



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

ОТЧЕТ

по лабораторной работе № 6

Название: Коллекции

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Э.А. Гаджиев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Файл Main.java:

```
package Lab61;

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

import java.io.IOException;
import java.util.HashMap;
import java.util.Scanner;
import java.util.Vector;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String ioErrorMsg = "Ошибка из-за некорректного ввода. Завершение программы.";

        int degree_1 = 0;
        HashMap<String, Integer> polynom_1 = new HashMap<String, Integer>();
        int degree_2 = 0;
        HashMap<String, Integer> polynom_2 = new HashMap<String, Integer>();
        int res_degree = 0;
        HashMap<String, Integer> res_polynom = new HashMap<String, Integer>();

        System.out.print("Степень первого слагаемого-многочлена = ");
        try {
            degree_1 = scan.nextInt();
        } catch (Exception e) {
            System.out.println(ioErrorMsg);
            System.exit(1);
        }
        for (int i = 0; i <= degree_1; i++) {
            String coef = "a" + i;
            System.out.print("    Коэффициент " + coef + " = ");
            try {
                polynom_1.put(coef, scan.nextInt());
            } catch (Exception e) {
                System.out.println(ioErrorMsg);
                System.exit(1);
            }
        }

        System.out.print("Степень второго слагаемого-многочлена = ");
        try {
            degree_2 = scan.nextInt();
        } catch (Exception e) {
            System.out.println(ioErrorMsg);
            System.exit(1);
        }
        for (int i = 0; i <= degree_2; i++) {
            String coef = "a" + i;
            System.out.print("    Коэффициент " + coef + " = ");
            try {
                polynom_2.put(coef, scan.nextInt());
            } catch (Exception e) {
                System.out.println(ioErrorMsg);
                System.exit(1);
            }
        }

        res_degree = Integer.max(degree_1, degree_2);

        for (int i = 0; i <= res_degree; i++) {
            String coef = "a" + i;
            Integer a1 = polynom_1.get(coef);
            Integer a2 = polynom_2.get(coef);
            res_polynom.put(coef, (a1 == null ? 0 : a1) + (a2 == null ? 0 : a2));
        }

        System.out.print("Результат сложения: ");
        System.out.print(res_polynom.get("a0"));
        for (int i = 1; i <= res_degree; i++) {
            String coef = "a" + i;
            String term = res_polynom.get(coef) + "x^" + i;
            System.out.print(" + " + term);
        }
    }

    // Сложение с использованием вектора (не по заданию)
    // public static void foo() {
    //     Scanner scan = new Scanner(System.in);
    //     String ioErrorMsg = "Ошибка из-за некорректного ввода. Завершение программы.";
    //     //
    //     //
    //     int degree_1 = 0;
    //     //
    //     int degree_2 = 0;
    //     //
    //     int res_degree = 0;
    //     //
    //     System.out.print("Степень первого слагаемого-многочлена = ");
    //     //
    //     try {
    //         degree_1 = scan.nextInt();
    //         //
    //         if (degree_1 < 0)
    //             throw new IOException("Степень многочлена должна быть положительной.");
    //         //
    //     } catch (IOException e) {
    //         System.out.println(e.getMessage());
    //         //
    //         System.out.println(ioErrorMsg);
    //         //
    //         System.exit(1);
    //     } catch (Exception e) {
    //         System.out.println(ioErrorMsg);
    //         //
    //         System.exit(1);
    //     }
    //     //
    //     //
    //     Vector<Integer> polynom_1 = new Vector<>(degree_1);
    //     //
    //     for (int i = 0; i <= degree_1; i++) {
    //         System.out.print("    Коэффициент a" + i + " = ");
    //     }
    // }
}
```

```

//      try {
//          int coef = scan.nextInt();
//          if (degree_1 == 1 && degree_1 != 0 && coef == 0)
//              throw new IOException();
//          polynom_1.add(coef);
//      } catch (IOException e) {
//          System.out.println("Переменная со старшей степенью должна иметь ненулевой коэффициент.");
//          System.out.println(ioErrorMsg);
//          System.exit(1);
//      } catch (Exception e) {
//          System.out.println(ioErrorMsg);
//          System.exit(1);
//      }
//  }
//
//  System.out.print("Степень второго слагаемого-многочлена = ");
//  try {
//      degree_2 = scan.nextInt();
//      if (degree_2 < 0)
//          throw new IOException("Степень многочлена должна быть положительной.");
//  } catch (IOException e) {
//      System.out.println(e.getMessage());
//      System.out.println(ioErrorMsg);
//      System.exit(1);
//  } catch (Exception e) {
//      System.out.println(ioErrorMsg);
//      System.exit(1);
//  }
//
//  Vector<Integer> polynom_2 = new Vector<>(degree_2);
//  for (int i = 0; i <= degree_2; i++) {
//      System.out.print("      Коэффициент a" + i + " = ");
//      try {
//          int coef = scan.nextInt();
//          if (degree_2 == i && degree_2 != 0 && coef == 0)
//              throw new IOException();
//          polynom_2.add(coef);
//      } catch (IOException e) {
//          System.out.println("Переменная со старшей степенью должна иметь ненулевой коэффициент.");
//          System.out.println(ioErrorMsg);
//          System.exit(1);
//      } catch (Exception e) {
//          System.out.println(ioErrorMsg);
//          System.exit(1);
//      }
//  }
//
//  if (degree_1 < degree_2) {
//      for (int i = degree_1 + 1; i <= degree_2; i++) {
//          polynom_1.add(0);
//      }
//  } else if (degree_2 < degree_1) {
//      for (int i = degree_2 + 1; i <= degree_1; i++) {
//          polynom_2.add(0);
//      }
//  }
//
//  res_degree = Integer.max(degree_1, degree_2);
//  Vector<Integer> res_polynom = new Vector<>(res_degree);
//
//  for (int i = 0; i <= res_degree; i++) {
//      res_polynom.add(polynom_1.get(i) + polynom_2.get(i));
//  }
//
//  System.out.print("Результат сложения: ");
//  System.out.print(res_polynom.get(0));
//  for (int i = 1; i <= res_degree; i++) {
//      Integer coef = res_polynom.get(i);
//      if (coef != 0) {
//          String term = res_polynom.get(i) + "x^" + i;
//          System.out.print(" + " + term);
//      }
//  }
//  }
//  }
}

```

Файл Main.java:

```

package Lab62;

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

import java.io.IOException;
import java.util.Scanner;
import java.util.Vector;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String ioErrorMsg = "Ошибка из-за некорректного ввода. Завершение программы.";

        int degree_1 = 0;
        int degree_2 = 0;
        int res_degree = 0;

        System.out.print("Степень первого множителя-многочлена = ");
        try {
            degree_1 = scan.nextInt();
            if (degree_1 < 0)
                throw new IOException("Степень многочлена должна быть положительной.");
        } catch (IOException e) {
            System.out.println(e.getMessage());
            System.out.println(ioErrorMsg);
        }
    }
}

```

```

        System.exit(1);
    } catch (Exception e) {
        System.out.println(ioErrorMsg);
        System.exit(1);
    }

Vector<Integer> polynom_1 = new Vector<>(degree_1);
for (int i = 0; i <= degree_1; i++) {
    System.out.print("    Коэффициент a" + i + " = ");
    try {
        int coef = scan.nextInt();
        if (degree_1 == i && degree_1 != 0 && coef == 0)
            throw new IOException("Переменная со старшей степенью должна иметь ненулевой коэффициент.");
        polynom_1.add(coef);
    } catch (IOException e) {
        System.out.println(e.getMessage());
        System.out.println(ioErrorMsg);
        System.exit(1);
    } catch (Exception e) {
        System.out.println(ioErrorMsg);
        System.exit(1);
    }
}

System.out.print("Степень второго множителя-многочлена = ");
try {
    degree_2 = scan.nextInt();
    if (degree_2 < 0)
        throw new IOException("Степень многочлена должна быть положительной.");
} catch (IOException e) {
    System.out.println(e.getMessage());
    System.out.println(ioErrorMsg);
    System.exit(1);
} catch (Exception e) {
    System.out.println(ioErrorMsg);
    System.exit(1);
}

Vector<Integer> polynom_2 = new Vector<>(degree_2);
for (int i = 0; i <= degree_2; i++) {
    System.out.print("    Коэффициент a" + i + " = ");
    try {
        int coef = scan.nextInt();
        if (degree_2 == i && degree_2 != 0 && coef == 0)
            throw new IOException("Переменная со старшей степенью должна иметь ненулевой коэффициент.");
        polynom_2.add(coef);
    } catch (IOException e) {
        System.out.println(e.getMessage());
        System.out.println(ioErrorMsg);
        System.exit(1);
    } catch (Exception e) {
        System.out.println(ioErrorMsg);
        System.exit(1);
    }
}

if (degree_1 < degree_2) {
    for (int i = degree_1 + 1; i <= degree_2; i++) {
        polynom_1.add(0);
    }
} else if (degree_2 < degree_1) {
    for (int i = degree_2 + 1; i <= degree_1; i++) {
        polynom_2.add(0);
    }
}

res_degree = degree_1 + degree_2;
Vector<Integer> res_polynom = new Vector<>(res_degree + 1); //
for (int i = 0; i <= res_degree; i++) {
    res_polynom.add(0);
}

for (int i = 0; i <= degree_1; i++) {
    for (int j = 0; j <= degree_2; j++) {
        System.out.printf("a%d^%d = %dx^%d, b%d^%d = %dx^%d, curr_res a%d^%d = %d, mul_res = %d\n",
            // i, i, polynom_1.get(i), i,
            // j, j, polynom_2.get(j), j,
            // (i+j), (i+j), res_polynom.get(i+j),
            // polynom_1.get(i) * polynom_2.get(j));
        res_polynom.set(i + j, res_polynom.get(i+j) + (polynom_1.get(i) * polynom_2.get(j)));
    }
}

System.out.print("Результат умножения: ");
System.out.print(res_polynom.get(0));
for (int i = 1; i <= res_degree; i++) {
    Integer coef = res_polynom.get(i);
    if (coef != 0) {
        String term = res_polynom.get(i) + "x^" + i;
        System.out.print(" + " + term);
    }
}
}
}

```

Файл Main.java:

```
package Lab63;
```

```

/*
    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.File;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Iterator;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        File inputFile = new File("src/Lab63/input.txt");
        List<String> inputLines = null;
        try {
            inputLines = Files.readAllLines(inputFile.toPath());
        } catch (IOException e) {
            System.out.println("Ошибка чтения входного файла. " + e.getMessage());
            System.exit(1);
        }

        ArrayList<SomeObject> objects = new ArrayList<>();

        for (String line : inputLines) {
            String[] objInfo = line.split(" ");
            if (objInfo.length > 2) {
                System.out.println("Некорректный формат данных во входном файле.");
                System.exit(1);
            }
            objects.add(new SomeObject(objInfo[0], objInfo[1]));
        }

        objects.sort(Comparator.comparing(t -> t.code));
        System.out.println("Исходный список объектов: " + objects);

        // ListIterator<SomeObject> it = objects.listIterator();
        // while (it != objects.listIterator(objects.size()-1)) {
        int it = 0;
        while (it < objects.size() - 1) {
            if (objects.get(it + 1).equals(objects.get(it)))
                objects.remove(it);
            else
                it++;
        }
        System.out.println("Сжатый список объектов: " + objects);
    }
}

```

Файл SomeObject.java:

```

package Lab63;

import java.util.Objects;

public class SomeObject {
    public String name;
    public String code;

    public SomeObject(String name, String code) {
        this.name = name;
        this.code = code;
    }

    @Override
    public String toString() {
        return name + "[" + code + "]";
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        SomeObject that = (SomeObject) o;
        return Objects.equals(name, that.name) && Objects.equals(code, that.code);
    }

    @Override
    public int hashCode() {
        return Objects.hash(name, code);
    }
}

```

Файл Main.java:

```

package Lab64;

/*
    Во входном файле расположены два набора положительных чисел; между наборами стоит
    отрицательное число. Построить два списка C1 и C2, 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 C2 в один
    упорядоченный список, изменяя только значения полей ссылочного типа.
*/

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        File inputFile = new File("src/Lab64/input.txt");
    }
}

```

```

List<String> inputLines = null;
try {
    inputLines = Files.readAllLines(inputFile.toPath());
} catch (IOException e) {
    System.out.println("Ошибка чтения входного файла. " + e.getMessage());
    System.exit(1);
}

if (inputLines.size() > 1) {
    System.out.println("Некорректный формат входного файла");
    System.exit(1);
}

ArrayList<Integer> listC1 = new ArrayList<>();
ArrayList<Integer> listC2 = new ArrayList<>();
String[] inputNumbers = inputLines.get(0).split(" ");

int i = 0;
for (int num = Integer.parseInt(inputNumbers[i]); num >= 0; num = Integer.parseInt(inputNumbers[i])) {
    int j = 0;
    while (j < listC1.size() && num > listC1.get(j))
        j++;
    listC1.add(j, num);
    i++;
}
System.out.println("C1: " + listC1);
i++;
while (i < inputNumbers.length) {
    listC2.add(Integer.parseInt(inputNumbers[i]));
    i++;
}
System.out.println("C2: " + listC2);

ArrayList<Integer> listC3 = new ArrayList<>();
listC3.addAll(listC1);
listC3.addAll(listC2);
Collections.sort(listC3);
System.out.println("C1+C2: " + listC3);
}
}

```