Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до лабораторної роботи № 1

з дисципліни

​Прикладне програмування

на тему:

“ Основи Java”

Виконав: студент гр. ОІ-25

**Паньків Олег**

Прийняв: Римар Ю.М.

Львів – 2025

Мета роботи: закріпити знання основ java на практиці.

**Хід роботи**

Варіант 16

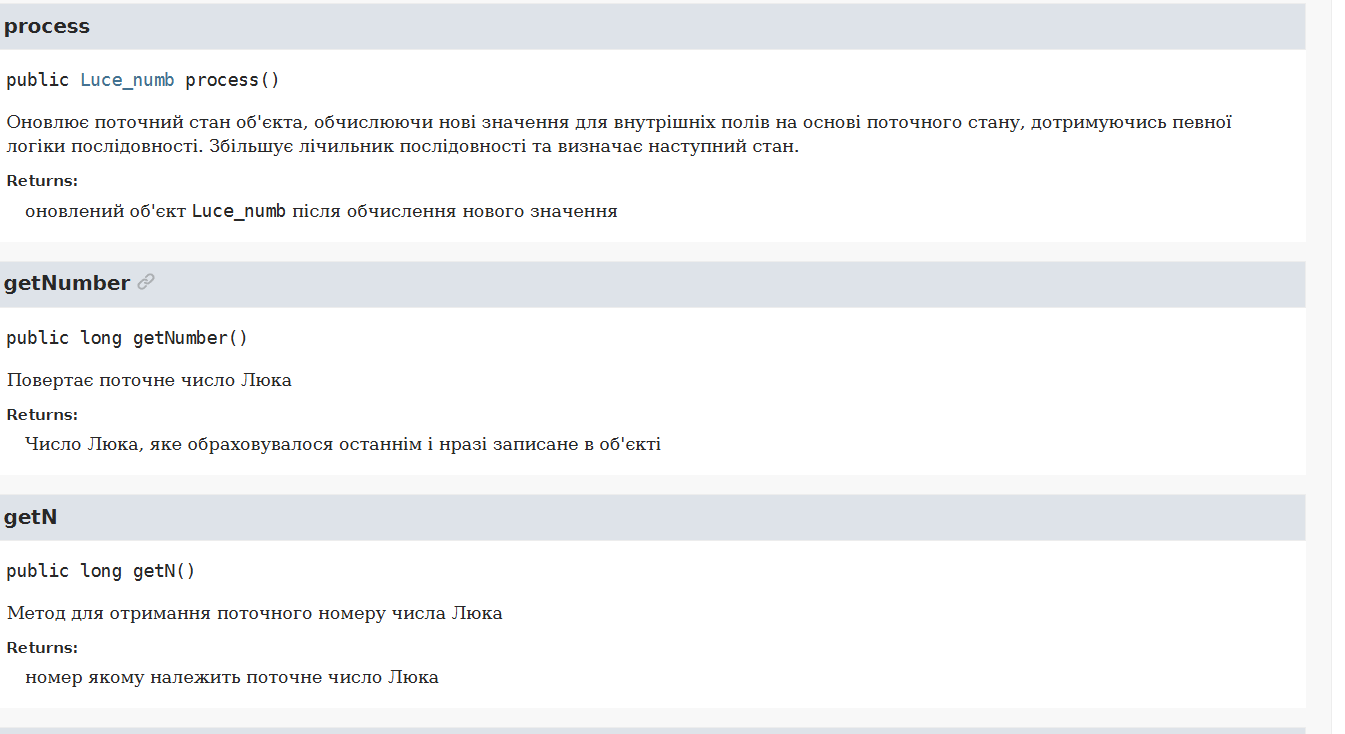
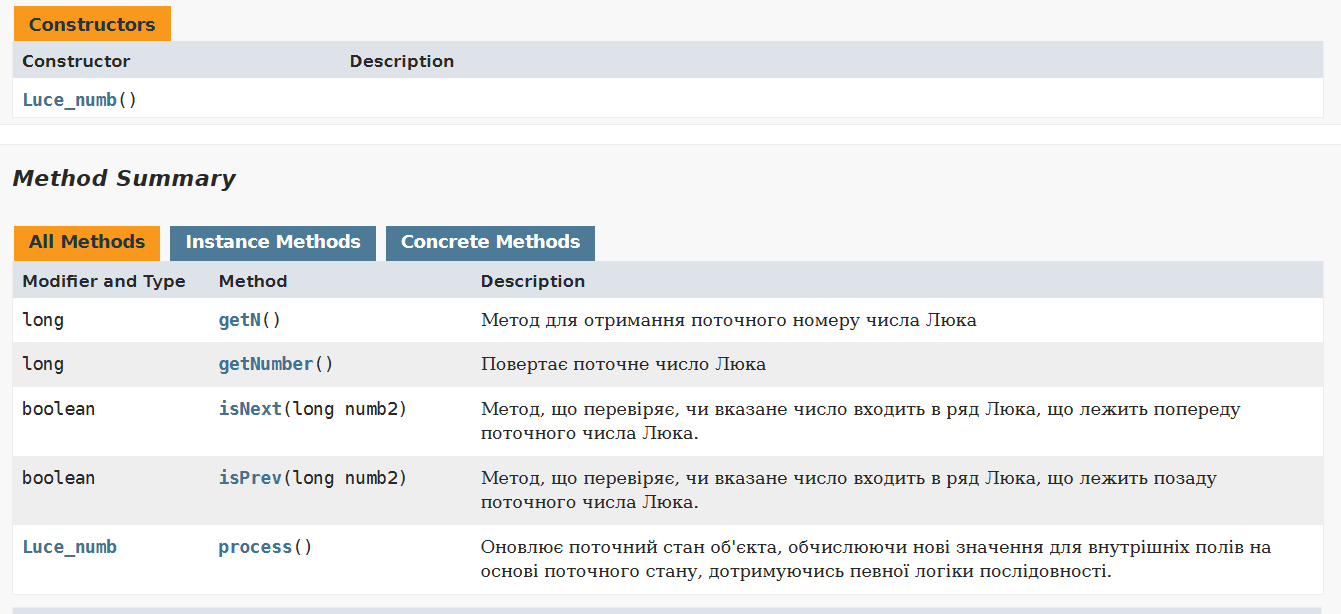
Завдання 1. Визначити, які числа серед перших M чисел Люка можна задати у формі n(n+1)/2 (трикутні числа).

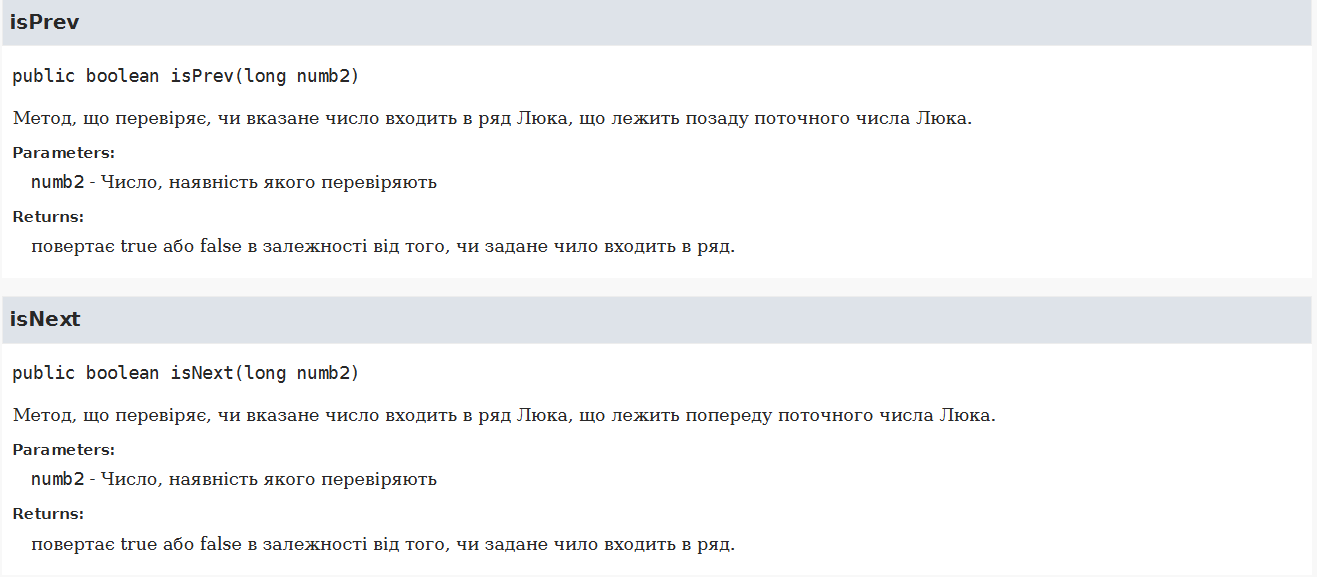
Лістинг програми та результат виконання:

package org.example;  
import java.util.Scanner;  
  
  
class Main{  
 public static void main(String[] args) {  
 System.*out*.println("enter M max(91): ");  
 Scanner scan = new Scanner(System.*in*);  
 int M , count =0;  
 long numb1 , numb2 , n;  
 M = scan.nextInt();  
 Luce\_numb luce = new Luce\_numb();  
 n = luce.getN();  
 System.*out*.println("n Числа люка - n(n+1)/2");  
 while(n<M){  
 luce=luce.process();  
 n = luce.getN();  
 numb1 = luce.getNumber();  
 numb2 = n\*(n+1)/2;  
 if(numb1 == numb2){  
 count++;  
 }  
 else if(luce.isNext(numb2)|luce.isPrev(numb2)){  
 count++;  
 }  
 System.*out*.println(n+" "+numb1+"-"+numb2+" ");  
 }  
 System.*out*.println("M - "+M);  
 System.*out*.println("Чисел які можна подати у вигляді n(n+1)/2 - " + count);  
 scan.close();  
 }  
}

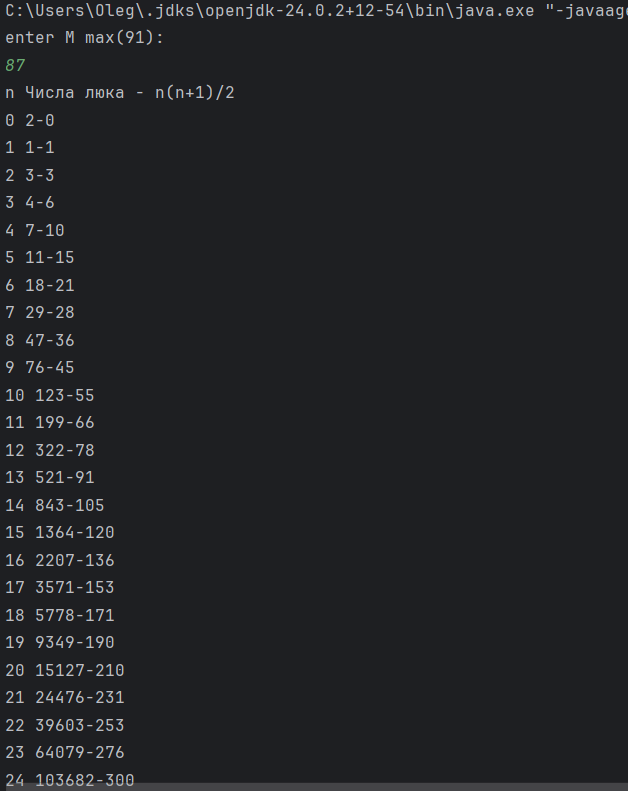
package org.example;  
  
public class Luce\_numb{  
 long i = -1 , j =3 , next , prev , n = -1;  
  
 */\*\*  
 \* Оновлює поточний стан об'єкта, обчислюючи нові значення  
 \* для внутрішніх полів на основі поточного стану, дотримуючись певної логіки послідовності.  
 \* Збільшує лічильник послідовності та визначає наступний стан.  
 \*  
 \* @return оновлений об'єкт {@code Luce\_numb} після обчислення нового значення  
 \*/* public Luce\_numb process(){  
 long numb1 ;  
 prev = i;  
 numb1 = i+j;  
 j=i;  
 i=numb1;  
 next = numb1+j;  
 n++;  
 return this ;  
}  
  
 */\*\*  
 \* Повертає поточне число Люка  
 \* @return Число Люка, яке обраховувалося останнім і нразі записане в об'єкті  
 \*/* public long getNumber(){  
 return i;  
}  
  
 */\*\*  
 \* Метод для отримання поточного номеру числа Люка  
 \* @return номер якому належить поточне число Люка  
 \*/* public long getN(){  
 return n;  
}  
  
 */\*\*  
 \* Метод, що перевіряє, чи вказане число входить в ряд Люка, що лежить позаду поточного числа Люка.  
 \* @param numb2 Число, наявність якого перевіряють  
 \* @return повертає true або false в залежності від того, чи задане чило входить в ряд.  
 \*/* public boolean isPrev( long numb2) {  
 long prev2 = prev , i2 = i;  
 if (i > numb2) {  
 if (numb2 == prev2) {  
 return true;  
 } else if (numb2 < prev2) {  
 do {  
 prev2 = i2 - prev2;  
 i2 = i2 - prev2;  
  
 } while (numb2 < prev2);  
 if (numb2 == prev2) {  
 return true;  
 }  
 }  
 }  
 return false;  
}  
  
 */\*\*  
 \* Метод, що перевіряє, чи вказане число входить в ряд Люка, що лежить попереду поточного числа Люка.  
 \* @param numb2 Число, наявність якого перевіряють  
 \* @return повертає true або false в залежності від того, чи задане чило входить в ряд.  
 \*/* public boolean isNext(long numb2){  
 long i2 = i , next2 = next;  
 if(i<numb2) {  
 if (numb2 == next) {  
 return true;  
 } else if (numb2 > next) {  
 do {  
 next2 = i2 + next2;  
 i2 = next2 - i2;  
 } while (numb2 > next2);  
 if (numb2 == next2) {  
 return true;  
 }  
 }  
 }  
 return false;  
 }  
}

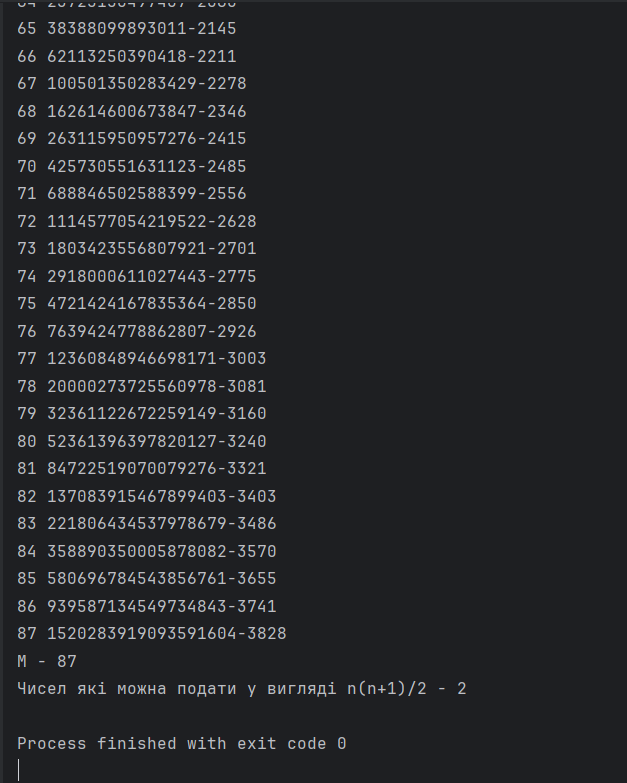
JavaDoc:



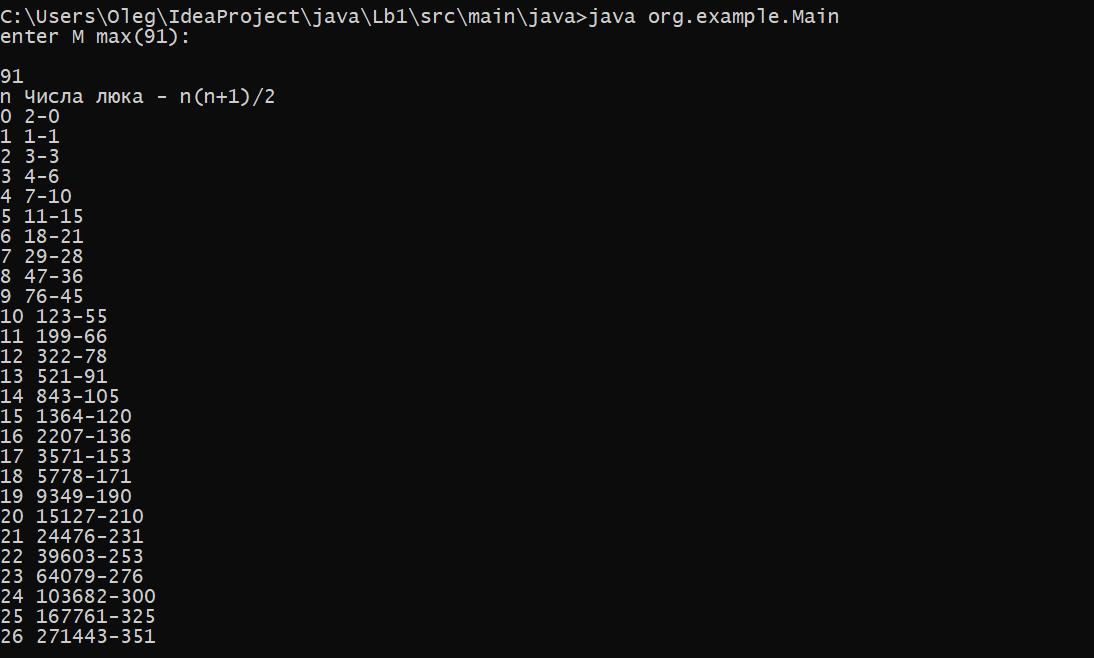


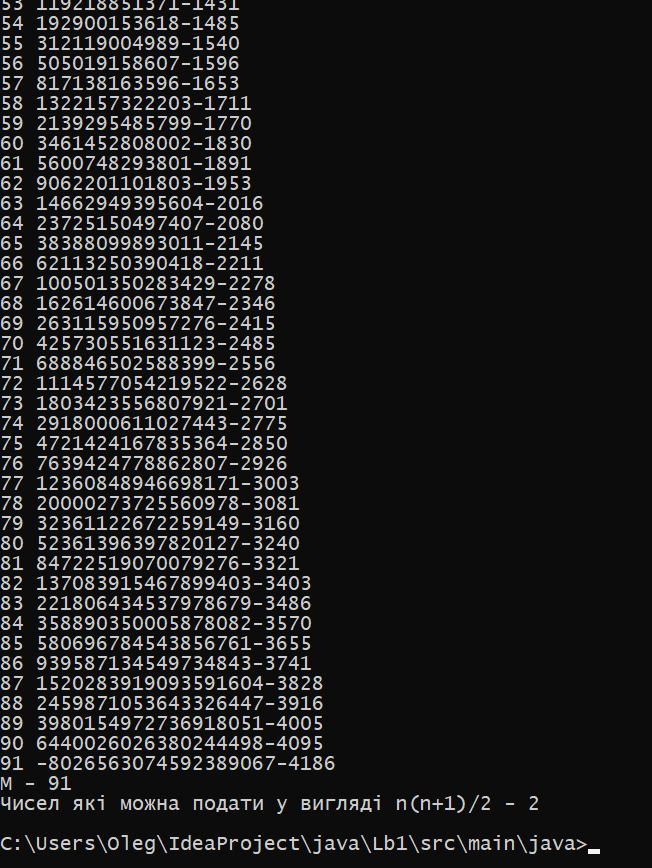
Результати виконання:





Запуск через командний рядок





Висновок: виконуючи цю лабораторну роботу , я закріпив на практиці знання про основи java. Зверніть увагу , максимальне число Люка обчислюване цією програмою є 90 за порядковим номером, така ситуація зумовлена тим, що розмірна сітка числа з типом long дозволяє записати число не більше за 9,223,372,036,854,775,807.