МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ”ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних

систем та мереж

Звіт з лабораторної роботи №6

з дисципліни "Технології програмування в розподілених інформаційних системах"

на тему:

**"Розроблення Web-сервісу з використанням не реляційної БД"**

Варіант №24

Виконав:

Студент групи КН-38

Стасишин О.З.

Прийняв:

доцент каф. ІСМ, к.т.н.

Щербак С. С.

ЛЬВІВ – 2018

**Мета роботи:** набуття практичних навичок роботи з рекурсивними функціями.

**Завдання на роботу:**

На основі додатку з першої лабораторної роботи розробити програму для обчислення математичних виразів згідно алгоритмів з використанням рекурсивної функції та без використання рекурсивної функції. Оцінити час виконання та складність алгоритму. Стан об’єктів програми зберігати в БД Firebase.

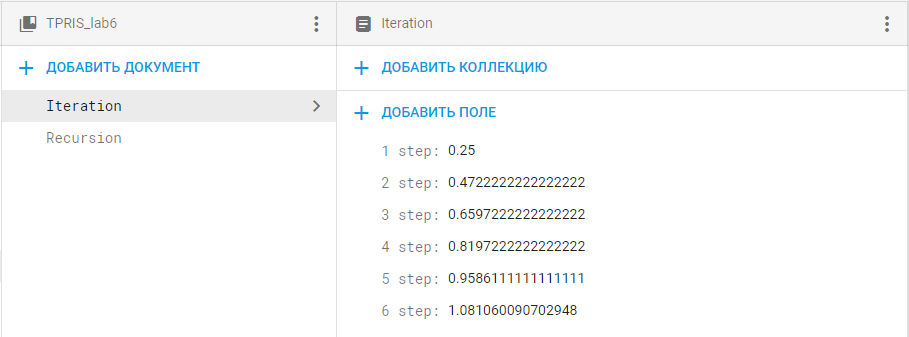
**Індивідуальне завдання:**



**Текст програми на мові Java:**

import com.google.api.core.ApiFuture;  
import com.google.auth.oauth2.GoogleCredentials;  
import com.google.cloud.firestore.DocumentReference;  
import com.google.cloud.firestore.Firestore;  
  
import com.google.cloud.firestore.WriteResult;  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.FirebaseOptions;  
import com.google.firebase.cloud.FirestoreClient;  
  
import java.io.FileInputStream;  
import java.io.InputStream;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.Scanner;  
  
public class Main {  
  
 public Map<String , Double> dataIteration = new HashMap<>();  
 public Map<String , Double> dataRecursion = new HashMap<>();  
  
  
 private void addDataBaseIteration(DocumentReference documentReference) throws Exception{  
 ApiFuture<WriteResult> addData = documentReference.set(dataIteration);  
 addData.get().getUpdateTime();  
 }  
  
 private void addDataBaseRecursion(DocumentReference documentReference) throws Exception{  
 ApiFuture<WriteResult> addData = documentReference.set(dataRecursion);  
 addData.get().getUpdateTime();  
 }  
  
 private void recursionData(DocumentReference documentReference , int r) throws Exception{  
 recursion(r);  
 addDataBaseRecursion(documentReference);  
 }  
  
 private double recursion(int r) throws Exception{  
 double sum = 0;  
 if(r < 2) {  
 return sum;  
 }  
 else {  
 dataRecursion.put((r-1) + " step" , sum += (r - 1) / Math.*pow*(r, 2) + recursion(r - 1));  
 }  
 return sum;  
 }  
  
 private void iteration(DocumentReference documentReference , int r) throws Exception{  
 double sum = 0;  
 for(int p = 2; p <= r; p++){  
 sum += (p - 1)/Math.*pow*(p , 2);  
 dataIteration.put((p-1) + " step" , sum);  
 }  
 addDataBaseIteration(documentReference);  
 }  
  
  
  
 public static void main(String[] args) throws Exception {  
  
 InputStream serviceAccount = new FileInputStream("C:\\Users\\User\\Downloads\\TPRIS-fff677a0ea61.json");  
 GoogleCredentials credentials = GoogleCredentials.*fromStream*(serviceAccount);  
 FirebaseOptions options = new FirebaseOptions.Builder()  
 .setCredentials(credentials)  
 .build();  
 FirebaseApp.*initializeApp*(options);  
  
 Firestore db = FirestoreClient.*getFirestore*();  
  
 DocumentReference documentReferenceIteration = db.collection("TPRIS\_lab6").document("Iteration");  
 DocumentReference documentReferenceRecursion = db.collection("TPRIS\_lab6").document("Recursion");  
 Scanner scanner = new Scanner(System.*in*);  
 int r = scanner.nextInt();  
 Main main = new Main();  
 main.iteration(documentReferenceIteration , r);  
 main.recursionData(documentReferenceRecursion , r);  
 }  
}

**Результати виконання програми:**





*Рис. 1. Виведення результату виконання програми*

**Висновок:** програма призначена для роботи з виразом, що обчислюється за допомогою рекурсії та ітеративним способом. Програма обчислює та зберігає свій стан з БД. На цій лабораторній роботі було набуто практичних навичок роботи з рекурсивними функціями. Програма написана на мові Java у середовищі IntelliJ.