

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

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики  
А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на Go
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Системное программирование и прикладная математика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: О.Н. Ивченко, старший преподаватель

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 02.04.2024

## Аннотация

Целью данного курса является получение навыков разработки программного обеспечения на языке Golang, а также на технологиях, популярных в среде разработчиков Golang. Будут рассмотрены основы программирования на Golang, средства разработки, сборки и тестирования программ на этом языке. Также в курсе будут затронуты темы шаблонов проектирования и работы с памятью. На практических занятиях будут отрабатываться навыки написания и тестирования программ на языке Golang, связывания программ с другими системами (например, с базами данных и веб-сервисами), развёртывания, сборки и публикации программ.

### 1. Цели и задачи

#### Цель дисциплины

- овладение студентами правил языка программирования Golang и современными инструментами разработки, являющимися стандартами среди Golang-разработчиков.

#### Задачи дисциплины

- приобретение студентами навыков проектирования и реализации приложений на языке Golang с использованием современных пакетов и библиотек, приемов асинхронности и веб-технологий;
- овладение студентами современных практик разработки: использование системы контроля версий, тестирования.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- Историю и область применения языка Golang;
- структуру программы на Golang: объявления, переменные, новые типы, пакеты и файлы, область видимости;
- типы данных языка Golang: числа, булевы, строки, константы, юникод;
- составные типы данных языка Golang: массивы, maps, structs, slices, динамические списки;
- функции, обработку исключений: panic, recover, defer;
- методы;
- интерфейсы;
- асинхронность: горутины и каналы;
- асинхронность: общедоступные переменные;
- пакеты и библиотеки Golang;
- тестирование;
- рефлексия;
- пакет unsafe;
- сборщик мусора в Golang;
- шаблоны проектирования.

уметь:

- Реализовывать библиотеку общего назначения на языке Golang по заданным интерфейсам;
- добавлять в приложение поддержку асинхронности, анализировать;
- покрывать код тестами, анализировать покрытие кода тестами;
- работать с распределенной системой контроля версий git;
- использовать средства code review на сервисе Github или GitLab.

владеть:

- Навыками работы с полученными знаниями в выборе архитектурного решения поставленной задачи.

#### 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		5
2	Структура программы	2	2		5
3	Типы данных	2	2		5
4	Составные типы	2	2		5
5	Методы	2	2		5
6	Интерфейсы	2	2		5
7	Асинхронность: горутины и каналы	2	2		5
8	Асинхронность: общедоступные переменные	2	2		5
9	Пакеты и библиотеки	2	2		5
10	Тестирование	2	2		5
11	Рефлексия	2	2		5
12	Пакет unsafe	2	2		5
13	Сборщик мусора	2	2		5
14	Шаблоны проектирования	2	2		5
15	Функции, обработка исключений	2	2		5
Итого часов		30	30		75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 4 (Весенний)

###### 1. Введение в Go

История языка Golang. Особенности языка Golang. Применение языка Golang.

###### 2. Структура программы

Объявления. Переменные. Новые типы. Пакеты и файлы. Область видимости.

###### 3. Типы данных

Числа. Булевые. Строки. Константы. Юникод.

#### 4. Составные типы

Массивы. Maps. Structs. Slices. Динамические списки.

#### 5. Методы

Основные методы разработки.

#### 6. Интерфейсы

Разработка интерфейсов и их виды.

#### 7. Асинхронность: горутины и каналы

Виды функций и каналов.

#### 8. Асинхронность: общедоступные переменные

Способы работы переменных.

#### 9. Пакеты и библиотеки

Основные виды библиотек и их разработка.

#### 10. Тестирование

Возможности тестирования при разработке на Go.

#### 11. Рефлексия

Изучение и модифицирование структуры и поведения.

#### 12. Пакет unsafe

Обход ограничений и ускорения в Go.

#### 13. Сборщик мусора

Удаление неиспользуемых объектов.

#### 14. Шаблоны проектирования

Виды паттернов проектирования.

#### 15. Функции, обработка исключений

Panic. Recover. Defer.

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

Компьютеры, либо ноутбуки с доступом в интернет, камерой и микрофоном.

## **6.Перечень рекомендуемой литературы**

### Основная литература

1. 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 используются

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

GoLand: IDE для Go с расширенной поддержкой JavaScript, TypeScript и баз данных

## **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 (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение индивидуальных домашних заданий и итогового проекта.

Промежуточный контроль знаний проводится в виде индивидуальных домашних работ в формате проектной работы.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Системное программирование и прикладная математика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 4 (весенний) - Дифференцированный зачет

**Разработчик:** О.Н. Ивченко, старший преподаватель

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности

## 2. Показатели оценивания компетенций

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

### знать:

- Историю и область применения языка Golang;
- структуру программы на Golang: объявления, переменные, новые типы, пакеты и файлы, область видимости;
- типы данных языка Golang: числа, булевы, строки, константы, юникод;
- составные типы данных языка Golang: массивы, maps, structs, slices, динамические списки;
- функции, обработку исключений: panic, recover, defer;
- методы;
- интерфейсы;
- асинхронность: горутины и каналы;
- асинхронность: общедоступные переменные;
- пакеты и библиотеки Golang;
- тестирование;
- рефлексия;
- пакет unsafe;
- сборщик мусора в Golang;
- шаблоны проектирования.

### уметь:

- Реализовывать библиотеку общего назначения на языке Golang по заданным интерфейсам;
- добавлять в приложение поддержку асинхронности, анализировать;
- покрывать код тестами, анализировать покрытие кода тестами;
- работать с распределенной системой контроля версий git;
- использовать средства code review на сервисе Github или GitLab.

### владеть:

- Навыками работы с полученными знаниями в выборе архитектурного решения поставленной задачи.

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

Пример типовых (примерных) заданий текущего контроля

1. Написать программу, которая вычисляет сумму всех элементов в массиве.
2. Реализовать функцию, которая принимает на вход число и возвращает true, если число является простым, и false в противном случае.
3. Написать программу, которая считывает текстовый файл и подсчитывает количество слов в нем.
4. Создать структуру данных для представления информации о студенте (имя, возраст, оценки) и реализовать методы для работы с этой структурой.
5. Написать программу, которая генерирует случайный пароль заданной длины.

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

1. Каков идиоматический способ приостановить выполнение текущей области до тех пор, пока не будет возвращено произвольное количество подпрограмм?
2. Каков побочный эффект использования `time.After` в утверждении `select`?
3. Какие ограничения есть на тип `var` для компиляции `i := myVal.(int)`?
4. Как правильно передать это как тело запроса HTTP POST?
5. Каким должно быть идиоматическое имя для интерфейса с одним методом и сигнатурой `Save() error`?
6. Какова чувствительность к регистру по умолчанию для функции `JSON.Unmarshal`?
7. Где полезен встроенный метод `recover`?
8. В чём разница между `time` пакетами `Time.Sub()` и методами `Time.Add()`?
9. Каков риск использования нескольких тегов полей в одной структуре?
10. Если вы перебираете `map` в цикле `for range`, в каком порядке будет осуществляться доступ к парам `ключ:значение`?

#### Критерии оценивания

Оценка «отлично (10)» выставляется обучающемуся, если он показал 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 обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений, но при этом были допущены небольшие неточности, которые были самостоятельно обнаружены и исправлены;

оценка «отлично (8)» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений, но при этом были допущены небольшие неточности, которые после указания экзаменатора были самостоятельно исправлены;

оценка «хорошо (7)» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает неточности в ответе или делает несущественные ошибки при решении задач;

оценка «хорошо (6)» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает небольшие ошибки в ответе и (или) при решении задач;

оценка «хорошо (5)» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но отвечает неуверенно и (или) допускает ошибки при решении задач;

оценка «удовлетворительно (4)» выставляется обучающемуся, показавшему фрагментарный, разрозненный характер знаний, неточные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, если 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 обучающемуся, показавшему фрагментарный, разрозненный характер знаний, неточные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, не владеющему некоторыми разделами учебной программы, но умеющему применять полученные знания по образцу в стандартной ситуации;

оценка «неудовлетворительно (2)» выставляется обучающемуся, 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 обучающемуся, показавшему полное незнание учебной программы дисциплины.



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

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.