



Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

## **ЛАБОРАТОРНА РОБОТА №5**

з дисципліни «Основи програмування - 2»

Тема: «Потоки вводу-виводу»

### **Виконали:**

студенти групи ІА-31  
Клим'юк В.Л, Самелюк А.С,  
Дук М.Д, Сакун Д.С

### **Перевірив:**

асистент кафедри ІСТ  
Степанов А. С.

**Тема:** Потоки вводу-виводу

**Мета:** Мета лабораторної роботи полягає у вивченні основних концепцій потоків введення-виведення в мові програмування Java, їх використання для зчитування даних з клавіатури та запису даних у файли. У процесі виконання роботи студенти отримають навички роботи з різними типами потоків, розуміння принципів їх роботи та здатність застосовувати їх у власних програмах для взаємодії з користувачем та зберігання даних на диску.

### Хід роботи

1. Пригадати API для здійснення операцій вводу-виводу. Особливу увагу звернути на такі класи та інтерфейси:

- InputStream
  - FileInputStream
- Reader
  - FileReader
- AutoCloseable
  - Closable
- IOException
  - FileNotFoundException

2. Пригадати Collection Framework. Особливу увагу звернути на такі класи та інтерфейси:

- Map
  - HashMap
  - TreeMap
- List
  - ArrayList

3. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1.

- Завдання має бути реалізовано як окремий клас.
- Клас має складатись щонайменше з таких методів:

- `public static void main(String[] args)` - точка входу.

- Метод, що реалізує задане завдання. Метод має перевіряти аргументи та у разі їх помилковості аварійно закінчувати свою роботу шляхом викидання стандартного виключення `IllegalArgumentException` або `NullPointerException`. В разі неможливості виконання операції, метод повинен викидати `IOException` або `FileNotFoundException`. В жодному разі цей метод не повинен напряду взаємодіяти з користувачем через консоль або інший UI (ніколи не змішуйте бізнес-логіку та користувацький інтерфейс).

- Клас може містити інші допоміжні методи.

- При виконанні завдань слід звернути увагу на ефективність з точки зору швидкодії:

- операції вводу/виводу слід здійснювати через буфер;

- програму потрібно написати так, щоб зчитування кожного фрагмента файлу здійснювалось лише один раз (тобто від початку файлу до кінця за один прохід без повторного зчитування).

```
import java.io.*;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;

public class WordFrequency {

    public static void main(String[] args) {
        try {
            String filename = "prog.txt";
            ArrayList<String> rarestWords = rarestWords(filename);
            System.out.println("Rarest word(s):");
            for (String word : rarestWords) {
                System.out.println(word);
            }
        } catch (IllegalArgumentException | NullPointerException | IOException e) {
            System.out.println("An error occurred: " + e.getMessage());
        }
    }

    public static ArrayList<String> rarestWords(String filename) throws IOException {
        if (filename == null) {
            throw new NullPointerException("Filename cannot be null");
        }

        Map<String, Integer> wordFrequency = new HashMap<>();
```



**Висновки:** Під час виконання лабораторної роботи ми успішно вивчили основні концепції потоків введення-виведення в мові програмування Java. Ми набули навичок роботи з різними типами потоків, зокрема зчитування даних з клавіатури та запису даних у файли. Ці знання будуть корисними для подальшого програмування, дозволяючи нам ефективно взаємодіяти з користувачем та здійснювати зберігання даних на диску.