

## ЛАБОРАТОРНА РОБОТА №2

### Консольні програми для платформи Java SE. Прості алгоритми обробки даних.

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

**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. Забороняється використання даних типу *String* та масивів при знаходженні рішення прикладної задачі.
5. Застосувати функціональну (процедурну) декомпозицію і забезпечити рішення прикладної задачі за допомогою відповідних статичних методів.

## ЗМІСТ

1. Індивідуальне завдання . . . . .	3
2. Розробка програми . . . . .	3
2.1. Опис програми . . . . .	3
2.2. Важливі фрагменти програми . . . . .	3
3. Результати роботи . . . . .	4
4. Висновки . . . . .	4

## 1. Індивідуальне завдання

Знайти найбільшу цифру в десятковому запису цілочисельного значення.

## 2. Розробка програми

### 2.1. Опис програми

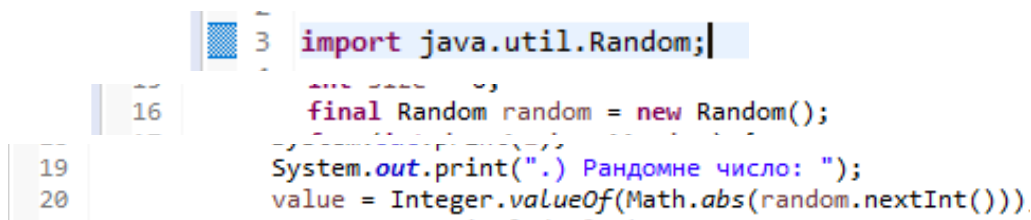
Для виконання прикладної задачі за принципом декомпозиції завдання було розділено на три підзадачі:

1. Отримання випадкового цілого числа;
2. Знаходження кількості розрядів у цьому числі;
3. Знаходження найбільшої цифри з застосованих у числі.

Наведені три дії повторюються 10 разів.

### 2.2. Важливі фрагменти програми

Для формування випадкових чисел було застосовано `java.util.Random` (рис.1)

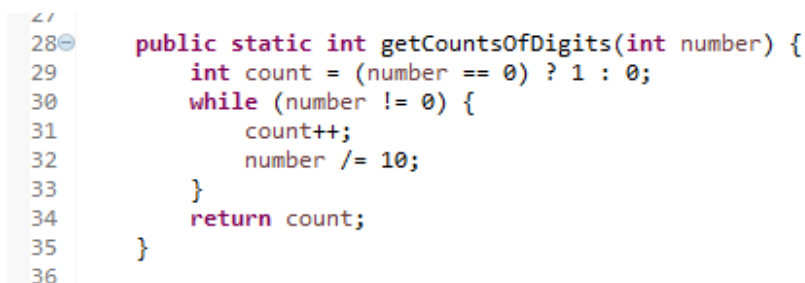
A screenshot of an IDE showing Java code. Line 16: `import java.util.Random;` Line 19: `final Random random = new Random();` Line 20: `System.out.print(". Рандомне число: ");` Line 21: `value = Integer.valueOf(Math.abs(random.nextInt()));`

```
3 import java.util.Random;
16
19 final Random random = new Random();
20 System.out.print(". Рандомне число: ");
    value = Integer.valueOf(Math.abs(random.nextInt()));
```

рис.1. Використання `java.util.Random`

Для отримання позитивного числа було застосовано статичну функцію `Math.abs()` (показано на попередньому екрані).

Для знаходження кількості розрядів у числі було написано спеціальну функцію (рис. 2)

A screenshot of an IDE showing a Java function. Line 28: `public static int getCountsOfDigits(int number) {` Line 29: `int count = (number == 0) ? 1 : 0;` Line 30: `while (number != 0) {` Line 31: `count++;` Line 32: `number /= 10;` Line 33: `}` Line 34: `return count;` Line 35: `}`

```
28 public static int getCountsOfDigits(int number) {
29     int count = (number == 0) ? 1 : 0;
30     while (number != 0) {
31         count++;
32         number /= 10;
33     }
34     return count;
35 }
36
```

рис. 2. Функція `getCountsOfDigits()`

Для знаходження максимального значення цифри було написано окрему функцію (рис. 3)

```

36
37 public static int findMax(int value, int size) {
38     int temp_1 = 0;
39     int temp_2 = 0;
40     for (int j = 0; j < size; j++) {
41         temp_1 = value % 10;
42         value /= 10;
43         if (temp_2 < temp_1)
44             temp_2 = temp_1;
45     }
46     return temp_2;
47 }
48 }

```

рис. 3. Функція findMax()

### 3. Результати роботи

Результати роботи програми можна побачити на рис. 4.

<terminated> Main (4) [Java Application] C:\Java\jdk\bin\javaw.exe (14 сент. 2017 г., 4:29:28)		
Лабораторна робота №2		
Номер прикладної задачі: 3		
№	Рандомне число	Найбільша цифра числа
1	1561104505	6
2	1655692354	9
3	1834459937	9
4	1397617524	9
5	1801378382	8
6	1825172343	8
7	528897899	9
8	1962504869	9
9	2118730531	8
10	1493141281	9

рис. 4. Результати роботи програми

### 4. Висновки

В результаті виконання лабораторної роботи ми навчились розробляти найпростіші консольні програми для платформи *Java SE*.