

Лабораторна робота №2

Алгоритмічна декомпозиція. Прості алгоритми обробки даних.

Мета: Розробка простих консольних програм для платформи *Java SE*.

1 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.

1.1 Розробник

Виконавець: Сиромятников Марк

Група: КІТ-119а

Варіант: 19

1.2 Загальне завдання

Підрахувати кількість цифр, що відображаються за допомогою латинських букв в шістнадцятковому запису 10-значного цілого числа.

2 ОПИС ПРОГРАМИ

Було використано наступні засоби:

- *java.util.Random* – для рандомізації результатів виконання роботи.
- *System.out.printf()* – для виводу тексту у консоль.

2.1 Ієрархія та структура класів

Було створено 1 клас *C_Lab02* з двома методами.

2.2 Важливі фрагменти програми

```
static int countLettersInHex(int num) {  
    int counter = 0;           // initialization of counter  
  
    while (num > 0) {  
        if (num % 16 > 9) { // division by 16 with reminder  
            counter++;      // increasing counter  
        }  
        num = num/16;        // cutting off the tale of the  
number  
    }  
    return counter;           // function returning the counter  
}  
  
static void generatingNumbers()  
{  
    int temp = 0;  
  
    Random random = new Random();  
  
    for (int i = 0; i < 11; i++) {  
  
        int number = random.nextInt(); // generating number  
  
        if (number < 0) {  
            temp = number;  
            number *= -1;  
  
            int result = countLettersInHex(number);  
            System.out.printf("%20d", temp);  
            System.out.printf("%20x", number);  
            System.out.printf("%20d\n", result);  
        }  
        else {  
            int result = countLettersInHex(number);  
            // summoning the function of counting  
            System.out.printf("%20d", number);  
            System.out.printf("%20x", number);  
            System.out.printf("%20d\n", result); // calling the output of table  
        }  
    }  
}
```

}

3 ВАРІАНТИ ВИКОРИСТАННЯ

Розроблено 2 методи. Метод `generatingNumbers` циклічно генерує вхідні дані та викликає наступний метод для виконання індивідуального завдання. Метод `countLettersInHex` отримує рандомізовані дані та шляхом ділення на 16 з остачею та без визначає кількість літер латинського алфавіту у шістнадцятковому записі числа.

Результат роботи програми

Decemal	Hexidecimal	Quantity of letters
-1501570275	598024e3	1
-1696031899	6517649b	1
1687708496	64986350	0
749884392	2cb253e8	3
1213780898	4858d3a2	2
1198809647	4774622f	1
491718611	1d4f07d3	3
178380185	aa1dd99	4
-2026993504	78d17760	1

ВИСНОВКИ

У результаті виконання лабораторної роботи було створено просту консольну програму Java SE на основі простого алгоритму.