

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО
Заместитель директора

Ю.О. Соболев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Язык Go
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	2
квалификация:	магистр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 24 всего, в том числе:

лекции: 12 час.

семинары: 12 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 66 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: К.А. Лапин, старший методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и
онлайн-образования "Пуск" 01.03.2025

Аннотация

Изучение данной дисциплины направлено на формирование профессиональных компетенций бэкенд-разработчика. Студенты приобретут базовые знания и практические навыки, необходимые для создания высокопроизводительных и масштабируемых приложений с использованием языка Go. В ходе обучения осваивают базовый синтаксис, многопоточное программирование, разработку сетевых и распределенных приложений, а также тестирование и отладку кода. Полученные компетенции позволят студентам успешно приступить к работе в качестве квалифицированных специалистов в сфере бэкенд-разработки.

1. Цели и задачи

Цель дисциплины

- изучение основ языка программирования Go, понимание и применение принципов многопоточного и распределенного программирования, формирование навыков разработки высокопроизводительных и масштабируемых приложений с использованием языка Go.

Задачи дисциплины

- изучение базового синтаксиса и ключевых особенностей языка Go;
- освоение концепций многопоточности, в том числе горутин, синхронизации и асинхронных операций;
- изучение паттернов разработки и принципов проектирования приложений на Go;
- ознакомление с основными библиотеками и фреймворками для разработки веб-приложений и микросервисов на Go;
- практическое применение полученных знаний в разработке сетевых приложений, веб-сервисов и распределенных систем;
- изучение методов тестирования и отладки приложений на Go, включая использование инструментов для автоматизации тестирования.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-5 Способен и готов к профессиональному росту и руководству коллективом в области информатики и вычислительной техники, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере

ПК-6 Способен применять в профессиональной деятельности современные языки программирования и программное обеспечение; операционные системы и сетевые технологии, электронные библиотеки и пакеты программ	ПК-6.1 Знает современные языки программирования и методы параллельной обработки данных
	ПК-6.2 Умеет реализовывать и применять численные методы решения прикладных задач в профессиональной сфере деятельности, используя пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы синтаксиса и структуру языка программирования Go;
- принципы и особенности многопоточного программирования на Go;
- основные паттерны разработки и принципы проектирования приложений на Go;
- основные библиотеки и фреймворки для разработки веб-приложений и микросервисов на Go;
- методы тестирования и отладки приложений на Go.

уметь:

- разрабатывать многопоточные приложения с использованием горутин, синхронизации и асинхронных операций;
- применять паттерны разработки и принципы проектирования для создания модульных и поддерживаемых приложений на Go;
- использовать основные библиотеки и фреймворки для разработки веб-приложений и микросервисов на Go.

владеть:

- навыками создания программ на Go с использованием базового синтаксиса и структуры данных языка.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Введение в Go	2	2		10
2	Работа с основными конструкциями Go	2	2		12
3	Многопоточное программирование	2	2		10
4	Разработка сетевых приложений	2	2		10
5	Разработка веб-приложений	2	2		12
6	Разработка распределенных систем на Go	2	2		12
Итого часов		12	12		66
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 3 (Осенний)

1. Введение в Go

История создания языка Go. Основные принципы языка Go. Основные элементы языка Go: переменные, типы данных, операторы, выражения, структуры, функции и пакеты.

2. Работа с основными конструкциями Go

Основные конструкции языка: условные операторы, циклы, массивы, карты, срезы и каналы. Обработка исключений. Работа с файлами и потоками.

3. Многопоточное программирование

Основные понятия многопоточности: горутины, синхронизация, блокировки, очереди. Параллельное программирование: конкурентность, асинхронность, взаимодействие горутин.

4. Разработка сетевых приложений

Основные понятия сетевого программирования: сокет, протоколы, порты. Разработка клиент-серверных приложений. Сериализация и десериализация данных.

5. Разработка веб-приложений

Основные понятия веб-разработки: HTTP, HTML, CSS, JavaScript. Создание сервлетов и фильтров. Работа с базой данных через SQL.

6. Разработка распределенных систем на Go

Разработка распределенных приложений. Основные понятия распределенных систем: кластеры, балансировка нагрузки, отказоустойчивость. Работа с контейнерами и оркестраторами (Docker, Kubernetes). Разработка микросервисов.

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

Занятия по учебной дисциплине проводятся с применением дистанционных образовательных технологий. Каждый обучающийся обеспечен доступом к образовательной платформе (LMS).

6. Перечень рекомендуемой литературы

Основная литература

1. Параллельное программирование многопоточных систем с разделяемой памятью [Текст] : учеб. пособие для вузов / А. Г. Тормасов .— М : Физматкнига, 2014 .— 208 с.

Дополнительная литература

Цукалос, М. Golang для профи: работа с сетью, многопоточность, структуры данных и машинное обучение с Go: практическое руководство / М. Цукалос. - Санкт-Петербург: Питер, 2021. - 720 с. - (Серия «Для профессионалов»). - ISBN 978-5-4461-1617-1. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1733707> – Режим доступа: по подписке.

Аграновский, А. В. Многофункциональные информационные системы на основе интеграции прикладных программных сред: учебное пособие / А. В. Аграновский, Е. Л. Турнецкая. — Санкт-Петербург: ГУАП, 2022. — 91 с. — ISBN 978-5-8088-1732-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/340907> — Режим доступа: для авториз. пользователей.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Не используются

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Студенту для занятий потребуются:

1. Google Drive / Yandex disk для доступа к материалам курса
2. Zoom
3. LMS МФТИ
4. Приложение для коммуникации с преподавателями
5. Ноутбук для участия в интерактивных занятиях

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения прикладных задач.

Успешное освоение курса требует:

- посещения всех онлайн-занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение материалов на платформе дистанционного обучения и рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без вспомогательных материалов и конспектов отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

Возможен текущий контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	<u>2</u>
квалификация:	магистр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Разработчик: К.А. Лапин, старший методист

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
УК-6 Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1 Умеет решать задачи собственного личностного и профессионального развития, определять и реализовывать приоритеты совершенствования собственной деятельности
ОПК-5 Способен и готов к профессиональному росту и руководству коллективом в области информатики и вычислительной техники, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	ОПК-5.3 Стремится к получению новых знаний, профессиональному и личностному росту
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере
ПК-6 Способен применять в профессиональной деятельности современные языки программирования и программное обеспечение; операционные системы и сетевые технологии, электронные библиотеки и пакеты программ	ПК-6.1 Знает современные языки программирования и методы параллельной обработки данных
	ПК-6.2 Умеет реализовывать и применять численные методы решения прикладных задач в профессиональной сфере деятельности, используя пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые технологии

2. Показатели оценивания компетенций

В результате изучения дисциплины «Язык Go» обучающийся должен:

знать:

- основы синтаксиса и структуру языка программирования Go;
- принципы и особенности многопоточного программирования на Go;
- основные паттерны разработки и принципы проектирования приложений на Go;
- основные библиотеки и фреймворки для разработки веб-приложений и микросервисов на Go;
- методы тестирования и отладки приложений на Go.

уметь:

- разрабатывать многопоточные приложения с использованием горутин, синхронизации и асинхронных операций;
- применять паттерны разработки и принципы проектирования для создания модульных и поддерживаемых приложений на Go;
- использовать основные библиотеки и фреймворки для разработки веб-приложений и микросервисов на Go.

владеть:

- навыками создания программ на Go с использованием базового синтаксиса и структуры данных языка.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

В начале занятия проводится опрос по материалам предыдущего занятия и даются задачи для совместного решения.

Задание с самопроверкой

Цель: научиться применять принципы объектно-ориентированного программирования (ООП) на практике

Описание задания

Задача: написать программу, представляющую систему управления библиотекой.

Шаг 1. Создайте структуры Book и Library, где Book будет представлять информацию о книге (например, название, автор, год издания), а Library будет содержать список книг и методы для управления ими.

Шаг 2. Реализуйте методы AddBook, RemoveBook и DisplayBooks для структуры Library, предоставляя возможность добавлять, удалять и отображать информацию о книгах.

Шаг 3. Создайте интерфейс Readable с методом Read(), который будет представлять чтение книги.

Шаг 4. Реализуйте метод Read() для структуры Book, выводящий сообщение о том, что читатель начал читать эту книгу.

Шаг 5. Напишите функцию ReadBook, которая принимает объект типа Readable и вызывает его метод Read().

Шаг 6. В функции main() создайте объекты Library, добавьте несколько книг, отобразите список книг и затем вызовите ReadBook для одной из них.

Примечание.

Удостоверьтесь, что ваша программа следует принципам ООП, включая инкапсуляцию, наследование и полиморфизм в контексте Go.

Покажите, как ваша программа использует интерфейсы для представления операций чтения и как разделение интерфейсов поддерживает принципы SOLID.

Пример работы программы.

Программа выводит на экран список книг и информацию.

Пользователь имеет возможность добавлять, удалять и отображать информацию о книгах.

Книга выводится на экран для чтения.

Программа показывает, что пользователь начал читать эту книгу.

Критерии оценивания:

Созданы структуры Book и Library.

Программа осуществляет интерактивный ввод данных от пользователя с использованием методов AddBook, RemoveBook и DisplayBooks.

Создан интерфейс Readable, с методом Read(), который предоставляет книгу для чтения.

Реализован метод Read() для структуры Book, выводящий сообщение о том, что читатель начал читать эту книгу.

Создана функция ReadBook, которая принимает объект типа Readable и вызывает его метод Read().

В функции main() созданы объекты Library, добавлено несколько книг, список книг отображается, функция ReadBook выводит книгу на экран для чтения.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Промежуточный контроль проходит в форме дифференцированного зачета:

Типовые задания:

1. Выберите верное имя переменной согласно соглашениям по именованию переменных в Golang:

- a) myDog
- b) myData
- c) my_data
- d) MyDog

2. Какой код скомпилируется без ошибок?
- a) `name := "Petya"`
 - b) `name := String "Petya"`
 - c) `age := "18"`
 - d) `age := strconv.Atoi("18")`
3. Что означает `func main()` в Golang?
- a) Объявление пакета для основной программы.
 - b) Функция, содержащая код для выполнения.
 - c) Стартовая функция приложения, с которой начинается выполнение программы.
 - d) Импорт пакетов и зависимостей.
4. Какой способ позволяет обратиться к публичному полю структуры из другого пакета?
- a) Просто написать название поля, Golang автоматически найдет структуру с этим полем.
 - b) Импортировать пакет и обратиться к полю через имя пакета и точку (например, `package.Struct.field`).
 - c) Написать полный путь к полю: `пакет.структура.поле`.
 - d) Переместить структуру, из которой осуществляется доступ к полю, в тот же пакет и обратиться к полю через его имя и точку.
5. Как объявить переменную, которая будет иметь отдельное значение для каждого экземпляра структуры?
- a) Объявить переменную с модификатором `var`.
 - b) Объявить переменную без модификатора `var`.
 - c) Не объявлять переменную, а обращаться к ней через имя структуры и точку.
 - d) Невозможно объявить такую переменную, она всегда будет иметь общее значение для всех экземпляров структуры.
6. Какая конструкция языка Go используется для создания многопоточных приложений?
- a) `goroutine`
 - b) `thread`
 - c) `process`
 - d) `coroutine`
7. Как обратиться к элементу массива по индексу в Golang?
- a) `arr[arr.length]`
 - b) `arr[arr.length - 1]`
 - c) `arr[arr.length + 1]`
 - d) `arr[index]`
8. Какой код приведет к ошибке компиляции в Golang?
- a) `var arr [5]string; arr[0] = "Hello"`
 - b) `var arr [5]string; arr[0] = [5]string{}`
 - c) `var arr [5]string; arr[0] = 5`
 - d) `var arr [5]string; arr[0] = nil`
9. Какой оператор позволяет создать экземпляр структуры в Golang?
- a) `new`
 - b) `create`
 - c) `instanceOf`
 - d) `instance`
10. В чем состоит полиморфизм в Golang?
- a) Возможность одного объекта иметь несколько типов данных.
 - b) В наследовании одной структуры от другой.

- с) В переопределении методов в дочерней структуре.
- д) В использовании интерфейсов для полиморфного поведения.

Критерии оценивания

Дифференцированный зачет по дисциплине проводится в форме выполнения тестирования.

Оценка отлично (10 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение

уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5 баллов) - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2 балла) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1 балл) - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет проходит в письменном формате, на lms платформе.

Время проведения письменного дифференцированного зачета оставляет 2 академических часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.