Міністерство науки та освіти України

Національний Університет «Львівська Політехніка»

Кафедра інформаційних систем та мереж

Лабораторна робота №3

з дисципліни «Екстремальне програмування»

Виконав:

Студент групи КН-311

Куренда Т.І.

Прийняв:

Щербак С.С.

Львів-2020

**Мета роботи:** навчитися вести логування з допомогою Log4j.

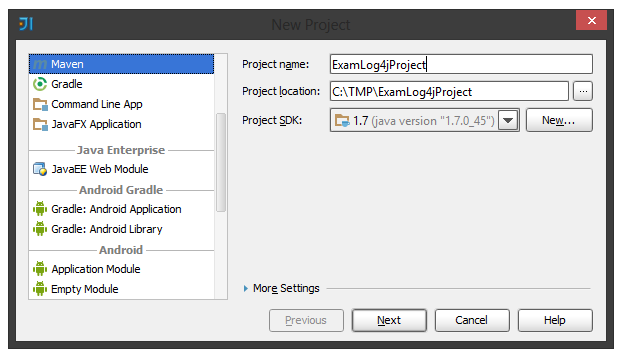
**Завдання**

Навчитися слідкувати за виконанням логіки програми, а саме використовуючи логування.

**Виконання завдання**

Логування – можливість слідкувати за процесом виконанням бізнес-логіки проекту.

**Крок 1. Створюємо проект та додаємо залежності.**



Тепер в **pom.xml** додаємо залежності:

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

Це всі залежності, які необхідно було підключити.

**Крок 2. Створення примітивної логіки для прикладу.**

Створюємо клас в якому була б бізнес-логіка; назвемо його **OrderLogic.**

package com.taras.logpack;

public class OrderLogic {

public void doOrder(){

// random logic

System.out.println("The order is placed!");

addToCart();

}

private void addToCart() {

// adding item to the cart

System.out.println("Item is added to the cart.");

}

}

Створюємо Main клас:

package com.taras.logpack;

public class Main {

private static OrderLogic logic;

public static void main(String[] args) {

logic = new OrderLogic();

logic.doOrder();

}

}

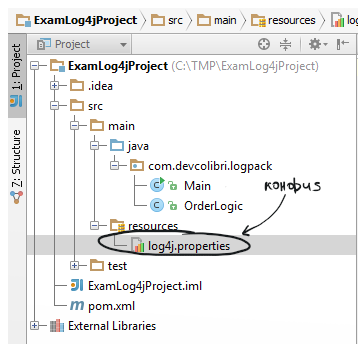
В результаті виконання даного коду, ми отримаємо наступне:

The order is placed!

Item is added to the cart.

**Крок 3. Конфігурація Log4j.**

Для того щоб гнучко керувати логуванням варто створити в **resources/** файл **log4j.**properties.



Тепер в цей файл додамо декілька стрічок конфігурації.

# Рівень логування

log4j.rootLogger=INFO, file

# Аппендер для роботи з файлами

log4j.appender.file=org.apache.log4j.RollingFileAppender

# Шлях де буде створюватись лог файл

log4j.appender.file.File=C:\\TMP\\log\_file.log

# Вказуємо максимальний розмір файлу з логами

log4j.appender.file.MaxFileSize=1MB

# Конфігуруємо шаблон виводу логів у файл

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n

**Крок 4. Додаємо примітивне логування.**

Тепер в клас OrderLogic добавимо логування і подивимось на результат.

package com.taras.logpack;

import org.apache.log4j.Logger;

public class OrderLogic {

// ініціалізація логера

private static final Logger log = Logger.getLogger(OrderLogic.class);

public void doOrder(){

// якась логіка

System.out.println("The order is placed!");

// логуємо інформацію

log.info("This is info message!");

addToCart();

}

private void addToCart() {

// додавання товара в корзину

System.out.println("The order is added to the cart");

// логуємо помилку

log.error("This is error message");

}

}

Тепер запустимо код знову. Ми отримаємо той самий результат, але на шляху C://TMP/ буде лежати файл **log\_file.log** з наступним вмістом:

2014-01-14 23:55:57 INFO OrderLogic:12 – This is info message!

2014-01-14 23:55:57 ERROR OrderLogic:19 – This is error message

**Висновок:** під час виконання лабораторної роботи я вивчив основи логування за допомогою Log4j.