ЛР2 Java Collections Framework
Часть 1 – общая

Все коллекции, которые необходимо реализовать должны быть обобщёнными.

Двусвязный список

Реализуйте собственный двусвязный список на языке Java. Ваш класс должен поддерживать следующий функционал:

- void add(T element): добавляет элемент в конец списка.
- void add(int index, T element): вставляет элемент по указанному индексу.
- T get(int index): возвращает элемент по индексу.
- T remove(int index): удаляет элемент по индексу и возвращает его.
- int size(): возвращает количество элементов в списке.
- boolean isEmpty(): проверяет, пуст ли список.

Поскольку список является двусвязным, то его необходимо реализовывать через “узлы”, которые имеют связи со следующим и предыдущим узлом.

Реализуйте итератор для вашего списка, чтобы можно было перебрать его элементы с помощью цикла for-each. Итератор должен реализовывать интерфейс Iterator<T> и поддерживать методы:

- boolean hasNext()
- T next()

При выходе за пределы списка выбрасывайте соответствующие исключения, например, IndexOutOfBoundsException.

Для демонстрации работы необходимо реализовать собственный класс с функцией main, в котором необходимо показать использование всех вышеперечисленных методов.

Хэш таблица

Реализуйте собственную хеш-таблицу на языке Java. Ваш класс должен поддерживать следующий функционал:

- V put(K key, V value): добавляет пару ключ-значение в таблицу. Если такой ключ уже существует, заменяет его значение новым и возвращает старое значение.
- V get(K key): возвращает значение, связанное с ключом.
- V remove(K key): удаляет пару по ключу и возвращает связанное значение
- boolean containsKey(K key): проверяет, присутствует ли ключ в таблице.
- int size(): возвращает количество пар в таблице.
- boolean isEmpty(): проверяет, пуста ли таблица.
- void clear(): очищает таблицу.

Если значение отсутствует при выполнении каких либо операций (получение, добавление, удаление) можно возвращать Null, либо исключение.

Реализуйте итератор для вашей хэш-таблицы, чтобы можно было перебрать его элементы с помощью цикла for-each. Итератор должен реализовывать интерфейс Iterator<K, V> и поддерживать методы:

- boolean hasNext()
- Node<K, V> next()

Используйте массив бакетов (Node<K, V>[] buckets), где каждый бакет представляет собой связанный список узлов с парами, имеющими одинаковый хеш-код.

Создайте внутренний класс Node<K, V>, представляющий узел в связанном списке:

- Поле K key — ключ.
- Поле V value — значение.
- Поле Node<K, V> next — ссылка на следующий узел.

Хеширование и обработка коллизий:

- Используйте метод hashCode() ключа для определения хеш-кода.
- Определите индекс бакета как hashCode % capacity. Убедитесь, что индекс неотрицательный.
- Обрабатывайте коллизии с помощью связных списков (метод цепочек).

Ресайзинг (изменение размера):

- Реализуйте динамическое изменение размера хеш-таблицы при достижении определенного коэффициента загрузки (например, 0.75).
- При ресайзинге перераспределите все пары по новой таблице.

Для демонстрации работы необходимо реализовать собственный класс с функцией main, в котором необходимо показать использование всех вышеперечисленных методов.

Часть 2 – по вариантам

Все три задания должны быть оформлены в виде отдельных классов. Для передачи входных параметров необходимо реализовать ввод с клавиатуры.

Stack

В данных заданиях необходимо использовать структуру данных Stack.

Задача 1: Вычисление максимальной глубины скобок

Дана строка, состоящая из символов '(' и ')'. Используя стек, определите максимальную глубину вложенности скобок.

Пример:
Вход: s = "(1+(2*3)+((8)/4))+1"
Выход: 3

Задача 2: Обращение строки

Дана строка s. Используя стек, обратите порядок символов в строке и верните результат.

Пример:
Вход: s = "hello"
Выход: "olleh"

Задача 3: Конвертация числа из десятичной системы в двоичную

Дано целое число n. Используя стек, конвертируйте его в двоичное представление.

Пример:
Вход: n = 10
Выход: "1010"

Задача 4: Удаление внешних скобок

Дана строка s, содержащая только символы '(' и ')', представляющая правильную скобочную последовательность. Используя стек, удалите внешние скобки из каждой пары скобок и верните новую строку без этих внешних скобок.

Пример:
Вход: s = "(()())(())"
Выход: "()()()"
Пояснение:
- Первая пара внешних скобок удалена из "(()())" -> "()()"
- Вторая пара внешних скобок удалена из "(())" -> "()"
- Итоговая строка: "()()()"

Задача 5: Сжатие строки

Дана строка s, состоящая из повторяющихся символов. Используя стек, сожмите строку, заменив последовательности одинаковых символов на один символ и количество его повторений.

Пример:
Вход: s = "aaabccccdd"
Выход: "a3b1c4d2"

Задача 6: Оценка обратной польской нотации

Дан массив строк tokens, представляющий обратную польскую нотацию арифметического выражения. Вычислите результат выражения.

Пример:
Вход: tokens = ["2", "1", "+", "3", "*"]
Выход: 9
Пояснение: ((2 + 1) * 3) = 9

Задача 7: Проверка корректности скобочной последовательности

Дана строка s, содержащая только символы '(', ')', '{', '}', '[' и ']'. Определите, является ли входная строка корректной скобочной последовательностью.

Пример:
Вход: s = "()[]{}"
Выход: true

Задача 8: Обратное чтение слов в предложении

Дана строка s, представляющая предложение. Используя стек, выведите предложение с обратным порядком слов.

Пример:
Вход: s = "Hello world this is Java"
Выход: "Java is this world Hello"

Задача 9: Базовый калькулятор II

Дана строка s, представляющая арифметическое выражение, содержащее числа, '+', '-', '*', '/' и пробелы. Вычислите результат выражения.

Пример:
Вход: s = "3+2*2"
Выход: 7

Задача 10: Удаление всех звёзд из строки

Дана строка s, в которой некоторые символы являются звёздочками '*'. Каждый символ звёздочки удаляет предыдущий символ. Верните итоговую строку после всех удалений.

Пример:
Вход: s = "leet**cod*e"
Выход: "lecoe"

Задача 11: Удаление пар повторяющихся символов

Дана строка s. Удалите из неё все пары соседних одинаковых символов повторно, пока это возможно, и верните итоговую строку.

Пример:
Вход: s = "abbba"
Выход: "a"
Пояснение:
"abbba" -> "aa" (удалены "bbb")
"aa" -> "" (удалены "aa")

Задача 12: Печать связанного списка в обратном порядке

Дан односвязный список head. Используя стек, выведите значения узлов в обратном порядке.

Пример:
Вход: head = [1, 2, 3]
Выход: [3, 2, 1]

Задача 13: Проверка возможности упрощения арифметического выражения

Дано арифметическое выражение в виде строки s, которое может содержать числа, '+', '-', '(' и ')'. Используя стек, упростите выражение, удалив лишние скобки.

Пример:
Вход: s = "((1+(2)))"
Выход: "1+2"

Задача 14: Удаление всех соседних дубликатов

Дана строка s, состоящая из строчных букв. Удалите все соседние дубликаты символов повторно, пока это возможно, и верните итоговую строку.

Пример:
Вход: s = "abbaca"
Выход: "ca"
Пояснение:
"abbaca" -> "aaca" (удалены "bb")
"aaca" -> "ca" (удалены "aa")

Задача 15: Проверка на палиндром

Дана строка s. Используя стек, определите, является ли строка палиндромом (читается одинаково слева направо и справа налево).

Пример:
Вход: s = "racecar"
Выход: true

Map

В данных заданиях необходимо использовать структуру данных Map.

Задача 1: Частота слов в строке

Дана строка s, состоящая из слов, разделенных пробелами. Подсчитайте частоту каждого слова и верните результат в виде Map.

Пример:
Вход: s = "this is a test this is only a test"
Выход: {this=2, is=2, a=2, test=2, only=1}

Задача 2: Подсчет частоты элементов

Дан массив целых чисел nums. Необходимо подсчитать, сколько раз каждый элемент встречается в массиве, и вернуть результат в виде Map.

Пример:
Вход: nums = [1, 2, 2, 3, 3, 3]
Выход: {1=1, 2=2, 3=3}

Задача 3: Проверка на перестановочные палиндромы

Даны две строки s1 и s2. Определите, можно ли одну строку переставить так, чтобы она стала палиндромом другой строки.

Пример:
Вход: s1 = "abc", s2 = "bca"
Выход: true

Задача 4: Римские цифры в целые числа

Дана строка, представляющая римское число. Преобразуйте его в целое число.

Пример:
Вход: s = "IX"
Выход: 9

Задача 5: Анаграммы палиндрома

Дана строка s. Определите, можно ли переставить буквы в строке так, чтобы получить палиндром.

Пример:
Вход: s = "carrace"
Выход: true
Пояснение: Можно получить "racecar"

Задача 6: Сумма двух чисел

Дан массив целых чисел nums и целое число target. Найдите индексы двух чисел, сумма которых равна target. Предполагается, что решение всегда существует.

Пример:
Вход: nums = [2, 7, 11, 15], target = 9
Выход: [0, 1]

Задача 7: Группировка слов по общей длине

Дан массив строк words. Сгруппируйте слова по их длине и верните результат в виде Map, где ключом является длина слова, а значением — список слов соответствующей длины.

Пример:
Вход: words = ["apple", "bat", "car", "dance", "egg"]
Выход: {3=[bat, car, egg], 5=[apple, dance]}

Задача 8: Слово с наибольшей частотой

Дана строка s, состоящая из слов, разделенных пробелами. Найдите слово, которое встречается наиболее часто. Если таких слов несколько, верните любое из них.

Пример:
Вход: s = "hello world hello"
Выход: "hello"

Задача 9: Суммирование значений по ключу

Дан список транзакций в виде массива строк transactions, где каждая строка имеет формат "ключ:значение". Объедините транзакции с одинаковыми ключами, суммируя их значения.

Пример:
Вход: transactions = ["apple:100", "banana:200", "apple:150", "banana:50"]
Выход: {apple=250, banana=250}

Задача 10: Распределение студентов по группам

Дан массив строк students, представляющий имена студентов, и массив строк groups, представляющий названия групп, куда они записаны (порядок соответствует). Создайте Map, где ключом является название группы, а значением — список студентов в этой группе.

Пример:
Вход:
students = ["Alice", "Bob", "Charlie", "David"]
groups = ["Math", "Science", "Math", "Literature"]
Выход:
{Math=[Alice, Charlie], Science=[Bob], Literature=[David]}

Задача 11: Проверка возможности построить заметку из журнала

Даны две строки ransomNote и magazine. Определите, можно ли составить строку ransomNote из символов строки magazine. Каждый символ в magazine может быть использован не более одного раза.

Пример:
Вход: ransomNote = "aabb", magazine = "ababab"
Выход: true

Задача 12: Словарь с обратным индексом

Дан массив строк words. Создайте Map, где ключом является слово, а значением — индекс этого слова в массиве.

Пример:
Вход: words = ["apple", "banana", "cherry"]
Выход: {apple=0, banana=1, cherry=2}

Задача 13: Проверка на взаимно однозначное отображение

Даны две строки s и t. Определите, существует ли взаимно однозначное соответствие между символами строк s и t.

Пример:
Вход: s = "paper", t = "title"
Выход: true

Задача 14: Первое уникальное число

Дан массив целых чисел nums. Найдите первое число, которое встречается в массиве только один раз. Если такого числа нет, верните -1.

Пример:
Вход: nums = [4, 5, 1, 2, 0, 4]
Выход: 5

Задача 15: Подсчет пар с одинаковой разностью

Дан массив целых чисел nums и целое число k. Найдите количество пар чисел в массиве, разность между которыми равна k.

Пример:

Вход: nums = [1, 2, 3, 4, 5], k = 2
Выход: 3
Пояснение: Пары — (1,3), (2,4), (3,5)

Содержание отчета

1. Правильно оформленный титульный лист.
2. Часть 1: краткое описание реализуемой структуры (теоретическая справка) и её листинг. Класс проверки вставлять не надо.
3. Часть 2: Название блока (структуры), текст варианта, листинг.
4. Листинг разработанной программы. Для сокращения места можно удалить комментарии, можно использовать `Consolas 10 – выглядит лучше и занимает меньше места.`