>Прикладной интерфейс разработки (API).</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - понятие API; - REST API; - Middleware; - Frameworks; - тестирование API; - пакет API; - контекст в HTTP-обработчиках; - авторизация; - WebSocket.
7	<p>Удалённый вызов процедур (RPC, gRPC).</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - основные сведения об RPC; - принцип работы; - реализация RPC в Go, .NET/RPC; - пример сервера и клиента RPC; - применение RPC; - тестирование RPC; - gRPC.
8	<p>SQL и NoSQL в Golang.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - database/sql и mysql; - GORM; - Memcached, Redis; - брокеры сообщений, RabbitMQ; - MongoDB.
9	<p>Микросервисы.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - понятие микросервисов и их отличительные особенности; - микросервисы и базы данных; - инфраструктура для микросервисов, варианты развёртывания; - управление конфигурацией; - сбор журналов; - мониторинг, профилирование и отладка; - фреймворки.
10	<p>Очереди сообщений.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - хеширование паролей; - CSRF-токены - на основе хеша, шифрования и JWT; - Сессии. Stateful и stateless сессии. Подводные камни сессий на основе JWT.
11	<p>Контейнеры.</p> <p>Рассматриваемые вопросы:</p>

№ п/п	Тематика лекционных занятий / краткое содержание
	- Docker и golang; - оптимизация сборки докер-контейнера, многошаговая сборка; - Docker-compose.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Основы Golang. В результате выполнения практической работы студент на типовых примерах знакомится с основами языка программирования Go.
2	ООП в Golang. В результате выполнения практической работы студент получает навык проектирования предметной области в приложениях на языке программирования Golang.
3	Конкурентность в Golang. В результате выполнения практической работы студент на типовых примерах знакомится с основными асинхронного и параллельного программирования в Go. В результате выполнения практической работы студент получает навык применения методов и принципов конкурентного программирования в Golang.
4	Веб-приложение и Golang. В результате выполнения практической работы студент получает навык разработки веб-приложения на языке программирования в Golang.
5	Сетевое программирование в Golang. В результате выполнения практической работы студент на типовых примерах знакомится с основами сетевого программирования в Go.
6	Прикладной интерфейс разработки (API). В результате выполнения практической работы студент на типовых примерах знакомится с основами разработки API на Go.
7	REST API и Golang. В результате выполнения практической работы студент получает навык разработки REST сервисов на языке программирования в Golang.
8	Удалённый вызов процедур. В результате выполнения практической работы студент на типовых примерах знакомится с основами разработки сервера и клиента RPC на Go. В результате выполнения практической работы студент получает навык разработки gRPC сервисов на языке программирования в Golang.
9	SQL и NoSQL в Golang. В результате выполнения практической работы студент на типовых примерах знакомится с основами работы с хранилищами в Go.
10	Микросервисы и Golang. В результате выполнения практической работы студент получает навык разработки микросервисов на языке программирования в Golang.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным и практическим занятиям.
3	Выполнение курсового проекта.
4	Подготовка к промежуточной аттестации.
5	Подготовка к текущему контролю.

4.4. Примерный перечень тем курсовых проектов

1. Проектирование и реализация микросервиса «Рекламный брокер».
2. Проектирование и реализация микросервиса «Статическое хранилище».
3. Проектирование и реализация микросервиса «Система рассылки».
4. Проектирование и реализация микросервиса «Auth0 провайдер».
5. Проектирование и реализация микросервиса «Фиды».
6. Проектирование и реализация микросервиса «Процессинг».
7. Проектирование и реализация микросервиса «Мессенджер».

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Облачный Go М. А. Титмус Москва : ДМК Пресс , 2021	https://e.lanbook.com/book/241106
2	Go на практике : руководство М. Батчер, М. Фарина Москва : ДМК Пресс , 2017	https://e.lanbook.com/book/97351
3	Docker на практике Э. Х. Сейерс, А. Милл Москва : ДМК Пресс , 2020	https://e.lanbook.com/book/131719
4	Микросервисы и контейнеры Docker : руководство П. С. Кочер Москва : ДМК Пресс , 2019	https://e.lanbook.com/book/123710
5	Программирование на Go. Разработка приложений XXI века : учебное пособие С. Марк Учебное пособие Москва : ДМК Пресс , 2013	https://e.lanbook.com/book/69944

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

<http://library.miit.ru/> – электронно-библиотечная система Научно-технической библиотеки МИИТ;

<http://e.lanbook.com/> – Электронно-библиотечная система издательства «Лань»;

<https://go.dev/doc/> – официальная документация Golang.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Microsoft Office

Golang

GoLand

Браузер с выходом в интернет

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Для лекционных занятий – наличие проектора и экрана.

Для лабораторных занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Курсовой проект во 2 семестре.

Экзамен во 2 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»

И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦГУТП

В.Е. Нутович

Председатель учебно-методической
комиссии

Н.А.Клычева