

Звіт

Лабораторна робота №2

АЛГОРИТМІЧНА ДЕКОМПОЗИЦІЯ. ПРОСТІ АЛГОРИТМИ ОБРОБКИ ДАНИХ

Мета. Розробка простих консольних програм для платформи Java SE.

Розробник:

- Чугунов Вадим Юрійович;
- КІТ-119а;
- Варіант №24.

Вимоги:

1. Розробити та продемонструвати програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 10 зменшеного на одиницю номера студента в журналі групи.
2. Для визначення вхідних даних використовувати генератор псевдовипадкових чисел (`java.util.Random`) та забезпечити циклічне (принаймні десять ітерацій) знаходження результату рішення прикладної задачі.
3. Забезпечити виведення до консолі відповідних значень вхідних даних та результатів обчислень у вигляді таблиці.
4. Застосувати функціональну (процедурну) декомпозицію і забезпечити рішення прикладної задачі за допомогою відповідних методів.
5. Забороняється використання даних типу `String` та масивів при знаходженні рішення прикладної задачі.

ЗАВДАННЯ ДО РОБОТИ

Знайти позиції всіх найменших цифр в десятковому запису цілочисельного значення.

ОПИС ПРОГРАМИ

Опис змінних

```
Rand rand; // змінна для визначення вхідних даних
int number; // вхідне число
ArrayList<Integer> minNums; // колекція мінімальних цифр числа
```

Ієрархія та структура класів

class Main – точка входу в програму

ТЕКСТ ПРОГРАМИ

```
package ua.oop.khpi.chugunov02;

import java.util.Random;
import java.util.ArrayList;
import java.util.Collections;

public class Main {

    /**
     * An entry point - main method.
     *
     * @param args - arguments of main method
     */
    public static void main(String[] args)
    {
        Random rand = new Random();

        ArrayList<Integer> minNums = new ArrayList<Integer>();
        System.out.print("\n-----");
        System.out.print("\nInput number\t\tIndex of min numbers|");
        System.out.print("\n-----");

        final int numOfIters = 10;
        for (int i = 0; i < numOfIters; i++) {
            int number = rand.nextInt(100);
            findMinNums(number, minNums);
            printInfo(number, minNums);
            minNums.clear();
        }
    }
}
```

```

    }

    /**
     * Finds min digits of a number.
     *
     * @param num - number for search of min digits
     * @param array - an array of min digits
     */

    private static void findMinNums(final int num,final ArrayList<Integer> array) {
        ArrayList<Integer> arr = new ArrayList<Integer>();

        int number = num;

        final int constant = 10;

        while (number > 0) {
            arr.add(0, number % constant);

            number /= constant;
        }

        int min = Collections.min(arr);

        for (int i = 0; i < arr.size(); i++) {
            if (arr.get(i) == min) {
                array.add(i);
            }
        }
    }

    /**
     * Prints array of min digits.
     *
     * @param num - a source number
     * @param array - an array of min digits
     */

    private static void printInfo(final int num,final ArrayList<Integer> array) {

        System.out.print("\n"+num);

        System.out.print("\t\t\t\t\t\t\t");

        for (int i : array) {
            System.out.format("%d ", i);

```

```

    }

    System.out.print("\t\t\t\t\t");

    System.out.print("\n-----");

}

}

```

ВАРІАНТИ ВИКОРИСТАННЯ

Input number	Index of min numbers	
81	1	
8	0	
35	0	
48	0	
81	1	
17	0	
50	1	
24	0	
52	1	
87	1	
Process finished with exit code 0		

Рис. 1 – Результат роботи програми

Програма використовується для знаходження позицій всіх мінімальних цифр у числі.

ВИСНОВКИ

При виконанні лабораторної роботи набуто практичних навичок , щодо розробки консольних програм для платформи Java SE. Використані пакети: `java.util.Random`, `java.util.ArrayList`, `java.util.Collections`. Програма виконується без помилок.