

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

УТВЕРЖДАЮ

декан факультета вычислительной математики и кибернетики



/И.А. Соколов /

« 14 » октябрь 2021г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины (модуля):

Программирование на языке Golang

Уровень высшего образования:

магистратура

Направление подготовки / специальность:

01.04.02 "Прикладная математика и информатика" (3++)

Направленность (профиль) ОПОП:

Искусственный интеллект в кибербезопасности

Форма обучения:

очная

Рабочая программа рассмотрена и утверждена
на заседании Ученого совета факультета ВМК
(протокол № 7, от 29 сентября 2021 года)

Москва 2021

Рабочая программа дисциплины (модуля) разработана в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) для реализуемых основных профессиональных образовательных программ высшего образования по направлению подготовки 01.04.02 "Прикладная математика и информатика" утвержденного Приказом Министерства образования и науки РФ от 10 января 2018 г. N 13.

1. Место дисциплины (модуля) в структуре ОПОП ВО:

Дисциплина (модуль) относится к части дисциплин основной профессиональной образовательной программы, формируемых участниками образовательных отношений.

2. Входные требования для освоения дисциплины (модуля), предварительные условия:

Учащиеся должны владеть знаниями по алгоритмам и алгоритмическим языкам, технологиям сети Интернет в объеме, соответствующем основным образовательным программам бакалавриата по укрупненным группам направлений и специальностей 02.00.00 «Компьютерные и информационные науки».

3. Результаты обучения по дисциплине (модулю), соотнесенные с требуемыми компетенциями выпускников.

Планируемые результаты обучения по дисциплине (модулю)		
Содержание и код компетенции.	Индикатор (показатель) достижения компетенции	Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций
ПК-8. Способен разрабатывать и модернизировать программное и аппаратное обеспечение технологий и систем искусственного интеллекта с учетом требований информационной безопасности в различных предметных областях	ПК-8.1. Разрабатывает программное и аппаратное обеспечение технологий и систем искусственного интеллекта для решения профессиональных задач с учетом требований информационной безопасности в различных предметных областях	ПК-8.1. З-1. Знает новые научные принципы и методы разработки программного и аппаратного обеспечения технологий и систем искусственного интеллекта для решения профессиональных задач в различных предметных областях ПК-8.1. У-1. Умеет разрабатывать программное и аппаратное обеспечение технологий и систем искусственного интеллекта с учетом требований информационной безопасности для решения профессиональных задач в различных предметных областях

4. Объем дисциплины (модуля) составляет 3 з.е., в том числе 72 академических часа, отведенных на контактную работу обучающихся с преподавателем, 36 академических часов на самостоятельную работу обучающихся.

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

Целью курса является обучение слушателей эффективному использованию языка Go, для достижения чего необходимо решить четыре задачи:

1. по возможности всесторонне ознакомить слушателей с языком программирования;
2. предоставить возможность наработки навыков программирования на языке во время решения домашних заданий;
3. представить эффективные приёмы программирования на Go на примере некоторых инструментально-прикладных компонент;
4. осветить современные тенденции программирования, нашедшие отражение в структуре языка, в том числе мультипарадигмальный подход при разработке программ.

Наименование и краткое содержание разделов и тем дисциплины, форма промежуточной аттестации по дисциплине	Всего (часы)	В том числе								
		Контактная работа (работа во взаимодействии с преподавателем), часы					Самостоятельная работа учащегося, часы			
		из них					из них			
Занятия лекционного типа	Практические занятия	Групповые консультации	Индивидуальные консультации	Учебные занятия, направленные на проведение текущего контроля успеваемости: коллоквиумы, практические контрольные занятия и др.	Всего	Выполнение домашних заданий	Подготовка рефератов и т.п..	Всего		
Тема 1. Введение в Go	6	3	3	-	-	-	6	-	-	-

<ul style="list-style-type: none"> - Версии Go; Примеры использования Go; - Популярность Go. 										
Тема 2. Лексическая структура <ul style="list-style-type: none"> - Кодировка и case sensitivity; - Пробелы; - Комментарии; - Идентификаторы; - Резервированные слова. 	9	3	3	-	-	-	6	3	-	3
Тема 3. Типы данных <ul style="list-style-type: none"> - Числа; - Строки; - Логический тип; - Массив - Срез (slice) - Map - Конвертация типов данных; - Проверка типов. 	11	4	4	-	-	-	8	3	-	3
Тема 4. Переменные и константы <ul style="list-style-type: none"> - Типизация и декларация; - Нулевые значения - Примитивные и ссылочные типы; - Сборщик мусора; - Область видимости. 	9	3	3	-	-	-	6	3	-	3
Тема 5. Операторы	9	3	3	-	-	-	6	3		3

<ul style="list-style-type: none"> - Обзор операторов; - Приоритеты операций; - Арифметические операторы; - Условные операторы; - Логические операторы; - Битовые операторы; - Оператор присваивания; - Прочие операторы. 										
Тема 6. Утверждения <ul style="list-style-type: none"> - Утверждения return и goto; - Встроенная функция panic; - Утверждение block; - Утверждение if; - Утверждение for; - Утверждение while; - Утверждение switch; - Утверждение select; - Утверждение continue; - Утверждение defer. 	9	3	3	-	-	-	6	3	-	3
Тема 7. Функции <ul style="list-style-type: none"> - Определение и вызов функции; - Аргументы функции; - Функции как данные; - Функции как методы. 	9	3	3	-	-	-	6	3	-	3
Тема 8. Методы и интерфейсы <ul style="list-style-type: none"> - Методы и функции; - Получатели и указатели; 	9	3	3	-	-	-	6	3	-	3

<ul style="list-style-type: none"> - Методы и косвенная адресация указателей; - Интерфейсы; - Пустой интерфейс; - Утверждение типа (type assertion); - Switch с типами; - Стандартные интерфейсы: Stringers, Reader. 										
<p>Тема 9. Асинхронное программирование</p> <ul style="list-style-type: none"> - Go-процедуры (goroutines); - Каналы, буферизированные каналы; - Range и Close; - Select; - Пакет sync. sync.Mutex, sync.RWMutex, sync.Map. 	9	3	3	-	-	-	6	3	-	3
<p>Тема 10. Практические аспекты</p> <ul style="list-style-type: none"> - Пакеты и повторное использование кода; - Модули; - Документирование кода; - Обработка ошибок; - Тестирование, Code coverage, Benchmark; - Оптимизация, Профилирование; - Linters, go vet; - Docker. 	11	4	4	-	-	-	8	3	-	3

Тема 11. Разработка сетевых приложений - Пакеты net, net/http; - JSON, аннотации полей структур; - gRPC.	13	4	4	-	-	-	8	5	-	5	
Промежуточная аттестация экзамен	4	-	-	-	-	4	4	-	-	-	
Итого	108						72	36			

6. Фонд оценочных средств (ФОС, оценочные и методические материалы) для оценивания результатов обучения по дисциплине (модулю).

6.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости, критерии и шкалы оценивания
Примеры домашних заданий.

Домашнее задание №1

Задача

№1

”Список”

Необходимо создать динамическую структуру данных односвязный (кто хочет, может сделать двух-связанный) список. Он должен поддерживать следующие операции: добавление нового узла в список, поиск по элементам списка, удаление элемента списка. Поле полезной нагрузки должно хранить число. Также необходимо создать функцию, которое будет принимать на вход число любой длины и представлять его в виде списка. Стандартные структуры типа list etc использовать нельзя.

Задача №2 ”Длинная арифметика”

В качестве входящих данных у вас есть два связанных списка представляющих два неотрицательных числа. Цифры хранятся в обратном порядке и каждый элемент списка хранит одну цифру. Сложите два числа и верните результат в виде связанного списка. Предполагается что оба числа не содержат лидирующих нулей, кроме случая когда число само по себе 0. Числа могут быть сколь угодно большие.

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)

Output: 7 -> 0 -> 8

Домашнее задание №2

Задача №1 "Строка — повторение подстроки"

Ввести непустую строку s. Найти такое наибольшее число k и такую строку t, что s совпадает со строкой t, выписанной k раз подряд. Вывести k.

Input: abcabcabcabc

Output: 4

Задача №2 "Самое популярное слово"

Ввести построчно текст, состоящий из пробелов, переводов строки и латинских букв, и заканчивающийся пустой строкой. Вывести слово, которое чаще других встречается в тексте, если оно такое одно, и —, если таких слов несколько.

Input: Sed tempus ipsum quis eros tempus lacinia Cras finibus lorem ut lacinia egestas nunc nibh iaculis est convallis tincidunt mi mi sed nisl Sed porttitor aliquam elit ullamcorper tincidunt arcu euismod quis Mauris congue elit suscipit leo varius facilisis Cras et arcu sodales laoreet est vitae pharetra orci Integer eget nulla dictum aliquet justo semper molestie neque Maecenas bibendum lacus tincidunt auctor varius purus felis ullamcorper dui et laoreet ligula ex et risus Donec eget fringilla nibh Cras congue tincidunt accumsan Maecenas euismod eleifend elit ut rhoncus tortor sodales a Cras egestas finibus lorem non tempor tincidunt aera

Output: tincidunt

Задача №3 "Буквенные комбинации номера телефона"

На вход алгоритму подается строка, состоящая из цифр от 0 до 9 (номер телефона), нужно вернуть все возможные комбинации букв, которые могут быть поставлены в номере.

Input: "23"

Output: ["ad "ae "af "bd "be "bf "cd "ce "cf"]

Задача №4 "Граф"

Реализовать структуру данных граф и два способа обхода графа:

- в глубину (DFS)
- в ширину (BFS)

На вход будет подан массив ребер графа [[0, 3], [1, 3], [2, 3], [4, 3], [5, 4]] В результате должна быть структура, принимающая массив ребер и имеющая две функции, при вызове которых будет выполнять обход вершин. Посещаемые вершины должны выводиться на экран.

Задача №5 "Поиск пути"

Пусть дан граф, причем каждому его ребру сопоставлен вес (взвешенный граф). Требуется найти путь между двумя заданными вершинами с наименьшим весом. На вход будет подан массив ребер графа [[0, 3, 5], [1, 3, 11], [2, 3, 56], [4, 3, 77], [5, 4, 89]] Третий параметр - вес ребра.

У структуры данных граф должна быть функция, принимающая два параметра (nodeStart, nodeEnd). Данная функция должна вывести кратчайший путь из nodeStart в nodeEnd.

Задача №6 "Время задержки сети"

У нас есть сеть состоящая из узлов, каждый из которых помечен меткой от 1 до

N . Дан список времени распространения сигнала от одного узла до другого в виде направленных ребер. $times[i] = (u, v, w)$, где u - начальный узел, v - конечный узел, и w - время распространения сигнала от начального узла к конечному. Мы отправляем сигнал из узла X и хотим узнать как долго он будет распространяться сигнал до всех узлов. Если это не возможно, то возвращаем -1.

Input: $times = [[2,1,1],[2,3,1],[3,4,1]]$, $N = 4$, $X = 2$

Output: 2

Дополнительные условия:

1. $N \in [1, 100]$
2. $X \in [1, N]$
3. $times \in [1, 6000]$
4. Все ребра $times[i] = (u, v, w)$ будут иметь $1 \leq u, v \leq N$ и $0 \leq w \leq 100$

Домашнее задание №3

Задача №1 "Разработка протокола прикладного уровня"

Необходимо разработать протокол прикладного уровня, который будет соответствовать следующей спецификации:

Сообщение клиент-сервер:

- Сначала передается командная строка состоящая из [Команда] [Размер поля полезной нагрузки]
- Команды - STAT и ENTI, команда STAT возвращает статистические данные, собранные в ходе анализа переданного текста, а команда ENTI возвращает найденные именованные сущности
- В качестве полезной нагрузки будет передаваться список твитов
- Нужно посчитать 10 наиболее часто встречающихся слов, 10 наиболее популярных твитов, их авторов и сколько раз они были ретвитнуты, 10 самых популярных авторов.

- Информация о странах, в которых пользователи создают контент (твиты) и в которых его потребляют (ретвитят) – Датасет - https://github.com/Ilyushin/JavaScriptCourse/blob/master/Tasks/tweets_analysis/input/dataSet.csv
 - Для распознавания именованных сущностей предлагается использовать Stanford CoreNLP Natural Language Processing Using Stanford's CoreNLP <https://towardsdatascience.com/natural-language-processing-using-stanfords-corenlp-d9e64c1e1024>
 - Сервер должен поддерживать несколько одновременных подключений
 - После передаются данные, размер которых указан в командной строке
 - Данные могут передавать в любом удобном формате - csv, json, xml etc
- Сообщение сервер-клиент: ·
- Ответ может передаваться в любом удобном формате - csv, json, xml etc
 - Первой строкой передается размер данных ответа

Задача №2 "Разработка веб сервера"

Требуется разработать веб-сервер, удовлетворяющий следующим минимальным требованиям:

- Сервер должен поддерживать два endpoint.
- Первый возвращает статистические данные, собранные в ходе анализа переданного текста, а второй возвращает найденные именованные сущности
- Основной протокол передачи данных http
- В качестве полезной нагрузки будет передаваться список твитов
- Нужно посчитать 10 наиболее часто встречающихся слов, 10 наиболее популярных твитов, их авторов и сколько раз они были ретвитнуты, 10 самых популярных авторов.
- Информация о странах, в которых пользователи создают контент (твиты) и в которых его потребляют (ретвитят) – Датасет - https://github.com/Ilyushin/JavaScriptCourse/blob/master/Tasks/tweets_analysis/input/dataSet.csv
- Для распознавания именованных сущностей предлагается использовать Stanford CoreNLP Natural Language Processing Using Stanford's CoreNLP <https://towardsdatascience.com/natural-language-processing-using-stanfords-corenlp-d9e64c1e1024>
- Сервер должен поддерживать несколько одновременных подключений

Особенности организации процесса обучения

Для эффективного освоения курса рекомендуется перед каждым занятием привести в порядок конспекты лекций. После каждого занятия рекомендуется найти и прочитать дополнительную литературу по теме лекции и прочитать свои конспекты.

Система контроля и оценивания

За каждую домашнюю выставляются баллы (максимум 40 баллов). Пусть M – максимальное число баллов, которое может набрать студент. В конце семестра баллы конвертируются в оценку $O1$ следующим образом:

меньше $M/2$ баллов: $O1=2$;

больше или равно $M/2$ баллов, но меньше $2M/3$: $O1=3$;

больше или равно $2M/3$ баллов, но меньше $5M/6$: $O1=4$;

больше или равно $5M/6$ баллов: $O1=5$.

На экзамене оценка $O1$ является стартовой. Окончательная оценка определяется исходя из оценки устного ответа студента, при этом она не может отличаться от стартовой оценки более чем на 1 балл.

Структура и график контрольных мероприятий

Устная сдача домашних заданий в конце каждой недели, устный экзамен в конце семестра.

6.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации по дисциплине, критерии и шкалы оценивания

Список вопросов для экзамена.

- 1) Система типов в ЯП Go.
- 2) Особенности обработки исключений.
- 3) Алгоритм сборки мусора в Go.
- 4) Протокол межсетевого взаимодействия gRPC.
- 5) Особенности объявления переменных.
- 6) Асинхронное программирование в Go.
- 7) Методы и адресация указателей.
- 8) Взаимодействие Go и C.
- 9) Особенности компилятора Go.
- 10) Функции первого класса и высшего порядка.

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)				
Оценка	2 (не зачтено)	3 (зачтено)	4 (зачтено)	5 (зачтено)
РО и соответствующие виды оценочных средств				
Знания <i>Экзамен</i>	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения <i>Практические задания</i>	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности непринципиального характера)	Успешное и систематическое умение
Навыки (владения, опыт деятельности) <i>Экзамен, практические занятия</i>	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач

7. Ресурсное обеспечение:

7.1. Перечень основной и дополнительной литературы

Основная литература

- 1) Донован, Алан А. А., Керниган, Брайан, У. Язык программирования Go. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2016. — 432 с.
- 2) Цукалос Михалис. Golang для профи: работа с сетью, многопоточность, структуры данных и машинное обучение с Go. — СПб.: Питер, 2020. — 720 с.

Дополнительная литература

- 1) Cormen T., Charles L., Ronald R., Clifford S. «Introduction to Algorithms», 3-е издание, Massachusetts Institute of Technology, 2009.
- 2) Орлов С.А. «Теория и практика языков программирования», СПб.: Питер, 2013.
- 3) The Go Programming Language Specification <https://golang.org/ref/spec>
- 4) "Mastering Go - Second Edition" <https://www.packtpub.com/programming/mastering-go-second-edition>

7.2. Перечень лицензионного программного обеспечения, в том числе отечественного производства
При реализации дисциплины может быть использовано следующее программное обеспечение:

1. Программное обеспечение для подготовки слайдов лекций MS PowerPoint
2. Программное обеспечение для создания и просмотра pdf-документов Adobe Reader
3. Издательская система LaTeX.

7.3. Перечень профессиональных баз данных и информационных справочных систем

1. <http://www.edu.ru> – портал Министерства образования и науки РФ
2. <http://www.ict.edu.ru> – система федеральных образовательных порталов «ИКТ в образовании»
3. <http://www.openet.ru> - Российский портал открытого образования
4. <http://www.mon.gov.ru> - Министерство образования и науки Российской Федерации
5. <http://www.fasi.gov.ru> - Федеральное агентство по науке и инновациям

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Math-Net.Ru [Электронный ресурс] : общероссийский математический портал / Математический институт им. В. А. Стеклова РАН ; Российская академия наук, Отделение математических наук. - М. : [б. и.], 2010. - Загл. с титул. экрана. - Б. ц. URL: <http://www.mathnet.ru>
2. Университетская библиотека Online [Электронный ресурс] : электронная библиотечная система / ООО "Директ-Медиа" . - М. : [б. и.], 2001. - Загл. с титул. экрана. - Б. ц. URL: www.biblioclub.ru

3. Универсальные базы данных East View [Электронный ресурс] : информационный ресурс / East View Information Services. - М. : [б. и.], 2012. - Загл. с титул. экрана. - Б. ц.

URL: www.ebiblioteka.ru

4. Научная электронная библиотека eLIBRARY.RU [Электронный ресурс] : информационный портал / ООО "РУНЭБ" ; Санкт-Петербургский государственный университет. - М. : [б. и.], 2005. - Загл. с титул. экрана. - Б. ц.

URL: www.eLibrary.ru

7.5. Описание материально-технического обеспечения.

Факультет ВМК, ответственный за реализацию данной Программы, располагает соответствующей материально-технической базой, включая современную вычислительную технику, объединенную в локальную вычислительную сеть, имеющую выход в Интернет. Используются специализированные компьютерные классы, оснащенные современным оборудованием. Материальная база факультета соответствует действующим санитарно-техническим нормам и обеспечивает проведение всех видов занятий (лабораторной, практической, дисциплинарной и междисциплинарной подготовки) и научно-исследовательской работы обучающихся, предусмотренных учебным планом.

8. Соответствие результатов обучения по данному элементу ОПОП результатам освоения ОПОП указано в Общей характеристике ОПОП.

9. Разработчик (разработчики) программы.

Ассистент, Ильюшин Евгений Альбинович
доцент, к.ф.-м.н., Чижов Иван Владимирович

10. Язык преподавания - русский.