



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

## **Лабораторна робота № 3**

Потоки вводу-виводу

Виконали  
студенти групи ІА-23:  
Архип'юк К. О.,  
Богаченко А. В.,  
Кашуб'як С. М.  
Єрмак Д.Р.

Перевірив:  
Колеснік В. М.

Київ 2023

## Хід роботи:

1. Ознайомитись з API класів та інтерфейсів для здійснення операцій вводу-виводу. Особливу увагу звернути на такі класи та інтерфейси:

- InputStream
  - FileInputStream
- OutputStream
  - FileOutputStream
- Reader
  - FileReader
- Writer
  - FileWriter
- AutoCloseable
  - Closable
- IOException

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

- Кожне завдання має бути реалізовано як окремий клас.
- Кожен клас має складатись щонайменше з двох 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. При виконанні завдань 1-6 слід використовувати клас `BufferedReader` та `BufferedWriter`, а при 7-11 – ні в якому разі не намагатись обробляти усі байти по одному, а використовувати методи `read(byte[] b)` та `write(byte[] b)`, які працюють з масивами.

## Варіант 4

4	<code>void copyToUpperCase(String source, String destination)</code> Скопіювати текстовий файл <i>&lt;source&gt;</i> у <i>&lt;destination&gt;</i> , замінивши в процесі усі символи у верхній регістр.
10	<code>void split(String source, String destinationPrefix, long maxSize)</code> Розбити файл <i>&lt;source&gt;</i> на кілька файлів (1 та більше) з назвами <i>&lt;destinationPrefix&gt;+“.000”</i> , <i>&lt;destinationPrefix&gt;+“.001”</i> , <i>&lt;destinationPrefix&gt;+“.002”</i> , ..., кожен з яких буде мати розмір, не більший ніж <i>&lt;maxSize&gt;</i> .

## Task4.java

```
import java.io.*;

public class Task4 {
    public static void main(String[] args) {
        try {
            copyToUpperCase("...\\LAB 14\\src\\filesTask4\\inputTask4.txt",
                           "...\\LAB 14\\src\\filesTask4\\outputTask4.txt");
        } catch (IOException e) {
            System.out.println("Exception! " + e.getMessage());
        }
    }

    public static void copyToUpperCase(String source, String destination)
    throws IOException {
        try (BufferedReader inputReader = new BufferedReader(new
        FileReader(source));
            BufferedWriter outputWriter = new BufferedWriter(new
        FileWriter(destination))) {
            String line;
            while ((line = inputReader.readLine()) != null) {
                outputWriter.write(line.toUpperCase() + "\n");
            }
        }
    }
}
```

## inputTask4.txt

```
1 Hello Biden it's Zelensky we need 5 million rockets to bomb Donetsk children Slava Ukraine
```

## outputTask4.txt

```
1 HELLO BIDEN IT'S ZELENSKY WE NEED 5 MILLION ROCKETS TO BOMB DONETSK CHILDREN SLAVA UKRAINE
```

## Task10.java

```
import java.io.*;
import java.nio.file.*;

public class Task10 {
    public static void main(String[] args){
        try {
            split("...\\LAB 14\\src\\filesTask10\\inputTask10.txt",
                 "...\\LAB 14\\src\\filesTask10\\outputTask10", 25);
        } catch (IOException e) {
            System.out.println("Exception! " + e.getMessage());
        }
    }

    public static void split(String source, String destinationPrefix, long
    maxSize) throws IOException {
        Path sourcePath = Paths.get(source);
        byte[] buffer = new byte[(int) maxSize];
        int partNumber = 0;
        try (InputStream in = Files.newInputStream(sourcePath)) {
            int bytesRead;
            while ((bytesRead = in.read(buffer)) > 0) {
                String destinationFileName = String.format("%s.%03d",
                destinationPrefix, partNumber++);
            }
        }
    }
}
```

```

        Path destinationPath = Paths.get(destinationFileName);
        try (OutputStream out =
Files.newOutputStream(destinationPath)) {
            out.write(buffer, 0, bytesRead);
        }
    }
}
}
}

```

inputTask10.txt

```

1 Hello Biden it's Zelensky we need 5 million rockets to bomb Donetsk children Slava Ukraine

```

outputTask10.000

```

1 Hello Biden it's Zelensky we n

```

outputTask10.001

```

1 eed 5 million rockets to bomb

```

outputTask10.002

```

1 Donetsk children Slava Ukraine

```

**Висновок:** згідно з метою нашої лабораторної роботи, ми успішно ознайомились API класів та інтерфейсів для здійснення операцій введення та виведення даних. Ми ознайомились з такими класами та інтерфейсами, як `InputStream`, `FileInputStream`, `OutputStream`, `FileOutputStream`, `Reader`, `FileReader`, `Writer`, `FileWriter`, `AutoCloseable`, `Closable` та `IOException`. Це дало нам підґрунтя для подальшого вивчення і розуміння роботи з введенням та виведенням даних в програмах на мові програмування та покращило наші навички кодування на Java.