Лабораторна робота №1

**СТРУКТУРА ПРОГРАМИ МОВОЮ JAVA. ТИПИ ДАНИХ, ЛІТЕРАЛИ, ОПЕРАЦІЇ І ОПЕРАТОР.**

**Мета:** Ознайомлення з JDK платформи Java SE та середовищем розробки Eclipse IDE.

**ВИМОГИ**

**Розробник:**

- Веремчук Дарина Анатоліївна;

- КІТ-119д;

- Варіант №5.

**Загальне завдання:**

1) Вирішити три прикладні задачі на мові Java в середовищі Eclipse.

2) Продемонструвати покрокове виконання програми та результати роботи в режимі налагодження, не використовуючи виведення до консолі.

3) Виконати компіляцію і запуск програми в командному рядку за допомогою відповідних утиліт JDK.

**Завдання до лабораторної роботи:**

1) Обрати тип змінних та встановити за допомогою констант та літералів початкові значення:

* число, що відповідає номеру залікової книжки за допомогою шістнадцяткового літералу;
* число, що відповідає номеру мобільного телефона (починаючи з 380...) за допомогою десяткового літералу;
* число, яке складається з останніх двох ненульових цифр номера мобільного телефону за допомогою двійкового літералу;
* число, яке складається з останніх чотирьох ненульових цифр номера мобільного телефону за допомогою вісімкового літералу;
* визначити збільшене на одиницю значення залишку від ділення на 26 зменшеного на одиницю номера студента в журналі групи;
* символ англійського алфавіту в верхньому регістрі, номер якого відповідає знайденому раніше значенню.

2) Використовуючи десятковий запис цілочисельного значення кожної змінної знайти і підрахувати кількість парних і непарних цифр.

3) Використовуючи двійковий запис цілочисельного значення кожної змінної підрахувати кількість одиниць.

**ОПИС ПРОГРАМИ**

**Опис змінних:**

final short bookNumber; // номер залікової книжки

final long phoneNumber; // номер мобільного телефону

final byte binaryPhoneOfNumber; // дві останні ненульові цифри номеру телефону

final short octalPhoneOfNumber; // останні 4 ненульові цифри номеру телефону

final byte numberInJournal; // порядковий номер у журналі

final byte constant; // константа для рішення формули

final byte number; // зберігання результату формули

final char symbol; // отримання з формули номер букви

**Ієрархія та структура класів:**

**class Main** – точка входу в програму.

**ТЕКСТ ПРОГРАМИ**

package ua.oop.khpi.veremchuk01;

public class Main {

/\*\*

\* An entry point - main method.

\*

\* @param args - arguments of main method

\*/

public static void main(String[] args) {

final short bookNumber = 0x3AEB;

final long phoneNumber = 380994711009L;

final byte binaryPhoneOfNumber = 0b10011;

final short octalPhoneOfNumber = 035353;

final byte numberInJournal = 5;

final byte constant = 26;

final byte number = ((numberInJournal - 1 ) % constant) + 1;

final char symbol = (char) number + 65;

String strBookNumber;

strBookNumber = Short.toString(bookNumber);

Amount.EvenOddNum(strBookNumber);

String strPhoneNumber;

strPhoneNumber = Long.toString(phoneNumber);

Amount.EvenOddNum(strPhoneNumber);

String strBinaryPhoneNumber;

strBinaryPhoneNumber = Byte.toString(binaryPhoneOfNumber);

Amount.EvenOddNum(strBinaryPhoneNumber);

String strOctalPhoneNumber;

strOctalPhoneNumber = Short.toString(octalPhoneOfNumber);

Amount.EvenOddNum(strOctalPhoneNumber);

String strConstant;

strConstant = Byte.toString(binaryPhoneOfNumber);

Amount.EvenOddNum(strConstant);

String strSymbol;

strSymbol = Integer.toString((int) symbol);

Amount.EvenOddNum(strSymbol);

strBookNumber = Integer.toBinaryString(bookNumber);

Amount.OneCountBinaryNumber(strBookNumber);

strPhoneNumber = Long.toBinaryString(phoneNumber);

Amount.OneCountBinaryNumber(strPhoneNumber);

strBinaryPhoneNumber = Integer.toBinaryString(binaryPhoneOfNumber);

Amount.OneCountBinaryNumber(strBinaryPhoneNumber);

strOctalPhoneNumber = Integer.toBinaryString(octalPhoneOfNumber);

Amount.OneCountBinaryNumber(strOctalPhoneNumber);

strConstant = Integer.toBinaryString(binaryPhoneOfNumber);

Amount.OneCountBinaryNumber(strConstant);

strSymbol = Integer.toBinaryString((int) symbol);

Amount.OneCountBinaryNumber(strSymbol);

}

}

class Amount{

/\*\*

\* The method that counting and printing even and odd numbers

\*

\* @param num - an argument that checks for parity

\*/

public static void EvenOddNum ( String num) {

byte EvenNum = 0; // count of even numbers

byte OddNum = 0; // count of odd numbers

for (byte i = 0; i < num.length(); i++) { //checking if a number is even or not

if (num.charAt(i) % 2 == 0) {

EvenNum++;

} else {

OddNum++;

}

}

System.out.println("Number: " + num);

System.out.println("Number of even numbers: " + EvenNum);

System.out.println("Number of odd numbers: " + OddNum);

System.out.println("\n ");

}

/\*\*

\* The method that counting binary ones in our numbers

\*

\* @param num - the argument in which we count the number of ones in a binary number

\*/

public static void OneCountBinaryNumber(String num)

{

byte CountOne = 0; // count of binary ones

for (byte i = 0; i < num.length(); i++) //comparison with 1

{

if (num.charAt(i) == '1')

{

CountOne++;

}

}

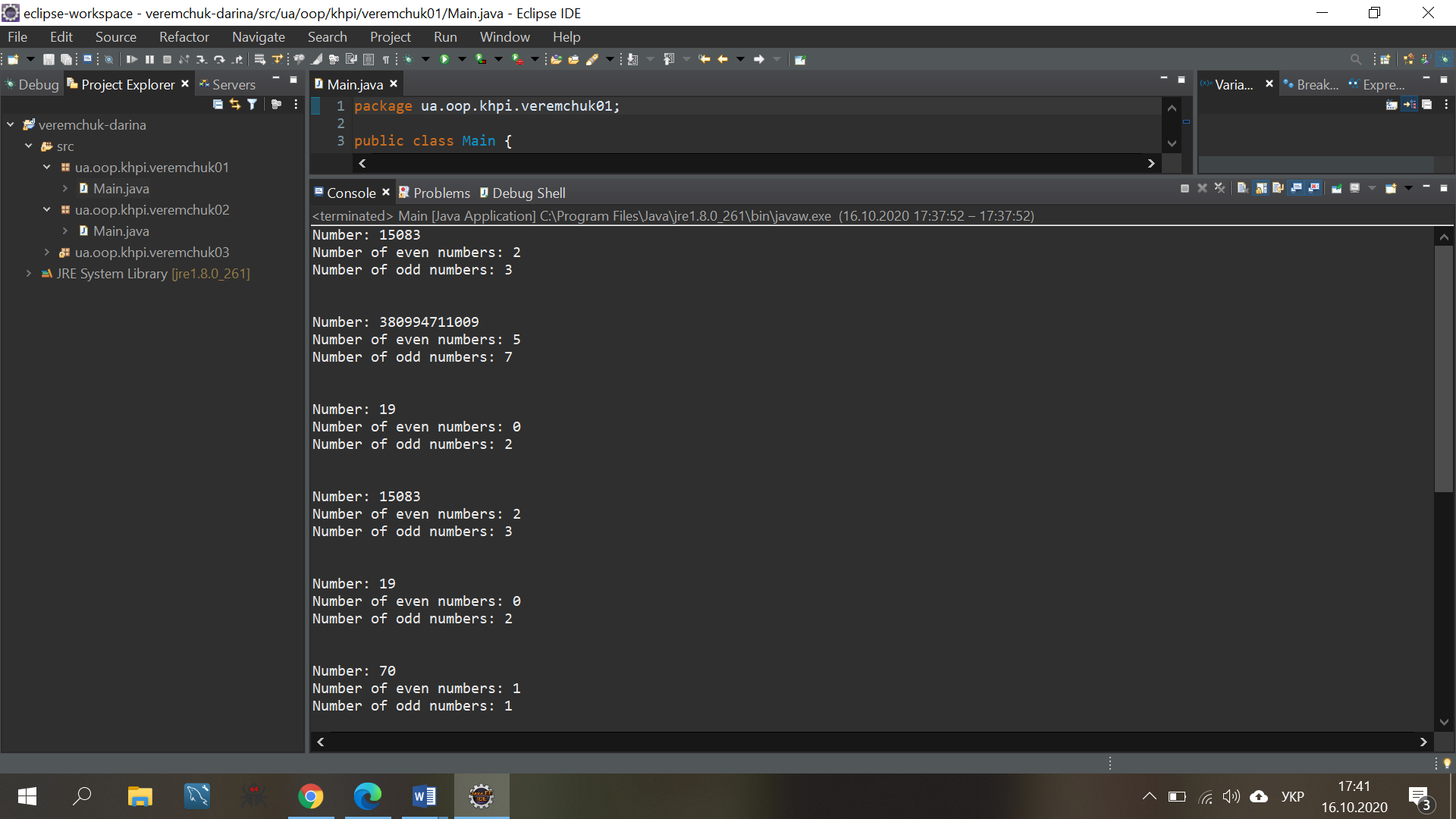
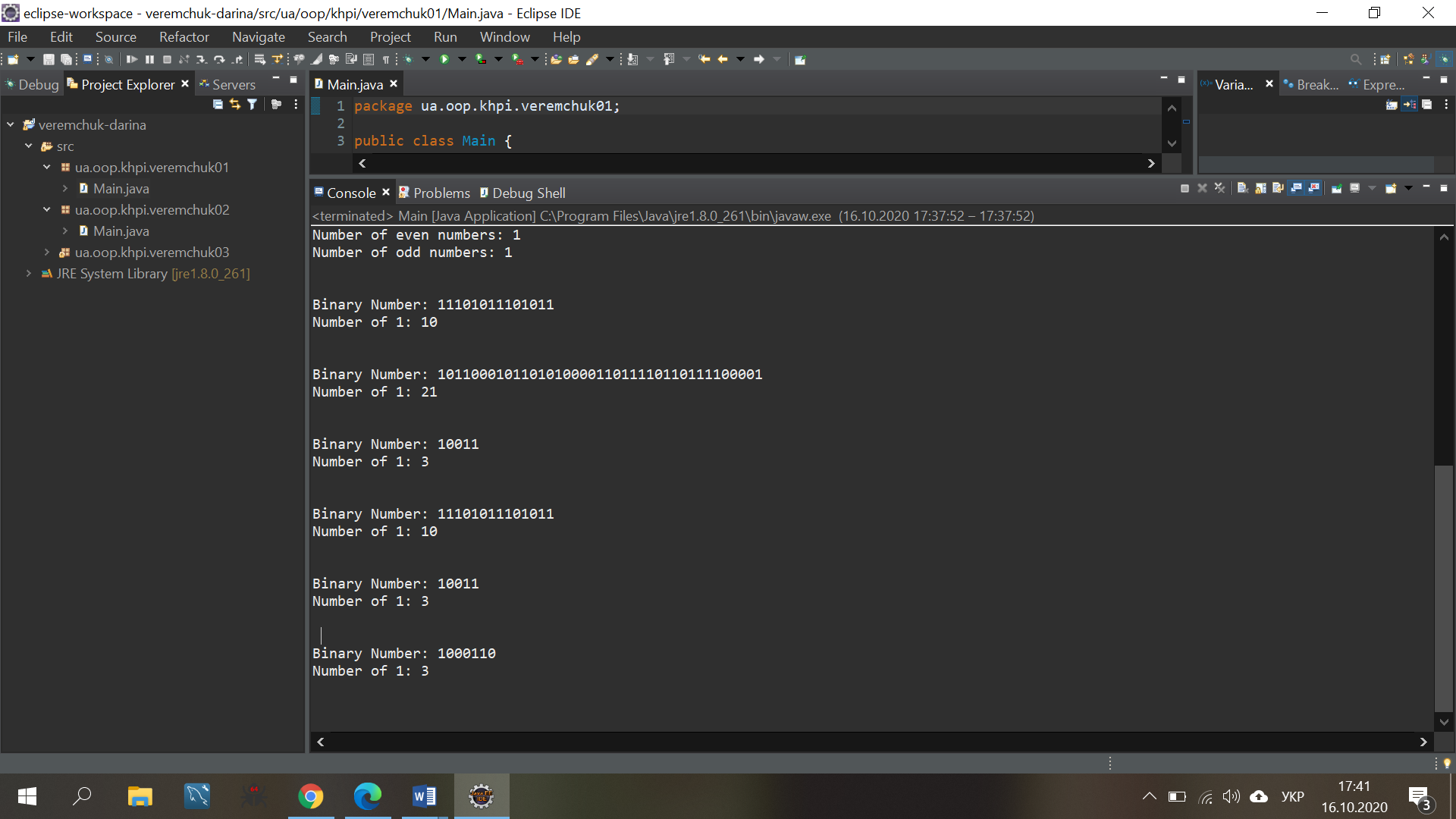
System.out.println("Binary Number: " + num);

System.out.println("Number of 1: " + CountOne);

System.out.println("\n ");

}

}

а) б)

Рисунок 1.1 – Результат роботи програми

**ВАРІАНТИ ВИКОРИСТАННЯ**

Можна використати схему знаходження кількості парних, непарних і взагалі будь-яких цифр у великій кількості чисел.

**ВИСНОВОК**

Ознайомилася та отримала практичні навички розробки програм за допомогою JDK платформи Java SE та середовищем розробки Eclipse IDE. Програма виконується без помилок.