**Лабораторна робота №2**

[**Алгоритмічна декомпозиція. Прості алгоритми обробки даних**](https://oop-khpi.gitlab.io/#tasks)

**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](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D1%8F) і забезпечити рішення прикладної задачі за допомогою відповідних методів.
5. Забороняється використання даних типу [String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) та [масивів](https://docs.oracle.com/javase/specs/jls/se8/html/jls-10.html) при знаходженні рішення прикладної задачі.
   1. **Розробник**

* П.І.Б: Заночкин. Є. Д.
* Группа: КІТ-119а
* Варіант: 7
  1. **Загальне завдання**

1) Перевірити чи рівні значення першої та останньої, а також другої та передостанньої цифри в вісімковому запису 4-значного цілого числа.

1. **ОПИС ПРОГРАМИ**
   1. **Було використано наступні засоби:**

Random rand = new Random() - генерування випадкових чисел;

* 1. **Ієрархія та структура класів**

Було створено 1 клас Main, який має 3 метода та main.

* 1. **Важливі фрагменти програми**

/\*\*

\* Метод getRandom, який генерує чотиризначне число.

\*/

public static int getRandom(int min, int max)

{

Random rand = new Random();

int randomNum = rand.nextInt((max - min) + 1) + min;

return randomNum;

}

/\*\*

\* Метод decimalToOctal, який переводить число з десяткової у вісімкову систему счислення.

\*/

public static int decimalToOctal(int a)

{

int temp = a;

int newch = 0;

int mn = 1;

while (temp>0)

{

newch += (temp%8) \*mn;

temp/=8;

mn\*=10;

if(temp<1)

break;

}

return newch;

}

/\*\*

\* Метод compare, який перевіряє чи рівні значення першої та останньої, а також другої та передостанньої цифри.

\*/

public static int compare(int a)

{

int temp = a;

int temp3 = a;

int printNum = a;

int temp2;

int count = 0;

while(a>0)

{

a/=10;

count++;

}

System.out.print(printNum);

if (count == 5 &&(temp%10 == temp/10000))

System.out.print("\t\t| " + temp3%10 + " ~ " + temp3/10000);

else if (count == 5 &&(temp%10 != temp/10000))

System.out.print("\t\t| " + " - ");

if (count == 4 &&(temp%10 == temp/1000))

System.out.print("\t\t| " + temp3%10 + " ~ " + temp3/1000);

else if (count == 4 &&(temp%10 != temp/1000))

System.out.print("\t\t| " + " - ");

if(count == 5)

{

temp/=10;

temp2 = temp/10;

temp2/=10;

if (temp%10 == temp2%10)

System.out.print("\t\t " + temp%10 + " ~ " + temp2%10 + " |");

else if (temp%10 != temp2%10)

System.out.print("\t\t " + " - " + " |");

}

else

{

temp/=10;

temp2 = temp/10;

if (temp%10 == temp2%10)

System.out.print("\t\t " + temp%10 + " ~ " + temp2%10 + " |");

else if (temp%10 != temp2%10)

System.out.print("\t\t " + " - " + " |");

}

System.out.println(" ");

return 0;

}

1. **Результати роботи програми**

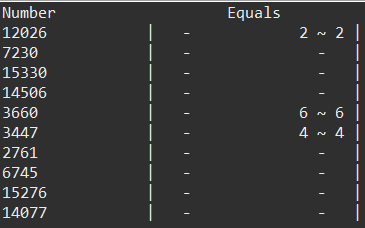
****

Рисунок 2.1 – Результат роботи програми у середовищі Eclipse

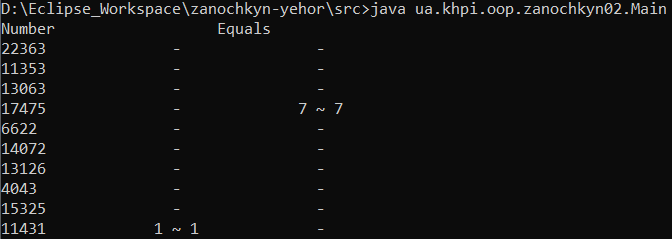
****

Рисунок 2.2 – Результат роботи програми у cmd

**Висновок**

Під час виконання лабораторної роботи було набуто навички роботи з простими алгоритмами обробки даних в середовищі Eclipse IDE.