**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**імені ІГОРЯ СІКОРСЬКОГО”**

**ЗВІТ**

**з лабораторної роботи №2**

**з навчальної дисципліни “****Рекурсивні алгоритми”**

**Тема:**

**Рекурсивні алгоритми**

**Виконав студент групи ТР–12**

Каркушевський Владислав

Лабораторну роботу захищено

з оцінкою \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Перевірив доцент кафедри**

\_\_\_\_\_\_\_\_\_\_\_ Андрій ОНИСЬКО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 року

**Київ 2022**

**Мета:** Метою лабораторної роботи є набуття практичних навичок з рекурсивними функціями.

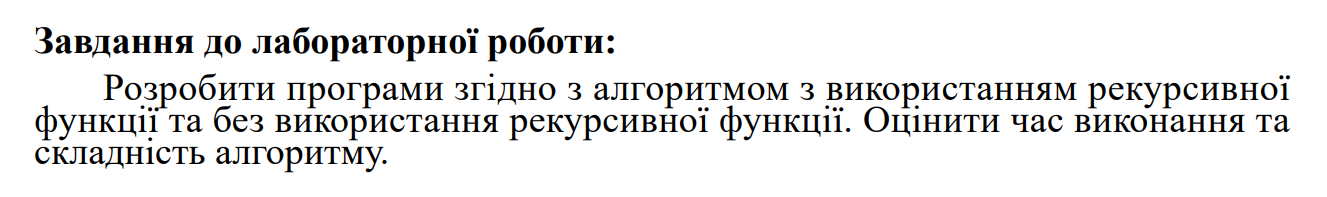
**Теоретична частина**

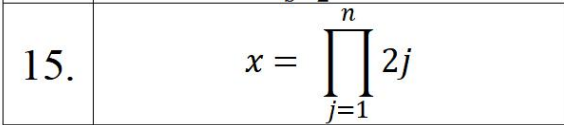
**Рекурсiєю** називається такий спосiб органiзацiї обробки даних, при якому програма (функцiя) викликає сама себе безпосередньо, або з iнших програм (функцiй). Рекурсивний алгоритм – це алгоритм, в описі якого прямо або побічно міститься зверненнядо самого себе.

**Прямою** рекурсією називається рекурсія, при якій всередині тіла деякої  
функції міститься виклик тієї ж функції.

**Непрямою** рекурсією називається рекурсія, що здійснює рекурсивний  
виклик функції шляхом ланцюга викликів інших функцій. При цьому всі  
функції ланцюга, що здійснюють рекурсію, вважаються рекурсивними.

**Завдання**





**Результати виконання**

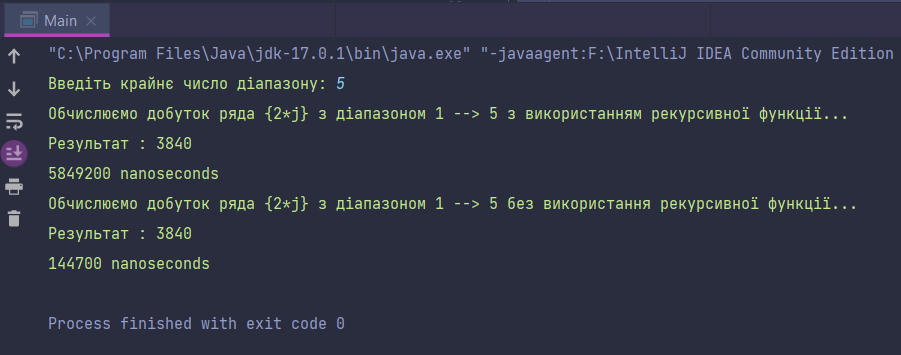
Класс Main

*import* java.util.Scanner;*//імпортуємо сканер  
  
public class* Main {  
  
 *public static void* main(String[] args) {  
 System.***out***.print("\u001B[32m");  
 Task task = *new* Task();  
  
 Scanner sc = *new* Scanner(System.***in***);  
 System.***out***.print("Введіть крайнє число діапазону: ");  
 *int* n = sc.nextInt();  
  
  
 System.***out***.printf("Обчислюємо добуток ряда {2\*j} з діапазоном 1 --> %d з використанням рекурсивної функції...\n", n);  
 *long* start = System.*nanoTime*();*//розпочинаємо вимір часу* System.***out***.println( "Результат : " + task.*recursion*(n));  
 *long* finish = System.*nanoTime*();*//закінчуємо вимір часу* System.***out***.printf( "%d nanoseconds\n", finish-start);*//виведення* System.***out***.printf("Обчислюємо добуток ряда {2\*j} з діапазоном 1 --> %d без використання рекурсивної функції...\n", n);  
 *long* start1 = System.*nanoTime*();*//розпочинаємо вимір часу* System.***out***.println( "Результат : " + task.*not\_recursion*(n));  
 *long* finish1 = System.*nanoTime*();*//закінчуємо вимір часу* System.***out***.printf( "%d nanoseconds\n", finish1-start1);*//виведення часу* }  
}

Класс Task

*public class* Task {*//створимо окремий клас task для методів обчислення завдання  
  
 public static int* recursion (*int* n ){*//створимо метод з використанням рекурсії  
 if*( n < 0 ){  
 System.***out***.print("\u001B[31m");  
 System.***out***.println("error");  
 System.*exit*(0);  
 }  
 *if*(n ==0 ) {  
 *return* 1;  
 }  
 *else* {  
 *return* 2 \* n \* *recursion*(n - 1);  
 }  
 }  
  
 *public static int* not\_recursion (*int* n ){*//створимо метод без використання рекурсії  
 int* result = 1 ;  
  
 *for*(*int* j = 1 ; j <= n ; j++ ){  
 result \*= 2 \* j ;  
 }  
 *return* result ;  
 }  
  
}

**Скріншоти виконання завдання**

****

**Складність алгоритму**

|  |  |  |
| --- | --- | --- |
| **Значення n** | **Час для рекурсії** | **Час для цикла** |
| **5** | **4250500** | **135600** |
| **10** | **4472400** | **117900** |
| **15** | **4395900** | **144500** |
| **20** | **4086800** | **135800** |
| **25** | **4522000** | **109100** |
| **30** | **4535500** | **117400** |

**Висновок**

На цій лабораторній роботі ми розробили програму згідно з алгоритмом з використанням рекурсивної функції та без використання рекурсивної функції. Оцінити час виконання та складність алгоритму.