

ЛАБОРАТОРНА РОБОТА № 6

Обробка виключчих ситуацій. Потоки вводу-виводу. Робота з файлами.

Мета: обробка виключчих ситуація, створення власних класів винятків, робота з потоками вводу-виводу.

Хід роботи:

Завдання 1. Створити консольний Java проект java_lab_6 з пакетом com.education.ztu. Створіть в корені проекту папку directory_for_files. Всі файли з якими ви будете працювати при виконанні завдань повинні знаходитись в ній.

Завдання 2. Перевірка логіну та паролю:

- Створити статичний метод checkCredentials, який приймає на вхід три параметри: login, password і confirmPassword.
- Login повинен містити лише латинські літери, цифри та знак підкреслення. Довжина login має бути меншою за 20 символів. Якщо login не відповідає цим вимогам, необхідно викинути WrongLoginException.
- Password повинен містити лише латинські літери, цифри та знак підкреслення. Довжина password має бути менше 20 символів. Також password і confirmPassword повинні бути рівними. Якщо password не відповідає цим вимогам, необхідно викинути WrongPasswordException.
- WrongPasswordException і WrongLoginException - користувачькі класи виключення з двома конструкторами - один за замовчуванням, другий приймає повідомлення виключення і передає його в конструктор класу Exception.
- Обробка винятків проводиться усередині методу.
- Використовуємо multi-catch block.
- Метод повертає true, якщо значення є вірними або false в іншому випадку.

Завдання 3. Запис звіту про покупки в текстовий файл та читання з нього:

- Перевикористати код для формування звіту з покупок з лабораторної роботи 4.
- Після покупки, записати звіт у файл, який містить інформацію про вміст кошика.
- Використовуємо клас FileWriter або PrintWriter для запису звіту.

| | | | | | | | |
|-----------|--------------------|----------|--------|------|--|--|--|
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.00.000 – Пр6 | | |
| Розроб. | Iцук Ол.С. | | | | Звіт з лабораторної роботи | | |
| Перевір. | Піонтківський В.І. | | | | | | |
| Керівник | | | | | | | |
| Н. контр. | | | | | | | |
| Зав. каф. | | | | | ФІКТ Гр. ІПЗ-23-1 | | |

- Використовуємо FileReader для читання звіту та відображення в консолі. • Не використовувати try-with-resources.

Завдання 4. Копіювання файлу до іншого файлу: • Написати клас, який копіює вміст текстового файлу та картинки з одного файлу до іншого. • Використовуємо класи BufferedReader, FileReader, BufferedWriter, FileWriter, FileInputStream, FileOutputStream. • Використати try-with-resources.

Завдання 5. Робота з класом RandomAccessFile: • Дописати текст в декількох місцях в текстовому файлі. Можна використати текстовий файл зі списком товарів (наприклад, дописати декілька товарів) або будь-який інший файл з текстом.

Завдання 6. Робота з класом File: • Створити нову папку з ім'ям inner_directory. • Вивести абсолютний шлях створеної папки. • Вивести ім'я батьківської директорії. • Створити два текстових файли всередині папки inner_directory. • Один файл видалити. • Переіменувати папку inner_directory в renamed_inner_directory • Вивести список файлів та папок в папці directory_for_files, їх розмір та тип (файл, папка).

Завдання 7. Створення архіву: • Додати всі створені файли в папці directory_for_files до архіву. Використати клас ZipOutputStream. • Вивести список файлів з архіву. Використати клас ZipInputStream.

Лістинг програми:

Main.java:

```
package com.education.ztu;

import java.io.*;
import java.util.*;
import java.util.zip.*;

public class Main {

    public static void main(String[] args) throws IOException {

        String baseDir = "directory_for_files";

        // ===== Завдання 2: Перевірка логіну та паролю =====
        System.out.println("==== Check credentials ===");
        boolean valid = checkCredentials("User_123", "Pass_123", "Pass_123");
        System.out.println("Credentials valid: " + valid);

        // ===== Завдання 3: Звіт про покупки =====
        List<String> cart = Arrays.asList("Apple - 2 pcs", "Banana - 3 pcs", "Milk - 1 L");
        String reportFile = baseDir + "/purchase_report.txt";
    }
}
```

| | | | | | | |
|------|------|-------------------|--------|------|--|------|
| | | Iлуць Ол.С. | | | ДУ «Житомирська політехніка».25.121.00.000 – Пр6 | Арк. |
| | | Плюнгівський В.І. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 2 |

```

PurchaseReport.writeReport(reportFile, cart);
System.out.println("\nPurchase report written. Reading:");
PurchaseReport.readReport(reportFile);

// ===== Завдання 4: Копіювання файлів =====
String copyText = baseDir + "/report_copy.txt";
FileCopyUtil.copyTextFile(reportFile, copyText);

// ===== Завдання 5: RandomAccessFile =====
try (RandomAccessFile raf = new RandomAccessFile(reportFile, "rw")) {
    raf.seek(raf.length());
    raf.writeBytes("Eggs - 12 pcs\n");
} catch (IOException e) {
    e.printStackTrace();
}

// ===== Завдання 6: Робота з File =====
File innerDir = new File(baseDir + "/inner_directory");
innerDir.mkdir();
System.out.println("\nAbsolute path: " + innerDir.getAbsolutePath());
System.out.println("Parent: " + innerDir.getParent());

File file1 = new File(innerDir, "file1.txt");
File file2 = new File(innerDir, "file2.txt");
file1.createNewFile();
file2.createNewFile();

file1.delete();
innerDir.renameTo(new File(baseDir + "/renamed inner directory"));

System.out.println("\nFiles in directory_for_files:");
File[] files = new File(baseDir).listFiles();
if (files != null) {
    for (File f : files) {
        System.out.println(f.getName() + " - " + f.length() + " bytes - "
+ (f.isDirectory() ? "folder" : "file"));
    }
}

// ===== Завдання 7: Архівування =====
String zipFile = baseDir + "/archive.zip";
try (ZipOutputStream zos = new ZipOutputStream(new
FileOutputStream(zipFile))) {
    for (File f : new File(baseDir).listFiles()) {
        if (f.isFile()) {
            try (FileInputStream fis = new FileInputStream(f)) {
                zos.putNextEntry(new ZipEntry(f.getName()));
                byte[] buffer = new byte[1024];
                int len;
                while ((len = fis.read(buffer)) > 0) {
                    zos.write(buffer, 0, len);
                }
                zos.closeEntry();
            }
        }
    }
}

System.out.println("\nFiles in ZIP archive:");
try (ZipInputStream zis = new ZipInputStream(new
FileInputStream(zipFile))) {
    ZipEntry entry;
    while ((entry = zis.getNextEntry()) != null) {

```

| | | | | |
|------|------|-------------------|--------|------|
| | | Iлуц Ол.С. | | |
| | | Плюнгівський В.І. | | |
| Змн. | Арк. | № докум. | Підпис | Дата |

```

        System.out.println(entry.getName());
    }
}

// ===== Метод перевірки логіну та паролю =====
public static boolean checkCredentials(String login, String password, String confirmPassword) {
    try {
        if (!login.matches("[A-Za-z0-9_]+") || login.length() >= 20)
            throw new WrongLoginException("Login must contain only letters, digits, '_' and less than 20 chars.");
        if (!password.matches("[A-Za-z0-9_]+") || password.length() >= 20 || !password.equals(confirmPassword))
            throw new WrongPasswordException("Password invalid or does not match confirmPassword.");
        return true;
    } catch (WrongLoginException | WrongPasswordException e) {
        System.out.println("Error: " + e.getMessage());
        return false;
    }
}
}

```

WrongLoginException.java:

```

package com.education.ztu;

public class WrongLoginException extends Exception {
    public WrongLoginException() {
        super();
    }

    public WrongLoginException(String message) {
        super(message);
    }
}

```

WrongPasswordException.java:

```

package com.education.ztu;

public class WrongPasswordException extends Exception {
    public WrongPasswordException() {
        super();
    }

    public WrongPasswordException(String message) {
        super(message);
    }
}

```

FileCopyUtil.java:

```

package com.education.ztu;

import java.io.*;

public class FileCopyUtil {

    public static void copyTextFile(String source, String dest) {
        try (BufferedReader reader = new BufferedReader(new FileReader(source));
             BufferedWriter writer = new BufferedWriter(new FileWriter(dest))) {

            String line;
            while ((line = reader.readLine()) != null) {

```

| | | | | | |
|------|------|--------------------|--------|------|--|
| | | Iлук Ол.С. | | | Арк. |
| | | Плюнгієвський В.І. | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.00.000 – Пр6 |

```

        writer.write(line);
        writer.newLine();
    }

} catch (IOException e) {
    e.printStackTrace();
}
}

public static void copyBinaryFile(String source, String dest) {
    try (FileInputStream fis = new FileInputStream(source);
         FileOutputStream fos = new FileOutputStream(dest)) {

        byte[] buffer = new byte[1024];
        int bytesRead;
        while ((bytesRead = fis.read(buffer)) != -1) {
            fos.write(buffer, 0, bytesRead);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

PurchaseReport.java:

```

package com.education.ztu;

import java.io.*;
import java.util.List;

public class PurchaseReport {
    public static void writeReport(String filename, List<String> items) {
        try {
            FileWriter writer = new FileWriter(filename);
            for (String item : items) {
                writer.write(item + "\n");
            }
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void readReport(String filename) {
        try {
            FileReader reader = new FileReader(filename);
            int c;
            while ((c = reader.read()) != -1) {
                System.out.print((char)c);
            }
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Результат програми:

| | | | | | |
|------|------|--------------------|--------|------|--|
| | | Ilyuk Ol.C. | | | Арк. |
| | | Плюнгієвський В.І. | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | ДУ «Житомирська політехніка».25.121.00.000 – Пр6 |

```

"C:\Program Files\Java\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\
== Check credentials ==
Credentials valid: true

Purchase report written. Reading:
Apple - 2 pcs
Banana - 3 pcs
Milk - 1 L

Absolute path: C:\Users\User\Desktop\Java\Lab_6\directory_for_files\inner_directory
Parent: directory_for_files

Files in directory_for_files:
archive.zip - 498 bytes - file
inner_directory - 0 bytes - folder
purchase_report.txt - 54 bytes - file
renamed_inner_directory - 0 bytes - folder
report_copy.txt - 43 bytes - file

Files in ZIP archive:
archive.zip
purchase_report.txt
report_copy.txt

```

Рис 1. Результат програми

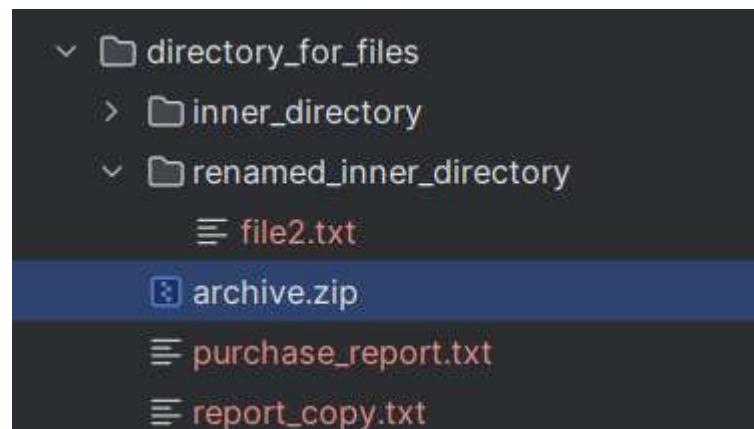
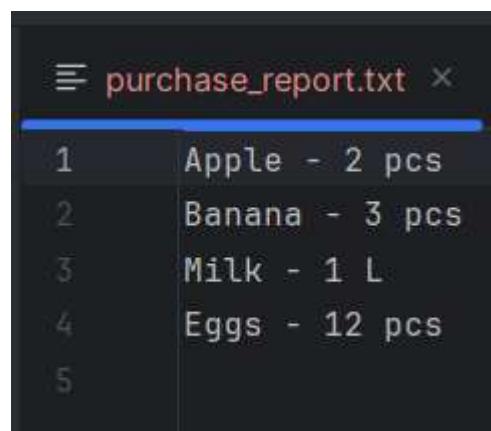


Рис 2. Створення файлів txt та zip файлу в папці directory_for_files



| | | | |
|-------------|-------------------|--|--------|
| Iuцук Ол.С. | Плюнгівський В.І. | ДУ «Житомирська політехніка».25.121.00.000 – Пр6 | Арк. |
| Змн. | Арк. | № докум. | Підпис |

Рис 3. Інформація в purchase_report.txt

```
≡ report_copy.txt ×  
1 Apple - 2 pcs  
2 Banana - 3 pcs  
3 Milk - 1 L  
4
```

A screenshot of a terminal window with a dark background and light-colored text. The title bar says "report_copy.txt". The content shows a list of four items, each consisting of a number and a product name followed by a quantity. The numbers are aligned to the left and have a vertical line to their right.

Рис 3. Інформація в report_copy.txt

Посилання на репозиторій: <https://github.com/Sasha1845/Java>

Висновок: Я навчився працювати з створенням власних класів та роботою вводу та виводу

| | | | | | | |
|------|------|-------------------|--------|------|--|------|
| | | Ilyuk Ol.C. | | | ДУ «Житомирська політехніка».25.121.00.000 – Пр6 | Арк. |
| | | Плюнгівський В.І. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | |