

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

УТВЕРЖДЕНО
Директор

А.В. Малеев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Программирование на языке Go
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
курс:	2
квалификация:	бакалавр

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

Аудиторных часов: 45 всего, в том числе:

лекции: 0 час.

семинары: 45 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 27 час.

Всего часов: 72, всего зач. ед.: 2

Программу составил: А.В. Малеев, заведующий кафедрой

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 06.06.2022

Аннотация

Язык Go обладает множеством особенностей, таких как система типов без наследования, сокращенный синтаксис определения переменных и синтаксис анонимных функций и параллелизм. Цель дисциплины – обучить студентов практическим навыкам работы на языке программирования Go, а также использовать продвинутое методы для решения задач в области программирования.

1. Цели и задачи

Цель дисциплины

- обучить студентов практическим навыкам работы на языке программирования Go, а также использовать продвинутое методы для решения задач в области программирования.

Задачи дисциплины

- обучить студентов навыкам программирования на языке Go, познакомить студентов с его основными функциями и возможностями;
- познакомить студентов с передовыми практиками решения задач на языке программирования Go;
- обучить студентов использовать паттерны и архитектурные решения на языке программирования Go.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.4 Знает, как определять оптимальные методы и средства проектирования программного обеспечения и структур данных
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- продвинутое понятия разработки веб-приложений, работающих на многопроцессорных системах;
- базовый синтаксис языка;
- особенности работы с HTTP-1, HTTP-2.

уметь:

- пользоваться продвинутыми методами для решения задач программирования,

владеть:

- стандартным и продвинутым инструментарием языка Go.

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.		2		2
2	Базовые конструкции языка.		4		2
3	Методы.		4		2
4	Горутины и каналы.		4		2
5	Продвинутое тестирование.		4		2
6	Concurrency with shared memory.		4		2
7	Package context.		4		2
8	Работа с базами данных.		4		3
9	Reflection.		4		2
10	Ввод-вывод.		4		2
11	Архитектура GC.		3		2
12	Go tooling.		2		2
13	Полезные библиотеки.		2		2
Итого часов			45		27
Подготовка к экзамену		0 час.			
Общая трудоёмкость		72 час., 2 зач.ед.			

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

Семестр: 3 (Осенний)

1. Введение.

Введение. Философия дизайна. if, switch, for. Hello, world. Command line arguments. Word count. Animated gif. Fetching URL. Fetching URL concurrently. Web server. Tour of go. Local IDE setup. Submitting solutions to automated grading. gofmt. goimports. linting. Submitting PR's with bug fixes.

2. Базовые конструкции языка.

Базовые конструкции языка. names, declarations, variables, assignments. type declarations. packages and files. scope. Zero value. Выделение памяти. Стек vs куча. Basic data types. Constants. Composite data types. Arrays. Slices. Maps. Structs. JSON. text/template. string и []byte. Работа с unicode. Unicode replacement character. Функции. Функции с переменным числом аргументов. Анонимные функции. Ошибки.

3. Методы.

Методы. Value receiver vs pointer receiver. Embedding. Method value. Encapsulation. Интерфейсы. Интерфейсы как контракты. io.Writer, io.Reader и их реализации. sort.Interface. error. http.Handler. Интерфейсы как перечисления. Type assertion. Type switch. The bigger the interface, the weaker the abstraction. Обработка ошибок. panic, defer, recover. errors.{Unwrap,Is,As}. fmt.Errorf. %w.

4. Горутины и каналы.

Горутини и каналы. clock server. echo server. Размер канала. Блокирующее и неблокирующее чтение. select statement. Channel axioms. time.After. time.NewTicker. Pipeline pattern. Cancellation. Parallel loop. sync.WaitGroup. Обработка ошибок в параллельном коде. errgroup.Group. Concurrent web crawler. Concurrent directory traversal.

5. Продвинутое тестирование.

Продвинутое тестирование. Subtests. *testing.B. (*T).Logf. (*T).Skipf. (*T).FailNow. testing.Short(), testing flags. Генерация моков. testify/{require,assert}. testify/suite. Test fixture. Интеграционные тесты. Goroutine leak detector. TestingMain. Coverage. Сравнение бенчмарков.

6. Concurrency with shared memory.

Concurrency with shared memory. sync.Mutex. sync.RWMutex. sync.Cond. atomic. sync.Once. Race detector. Async cache. Работа с базой данных. database/sql. sqlx.

7. Package context.

Package context. Passing request-scoped data. http middleware. chi.Router. Request cancellation. Advanced concurrency patterns. Async cache. Graceful server shutdown. context.WithTimeout. Batching and cancellation.

8. Работа с базами данных.

Работа с базами данных. database/sql, sqlx, работа с базами данных, redis.

9. Reflection.

Reflection. reflect.Type and reflect.Value. struct tags. net/rpc. encoding/gob. sync.Map. reflect.DeepEqual.

10. Ввод-вывод.

Ввод-вывод. Пакет io, реализации Reader и Writer из стандартной библиотеки. Low-level programming. unsafe. Package binary. bytes.Buffer. cgo, syscall.

11. Архитектура GC.

Архитектура GC. Write barrier. Stack growth. GC pause. GOGC. sync.Pool. Шедюлер горутин. GOMACPROCS. Утечка тредов.

12. Go tooling.

Go tooling. pprof. CPU and Memory profiling. Кросс-компиляция. GOOS, GOARCH. CGO_ENABLED=0. Build tags. go modules. godoc. x/analysis. Code generation.

13. Полезные библиотеки.

Полезные библиотеки. CLI applications with cobra. Protobuf and GRPC. zap logging.

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

компьютер, мультимедийное оборудование (проектор)

6.Перечень рекомендуемой литературы

Основная литература

1. Чистый код : создание, анализ и рефакторинг / Р. Мартин. — Санкт-Петербург, Питер, 2019.— URL: <https://ibooks.ru/bookshelf/361844/reading> (дата обращения: 24.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
2. Технологии разработки программного обеспечения, учебник для вузов, стандарт третьего поколения / С. А. Орлов, Б. Я. Цилькер. — Санкт-Петербург, Питер, 2012.— URL: <https://ibooks.ru/bookshelf/28460/reading> (дата обращения: 25.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
3. Введение в программирование , учебное пособие / И. Ю. Баженова, В. А. Сухомлин. — Москва, ИНТУИТ, 2016.— URL: <https://e.lanbook.com/book/100695> (дата обращения: 30.12.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

-

Дополнительная литература

-

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

Не требуется

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

the GNU Compiler Collection последней версии

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

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой.

Самостоятельная работа включает в себя:

- проработку учебного материала;
- подготовку к практическим занятиям, выполнение домашних заданий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
курс:	2
квалификация:	бакалавр
Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет	
Разработчик:	А.В. Малеев, заведующий кафедрой

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.4 Знает, как определять оптимальные методы и средства проектирования программного обеспечения и структур данных
	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем

2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование на языке Go» обучающийся должен:

знать:

- продвинутое понятие разработки веб-приложений, работающих на многопроцессорных системах;
- базовый синтаксис языка;
- особенности работы с HTTP-1, HTTP-2.

уметь:

- пользоваться продвинутыми методами для решения задач программирования,

владеть:

- стандартным и продвинутым инструментарием языка Go.

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

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлого занятия.

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

- 1) Выделение памяти в языке Go
- 2) Работа с базами данных
- 3) Пакет io, реализации Reader и Writer из стандартной библиотеки
- 4) Обработка ошибок в параллельном коде
- 5) Функции с переменным числом аргументов
- 6) Анонимные функции в языке Go
- 7) Request cancellation
- 8) Работа со строками в языке Go
- 9) Напишите функцию, которая принимает на вход юникодную строку и должна возвращать строку, состоящую из тех же юникодных рун, но записанных в обратном порядке.
- 10) Напишите консольную утилиту, которая принимает на вход набор файлов и неотрицательное целое число k и печатает в stdout все уникальные слова, которые встречаются суммарно хотя бы k раз.

Критерии оценивания

- оценка «отлично (10)» 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 (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» 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 (6)» выставляется студенту, если он 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)» выставляется студенту, если он знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины

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

Дифференцированный зачет проводится по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.