

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 /
«27» сентября 2023г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине
Программирование на языке Go

Уровень высшего образования:
бакалавриат

Направление подготовки / специальность:
02.03.02 "Фундаментальная информатика и информационные технологии" (3++)

Направленность (профиль) ОПОП:
Искусственный интеллект и анализ данных

Форма обучения:
очная

Рассмотрен и утвержден
на заседании Ученого совета факультета ВМК
(протокол №7, от 27 сентября 2023 года)

Москва 2023

1. ФОРМЫ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

В процессе и по завершении изучения дисциплины оценивается формирование у студентов следующих компетенций:

Планируемые результаты обучения по дисциплине (модулю)		
Содержание и код компетенции.	Индикатор (показатель) достижения компетенции	Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций
ОПК-5. Способен устанавливать и сопровождать программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учетом информационной безопасности	ОПК-5.1. Устанавливает программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учетом информационной безопасности ОПК-5.2. Сопровождает программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учетом информационной безопасности	Знать синтаксис языка программирования Golang; основные принципы объектно-ориентированного программирования. основные классы из библиотеки классов языка программирования Golang для создания объектно-ориентированных приложений. Уметь разрабатывать программы на языке программирования Golang, создавая собственные классы, а также использовать классы и модули из библиотек этого языка; создавать удобный интерфейс для использования созданных программных средств с помощью графических библиотек. Владеть навыками разработки консольных приложений в стиле объектно-ориентированного программирования на языке программирования Golang; навыками разработки приложений с внешними источниками данных (текстовыми файлами, xml-файлами, базами данных)

1.1. Текущий контроль успеваемости

Текущий контроль успеваемости осуществляется путем оценки результатов выполнения заданий практических (семинарских) занятий, самостоятельной работы, предусмотренных учебным планом и посещения занятий/активность на занятиях.

В качестве оценочных средств текущего контроля успеваемости предусмотрены:

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 представлены 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

1.2. Промежуточная аттестация

Промежуточная аттестация осуществляется в форме экзамена

В качестве средств, используемых на промежуточной аттестации предусматривается:

Билеты

1.3. Типовые задания для проведения промежуточной аттестации

Вопросы к экзамену

- 1) Система типов в ЯП Go.
- 2) Особенности обработки исключений.
- 3) Алгоритм сборки мусора в Go.
- 4) Протокол межсетевое 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.

2. КРИТЕРИИ ОЦЕНКИ ПО ДИСЦИПЛИНЕ

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине				
Оценка	2 (не зачтено)	3 (зачтено)	4 (зачтено)	5 (зачтено)
виды оценочных средств				
Знания (виды оценочных средств: приведены в п. 1.2.)	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения (виды оценочных средств: приведены в п. 1.2.)	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
Навыки (владения, опыт деятельности) (виды оценочных средств: приведены в п. 1.2..)	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