**Лабораторна робота №1. Структура програми мовою Java. Типи даних, літерали, операції і оператори**

**Мета:** Ознайомлення з JDK платформи Java SE та середовищем

розробки Eclipse IDE.

* **Вимоги**
* **. Розробник**
* Косінов Владислав Дмитрович;
* Студент групи КІТ-120б;
* Перевірив: Молчанов Георгій Ігорович.

**. Загальне завдання:**

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

1.2 число, що відповідає номеру залікової книжки за допомогою шістнадцяткового літералу;

1.3 число, що відповідає номеру мобільного телефона (починаючи з 380...) за допомогою десяткового літералу;

1.4 число, яке складається з останніх двох ненульових цифр номера мобільного телефону за допомогою двійкового літералу;

1.5 число, яке складається з останніх чотирьох ненульових цифр номера мобільного телефону за допомогою вісімкового літералу;

1.6 визначити збільшене на одиницю значення залишку від ділення на 26 зменшеного на одиницю номера студента в журналі групи;

1.7 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**

2.1 **. Засоби ООП**

1.Метод  main.  Використовується для виконання індивідуального завдання.

2.2 **. Ієрархія та структура класів**

В данній лабораторній роботі використовується один клас Main . В ньому відбувається виконання завдання.

public static void main(String[] args) {

        final short bookNumber = 0x3A3B;

        final long phoneNumber = 380506015527L;

        final byte binaryPhoneOfNumber = 0b11011;

        final short octalPhoneOfNumber = 012627;

        final byte numberInJournal = 8;

        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);

    }

Рис.1 - клас Main

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 ");

    }

Рис.2 - клас Amount

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 ");

    }

Рис.3 - клас OneCountBinaryNumber

* **Варіанти використання**

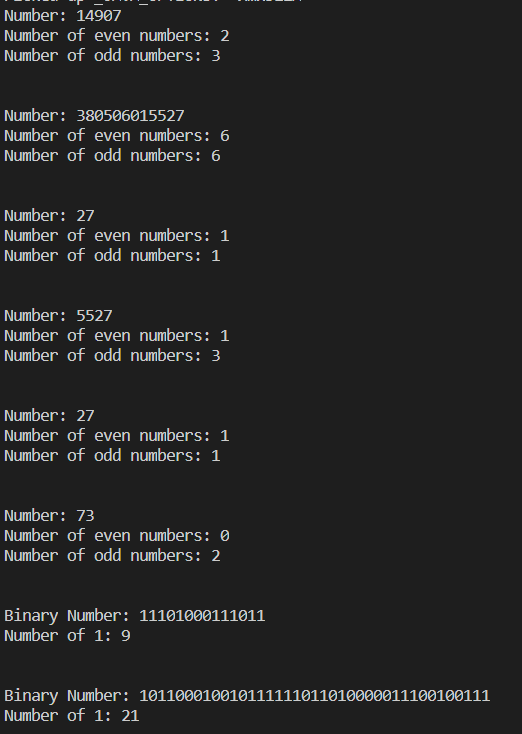
****

Рис.4 – результат програми

**Висновки**

На цій лабораторній роботі ми виконували розробку консольних програм для платформи Java SE. В ході виконання індивідуального завдання було розроблено програму по пошуку найменшої цифри в числі та її позиції. В звіті було продемемонстровано роботу розробленої програми.