|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 6 |

**Название:** Коллекции

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Н.Б. Гендина | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | | П.В. Степанов |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2024

**Цель работы:** освоить базовые принципы работы с коллекциями на языке Java.

**Задание 1 вариант 4**

Сложить два многочлена заданной степени, если коэффициенты многочленов хранятся в объекте HashMap.

|  |
| --- |
| import java.util.\*;  public class Main {  public static void main(String[] args) {  HashMap<Integer, Integer> polynomial1 = createPolynomial(  new ArrayList<Integer>(Arrays.asList(1, 2, 5, 10, 0, 0, 15))  );  HashMap<Integer, Integer> polynomial2 = createPolynomial(  new ArrayList<Integer>(Arrays.asList(0, 0, 23, 3, 4, 0))  );  System.out.println("Исходные многочлены:");  System.out.println(polynomialToString(polynomial1));  System.out.println(polynomialToString(polynomial2));  HashMap<Integer, Integer> result1 = sum(polynomial1, polynomial2);  HashMap<Integer, Integer> result2 = sum(polynomial2, polynomial1);  System.out.println("Результат polynomial1 + polynomial2:");  System.out.println(polynomialToString(result1));  System.out.println("Результат polynomial2 + polynomial1:");  System.out.println(polynomialToString(result2));  }  public static HashMap<Integer, Integer> createPolynomial(List<Integer> coefficients) {  HashMap<Integer, Integer> result = new HashMap<>();  int degree = 0;  for(int coefficient: coefficients) {  result.put(degree, coefficient);  degree++;  }  return result;  }  public static HashMap<Integer, Integer> sum(HashMap<Integer, Integer> polynomial1, HashMap<Integer, Integer> polynomial2) {  int size = Math.max(polynomial1.size(), polynomial2.size());  HashMap<Integer, Integer> result = new HashMap<>();  for (int i = 0; i < size; i++) {  int coefficient = 0;  if (polynomial1.containsKey(i)) {  coefficient += polynomial1.get(i);  }  if (polynomial2.containsKey(i)) {  coefficient += polynomial2.get(i);  }  result.put(i, coefficient);  }  return result;  }  public static String polynomialToString(HashMap<Integer, Integer> polynomial) {  StringBuilder result = new StringBuilder();  for(int key : polynomial.keySet()) {  if (polynomial.get(key) != 0) {  if (!Objects.equals(String.valueOf(result), "")) {  result.append(" + ");  }  if (key == 0) {  result.append(String.format("%d", polynomial.get(key)));  } else {  result.append(String.format("%d\*x^%d", polynomial.get(key), key));  }  }  }  return String.valueOf(result);  }  } |

**Задание 1 вариант 5**

Умножить два многочлена заданной степени, если коэффициенты многочленов хранятся в различных списках.

|  |
| --- |
| import java.util.\*;  public class Main {  public static void main(String[] args) {  List<Integer> polynomial1 = new ArrayList<>(Arrays.asList(0, 1, 2));  List<Integer> polynomial2 = new ArrayList<>(Arrays.asList(1, 3));  System.out.println("Имеются многочлены:");  System.out.println(polynomialToString(polynomial1));  System.out.println(polynomialToString(polynomial2));    List<Integer> result1 = mul(polynomial1, polynomial2);  List<Integer> result2 = mul(polynomial2, polynomial1);  System.out.println("Результат polynomial1 \* polynomial2:");  System.out.println(polynomialToString(result1));  System.out.println("Результат polynomial2 \* polynomial1:");  System.out.println(polynomialToString(result2));  }  public static List<Integer> mul(List<Integer> polynomial1, List<Integer> polynomial2) {  int size = Math.max(polynomial1.size(), polynomial2.size());  int min\_size = Math.min(polynomial1.size(), polynomial2.size());  List<Integer> result = new ArrayList<>(Collections.nCopies((2 \* size - 1), 0));  if (min\_size != size) {  if (polynomial1.size() == min\_size) {  polynomial1.addAll(Collections.nCopies((size - min\_size), 0));  } else {  polynomial2.addAll(Collections.nCopies((size - min\_size), 0));  }  }  for (int i = 0; i < size; i++) {  for (int j = 0; j < size; j++){  result.set(i + j, result.get(i + j) + polynomial1.get(i) \* polynomial2.get(j));  }  }  return result;  }  public static String polynomialToString(List<Integer> polynomial1) {  StringBuilder result = new StringBuilder();  for (int i = 0; i < polynomial1.size(); i++){  if (polynomial1.get(i) != 0) {  if (!result.toString().isEmpty()) {  result.append(" + ");  }  if (i > 0) {  result.append(String.format("%dx^%d", polynomial1.get(i), i));  } else {  result.append(String.format("%d", polynomial1.get(i)));  }  }  }  return result.toString();  }  } |

**Задание 2 вариант 4**

Во входном файле хранятся наименования некоторых объектов. Построить список C1, элементы которого содержат наименования и шифры данных объектов, причем элементы списка должны быть упорядочены по возрастанию шифров. Затем “сжать” список C1, удаляя дублирующие наименования объектов.

|  |
| --- |
| import java.io.BufferedReader;  import java.io.FileReader;  import java.io.IOException;  import java.util.HashSet;  import java.util.LinkedList;  import java.util.List;  import java.util.Set;  public class Main {  record Node(String name, int code) {  public Node(String name) {  this(name, name.hashCode());  }  public int getCode() {  return this.code;  }  }  public static void main(String[] args) {  List<Node> list = new LinkedList<>();  try {  BufferedReader reader = new BufferedReader(new FileReader("input\_file.txt"));  String line = reader.readLine();  while (line != null) {  list.add(new Node(line));  line = reader.readLine();  }  list.sort((n1, n2) -> n2.getCode() - n1.getCode());  System.out.println("Исходный отсортированный список:");  System.out.println(list);  System.out.println("Исходное количество элементов:");  System.out.println(list.size());  list = removeDuplicates(list);  System.out.println("Итоговый список без дубликатов:");  System.out.println(list);  System.out.println("Мтоговое количество элементов:");  System.out.println(list.size());  reader.close();  } catch (IOException e) {  e.printStackTrace();  }  }  public static List<Node> removeDuplicates(List<Node> list) {  List<Node> result = new LinkedList<>();  Set<String> names = new HashSet<>();  for(Node node: list) {  if (names.contains(node.name)) {  continue;  }  names.add(node.name);  result.add(node);  }  return result;  }  } |

**Задание 2 вариант 5**

Во входном файле расположены два набора положительных чисел; между наборами стоит отрицательное число. Построить два списка C1 и С2, элементы которых содержат соответственно числа 1-го и 2-го набора таким образом, чтобы внутри одного списка числа были упорядочены по возрастанию. Затем объединить списки C1 и С2 в один упорядоченный список, изменяя только значения полей ссылочного типа.

|  |
| --- |
| import java.io.BufferedReader;  import java.io.FileReader;  import java.io.IOException;  import java.util.Collections;  import java.util.LinkedList;  import java.util.List;  public class Main {  public static void main(String[] args) {  List<Integer> c1 = new LinkedList<>();  List<Integer> c2 = new LinkedList<>();  try {  BufferedReader reader = new BufferedReader(new FileReader("input\_file.txt"));  String[] numbers = reader.readLine().split(" ");  boolean key = true;  for (String number: numbers) {  int el = Integer.parseInt(number);  if (el < 0){  key = false;  continue;  }  if (key) {  c1.add(el);  } else {  c2.add(el);  }  }  Collections.sort(c1);  Collections.sort(c2);  System.out.println("Исходный список 1:");  System.out.println(c1);  System.out.println("Исходный список 2:");  System.out.println(c2);  c1.addAll(c2);  Collections.sort(c1);  System.out.println("Итоговый соединенный список:");  System.out.println(c1);  reader.close();  } catch (IOException e) {  e.printStackTrace();  }  }  } |

**Вывод:** в ходе лабораторной работы были освоены базовые принципы работы с коллекциями на языке Java.