



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

Лабораторна робота №7  
**Java**  
Варіант 3 (103)

Виконав:  
Студент групи ІА-33  
Ничик О. О.

Перевірив:  
Лесик В. О.

**Мета роботи:** практично дослідити можливості рефлексії, освоїти створення та налаштування лог-журналів за допомогою Log4j, набути навички створення додатків з можливістю локалізації.

## Хід роботи

### Task1

#### ReflectionDemo.java

```
public static void main(String[] args) throws Exception {
    Scanner sc = new Scanner(System.in);

    String literal = "Hello world";
    System.out.println("Літерал до зміни: " + literal);
    changeStringValue(literal, newValue: "CHANGED");
    System.out.println("Літерал після зміни: " + literal);

    System.out.print("\nВведіть рядок: ");
    String input = sc.nextLine();

    System.out.print("На що замінюємо: ");
    String replacement = sc.nextLine();

    System.out.println("Рядок до зміни: " + input);
    changeStringValue(input, replacement);
    System.out.println("Рядок після зміни: " + input);

    System.out.println("\nУвага: такий підхід ламає \"immutability\" String і "
        + "залежить від версії JVM. Використовувати тільки для демонстрації.");
}

private static void changeStringValue(String target, String newValue) throws Exception {
    Field valueField = String.class.getDeclaredField("value");
    valueField.setAccessible(true);

    Class<?> type = valueField.getType();

    if (type == char[].class) {
        char[] value = (char[]) valueField.get(target);
        char[] newChars = newValue.toCharArray();
        int len = Math.min(value.length, newChars.length);
        System.arraycopy(newChars, 0, value, 0, len);
        if (len < value.length) {
            Arrays.fill(value, len, value.length, '\0');
        }
    }
}
```

```

    } else if (type == byte[].class) {
        byte[] value = (byte[]) valueField.get(target);
        byte[] newBytes = newValue.getBytes(StandardCharsets.UTF_8);
        int len = Math.min(value.length, newBytes.length);
        System.arraycopy(newBytes, 0, value, 0, len);
        if (len < value.length) {
            Arrays.fill(value, len, value.length, (byte) 0);
        }
    } else {
        throw new IllegalStateException("Невідомий тип поля value: " + type);
    }
}
}
}

```

## Task2

### LogMenuApp.java

```

public class LogMenuApp {

    private static final Logger logger = Logger.getLogger(LogMenuApp.class.getName());
    private static final String BUNDLE_BASE = "location.messages";

    private static Locale currentLocale = new Locale("uk");
    private static ResourceBundle bundle = ResourceBundle.getBundle(BUNDLE_BASE, currentLocale);

    Run | Debug
    public static void main(String[] args) {
        configureLogging();
        logger.info("Application started");

        Scanner sc = new Scanner(System.in);

        while (true) {
            printMenu();

            System.out.print(bundle.getString("prompt.choice"));
            String choice = sc.nextLine().trim();

            switch (choice) {
                case "1":
                    logger.info("User chose: process file");
                    System.out.print(bundle.getString("prompt.enterFile"));
                    String path = sc.nextLine();
                    String best = findLineWithMostWords(path);
                    if (best != null) {
                        System.out.println(bundle.getString("result.bestLine"));
                        System.out.println(best);
                    }
                    break;
            }
        }
    }
}

```

```

        case "2":
            logger.info("User chose: change language");
            changeLanguage(sc);
            break;

        case "0":
            logger.info("User chose: exit");
            System.out.println("Bye!");
            return;

        default:
            logger.warning("Unknown menu choice: " + choice);
            System.out.println("???");
    }
}

private static void configureLogging() {
    logger.setUseParentHandlers(false);
    logger.setLevel(Level.ALL);

    try {
        ConsoleHandler console = new ConsoleHandler();
        console.setLevel(Level.INFO);

        FileHandler file = new FileHandler("lab10.log", true);
        file.setLevel(Level.WARNING);
        file.setFormatter(new SimpleFormatter());

        logger.addHandler(console);
        logger.addHandler(file);
    } catch (IOException e) {
        System.err.println("Cannot init logging: " + e.getMessage());
    }
}

```

```
private static void printMenu() {
    System.out.println("\n=== " + bundle.getString("menu.title") + " ===");
    System.out.println("1 - " + bundle.getString("menu.option.processFile"));
    System.out.println("2 - " + bundle.getString("menu.option.changeLang"));
    System.out.println("0 - " + bundle.getString("menu.option.exit"));
}
```

```
private static void changeLanguage(Scanner sc) {
    System.out.println(bundle.getString("lang.choose"));
    String choice = sc.nextLine().trim();

    switch (choice) {
        case "1":
            currentLocale = new Locale("en");
            bundle = ResourceBundle.getBundle(BUNDLE_BASE, currentLocale);
            logger.info("Language changed to EN");
            System.out.println(bundle.getString("lang.changed.en"));
            break;
        case "2":
            currentLocale = new Locale("uk");
            bundle = ResourceBundle.getBundle(BUNDLE_BASE, currentLocale);
            logger.info("Language changed to UK");
            System.out.println(bundle.getString("lang.changed.uk"));
            break;
        default:
            logger.warning("Unknown language choice: " + choice);
    }
}
```

```
private static String findLineWithMostWords(String path) {
    logger.fine("Opening file: " + path);
    String best = null;
    int maxWords = 0;
```

```

        try (BufferedReader br = new BufferedReader(new FileReader(path))) {
            String line;
            while ((line = br.readLine()) != null) {
                int cnt = countWords(line);
                if (cnt > maxWords) {
                    maxWords = cnt;
                    best = line;
                }
            }
            logger.info("File processed, maxWords = " + maxWords);
        } catch (IOException e) {
            logger.log(Level.SEVERE, "Error reading file: " + path, e);
            System.out.println(bundle.getString("error.file"));
        }

        return best;
    }

    private static int countWords(String line) {
        String trimmed = line.trim();
        if (trimmed.isEmpty()) return 0;
        return trimmed.split("\\s+").length;
    }
}

```

## messages\_en.properties

```

menu.title=Main menu
menu.option.processFile=Process file (line with the most words)
menu.option.changeLang=Change language
menu.option.exit=Exit

prompt.choice=Your choice:
prompt.enterFile=Enter file path:
lang.choose=Choose language: 1 - English, 2 - Ukrainian
lang.changed.en=Language changed to English
lang.changed.uk=Language changed to Ukrainian

result.bestLine=Line with the most words:
error.file=Error reading file

```

## messages\_uk.properties

```
menu.title=Головне меню
menu.option.processFile=Обробити файл (рядок з найбільшою кількістю слів)
menu.option.changeLang=Вибір мови
menu.option.exit=Вихід

prompt.choice=Ваш вибір:
prompt.enterFile=Введіть шлях до файлу:
lang.choose=Оберіть мову: 1 – Англійська, 2 – Українська
lang.changed.en=Мову змінено на англійську
lang.changed.uk=Мову змінено на українську

result.bestLine=Рядок з найбільшою кількістю слів:
error.file=Помилка читання файлу
```

<https://github.com/iamthegoose/java-fifth-semester/tree/lab10>

Висновок: В ході виконання даної лабораторної роботи я практично дослідив можливості рефлексії, освоїв створення та налаштування лог-журналів за допомогою Log4j, набув навички створення додатків з можливістю локалізації.