Звіт

Автор: Лунгол О., КІТ101.8б

Дата: 28.09.2019

Лабораторна робота №2

АЛГОРИТМІЧНА ДЕКОМПОЗИЦІЯ. ПРОСТІ АЛГОРИТМИ ОБРОБКИ ДАНИХ

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

**Вимоги:**

* 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.
* Для визначення вхідних даних використовувати генератор псевдовипадкових чисел (java.util.Random) та забезпечити циклічне (принаймні десять ітерацій) знаходження результату рішення прикладної задачі.
* Забезпечити виведення до консолі відповідних значень вхідних даних та результатів обчислень у вигляді таблиці.
* 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.
* Забороняється використання даних типу [String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) та [масивів](https://docs.oracle.com/javase/specs/jls/se7/html/jls-10.html) при знаходженні рішення прикладної задачі.

ЗАВДАННЯ ДО РОБОТИ

Знайти кількість комбінацій 01 та 10 в двійковому запису цілого числа.

ОПИС ПРОГРАМИ

**Опис змінних**

Random rnd= **new** Random(System.*currentTimeMillis*());//генератор рандомних чисел

**int** number//діапазон рандомних чисел

**Ієрархія та структура класів**

**class** Main – точка входу в програму(Рисунок 1).

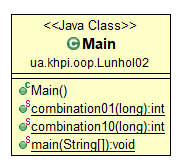


Рисунок 1 - **class** Main

ТЕКСТ ПРОГРАМИ

**package** ua.khpi.oop.Lunhol02;

**import** java.util.Random;

/\*\*

\* This is my Main class,here I will write my code

\* \*/

**public** **class** Main {

/\*\*

\* This function will return count\_bin

\* **@param** bin converts a number to a binary number

\* **@return** count\_bin

\* \*/

**public** **static** **int** combination01(**long** zero\_one) {

String bin = Long.*toBinaryString*(zero\_one);

**int** count\_bin = 0;

**for**(**int** i = 0; i < bin.length()-1; i++) {

**if**(bin.charAt(i) == '0' && bin.charAt(i+1) == '1') {

count\_bin++;

}

}

**return** count\_bin;

}

/\*\*

\* This function will return count\_bin

\* **@param** bin converts a number to a binary number

\* **@return** count\_bin

\* \*/

**public** **static** **int** combination10(**long** zero\_one) {

String bin = Long.*toBinaryString*(zero\_one);

**int** count\_bin = 0;

**for**(**int** i = 0; i < bin.length()-1; i++) {

**if**(bin.charAt(i) == '1' && bin.charAt(i+1) == '0') {

count\_bin++;

}

}

**return** count\_bin;

}

/\*\*

\* **@author** ALEXANDRA

\* **@version** 2.0

\* Here start point of the program

\* **@param** args command line values

\* **@param** rnd generates random numbers

\* **@param** number int value that randomly selects numbers in the specified interval

\* \*\*/

**public** **static** **void** main(String[] args) {

Random rnd= **new** Random(System.*currentTimeMillis*());

**for**(**int** i=1;i<=10;i++) {

**int** number = 50 + rnd.nextInt(170 - 50 + 1);

System.***out***.println(i+".Number: " + number + ", BinaryNumber:"+Long.*toBinaryString*(number)+ ", Combination\_01: "+*combination01*(number)+" ,Combination\_10: "+*combination10*(number) );

}

}

}

РОБОТА ПРОГРАМИ

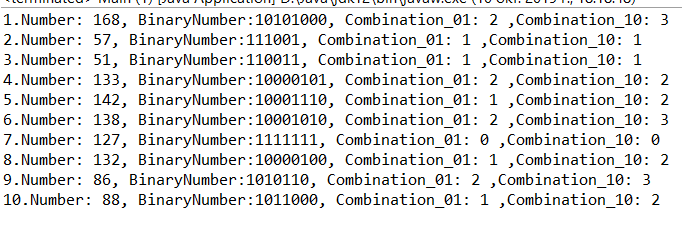


Рисунок 2 – Результат роботи програми

ВАРІАНТИ ВИКОРИСТАННЯ

Використати програму у корисних цілях скоріш за все не вдасться. Але можна використати схему знаходження кількості комбінацій у двійковому коді.

ВИСНОВКИ

При виконанні лабораторної роботи набуто практичних навичок, щодо розробки програм за допомогою JDK платформи Java SE. Програма виконується без помилок.