**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗІНА**

**ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ НАУК**

**КАФЕДРА БЕЗПЕКИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ**

**Лабораторна робота № 4**

**з дисципліни навчальної дисципліни**

**«Математичні методи та технології тестування та верифікації програмного забезпечення»**

Виконала:

студентка групи КС-23

Рузудженк Сабіна

Перевірив:

Доцент

Нарєжній О.П

Харків – 2019

**Лабораторная работа № 4**

з навчальної дисципліни

«Математичні методи та технології тестування та верифікації програмного забезпечення»

**Тема роботи:** Фреймворк для логування log4j.

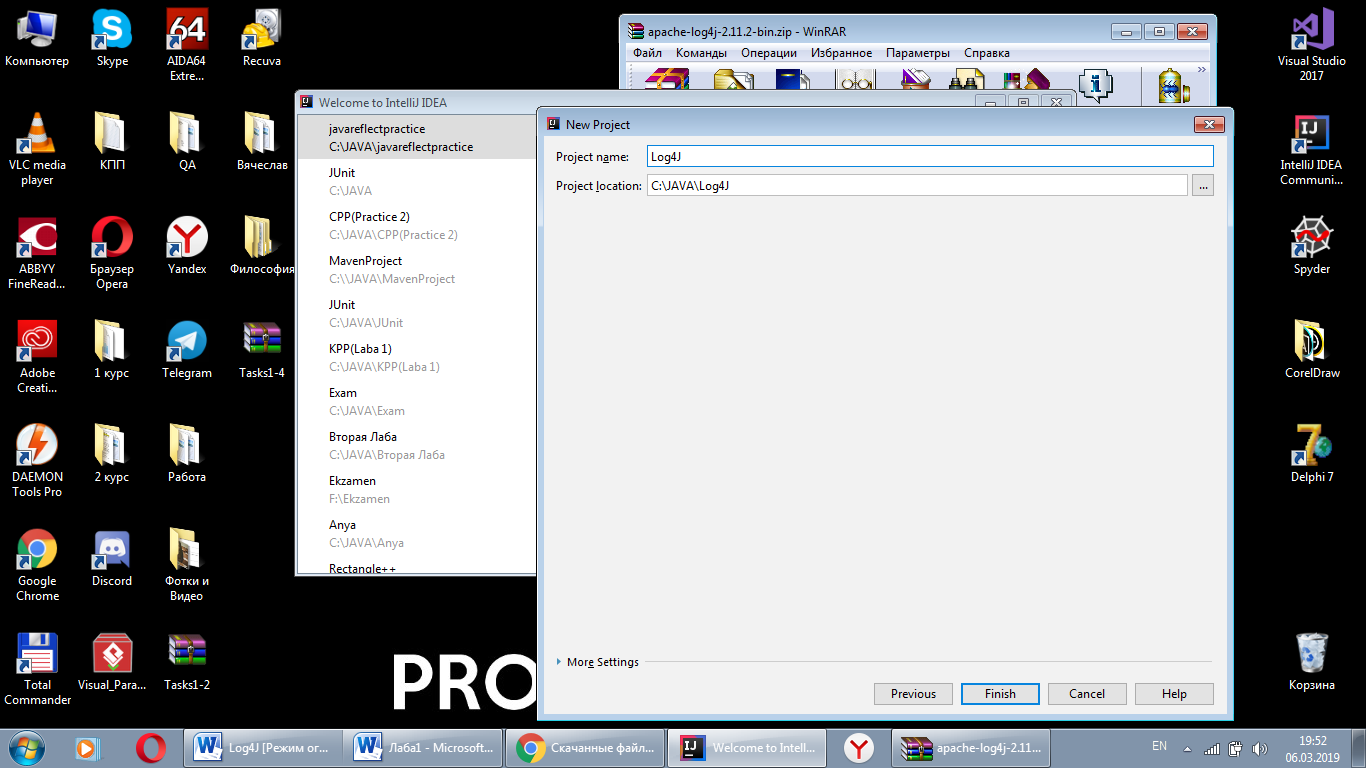
**Мета роботи:** Вивчення фреймворку запису логування log4j.

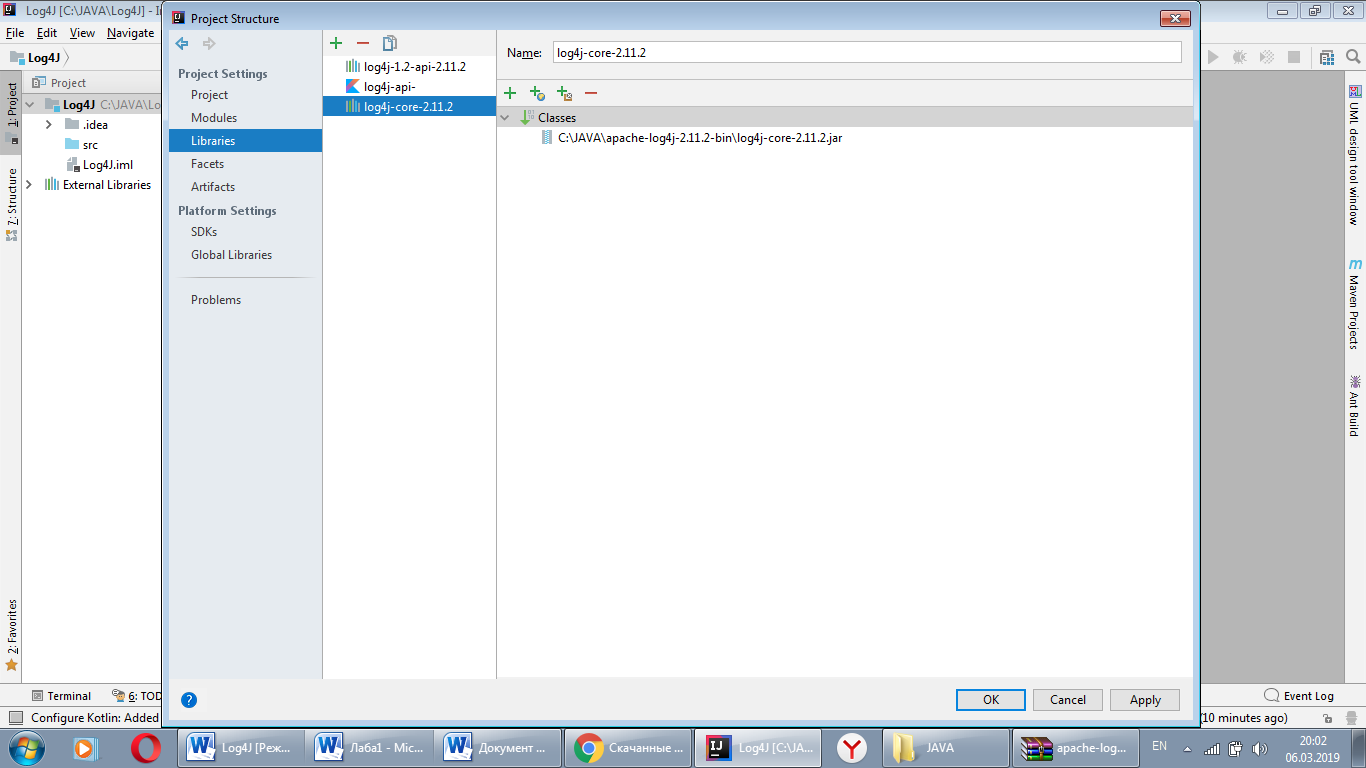
**Хід роботи**

Для початку роботи необхідно скачати Log4J на комп’ютер з електронного ресурсу: *https://logging.apache.org/log4j/2.x/download.html*. Після скачування необхідно розпакувати архів та перемістити папку до репозиторію.

Далі переходимо безпосередньо до роботи з Java-проектом.

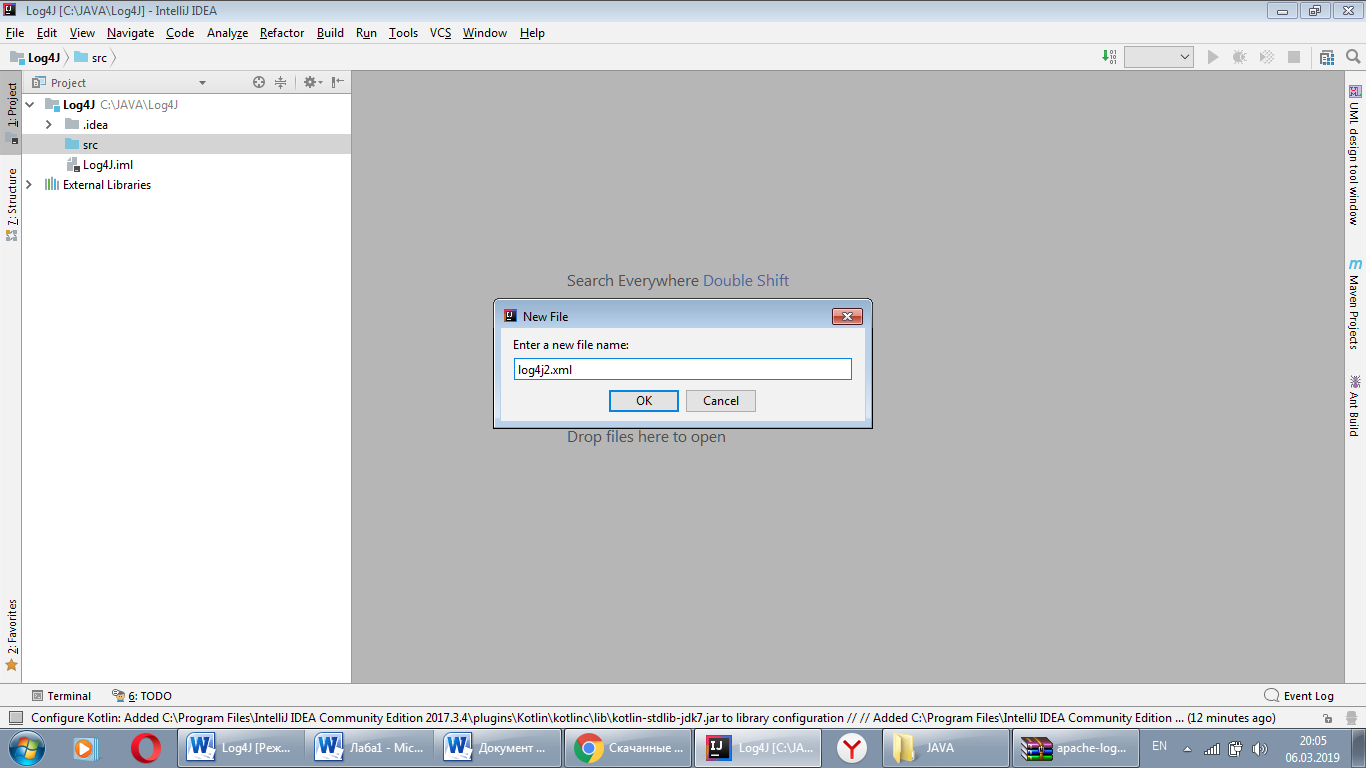
Створюємо новий проект «Log4J».



Для подальшої роботи до проекту підключаємо Log4J, як бібліотеку. Для цього необхідно послідовно виконати такі дії: *File – Project Structure – Libraries;* та додаємо нову бібліотеку до проекту.

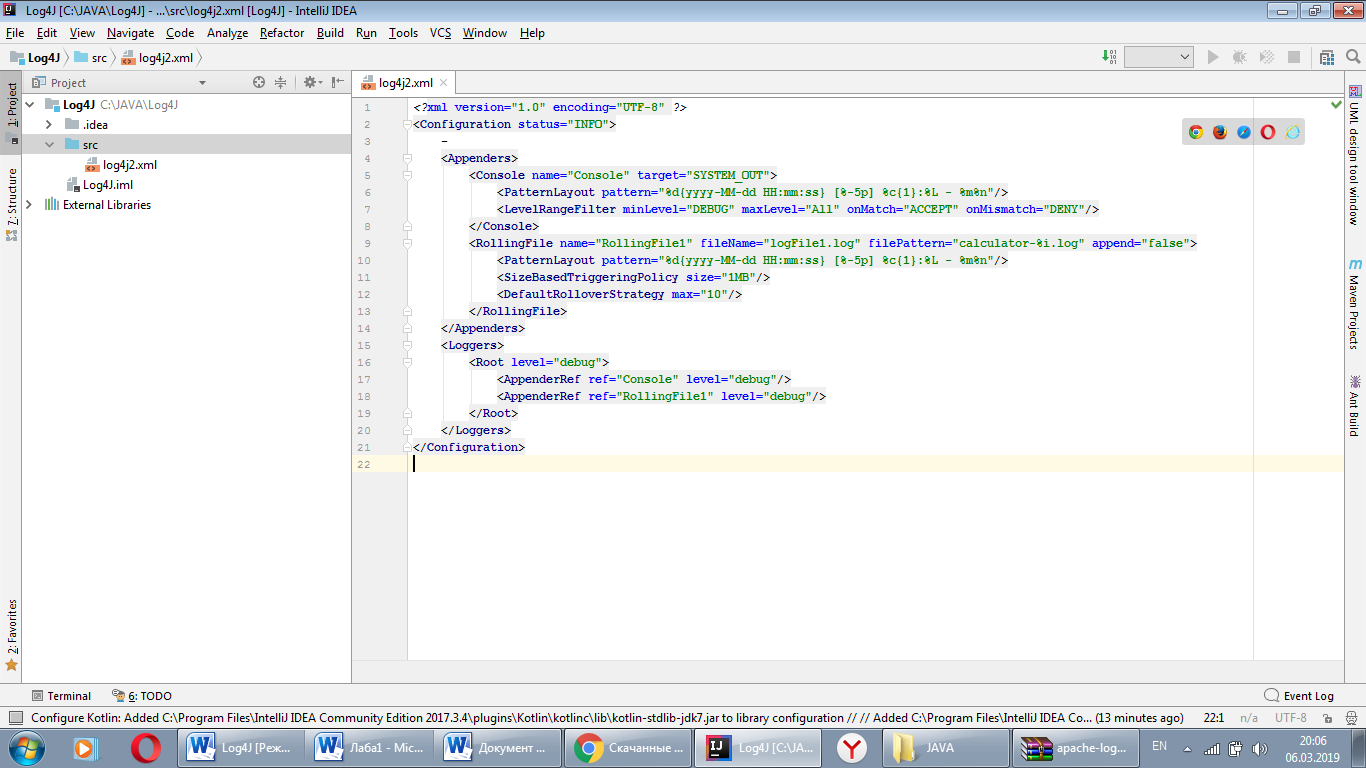
.

Створюємо файл *log4j2.xml*.



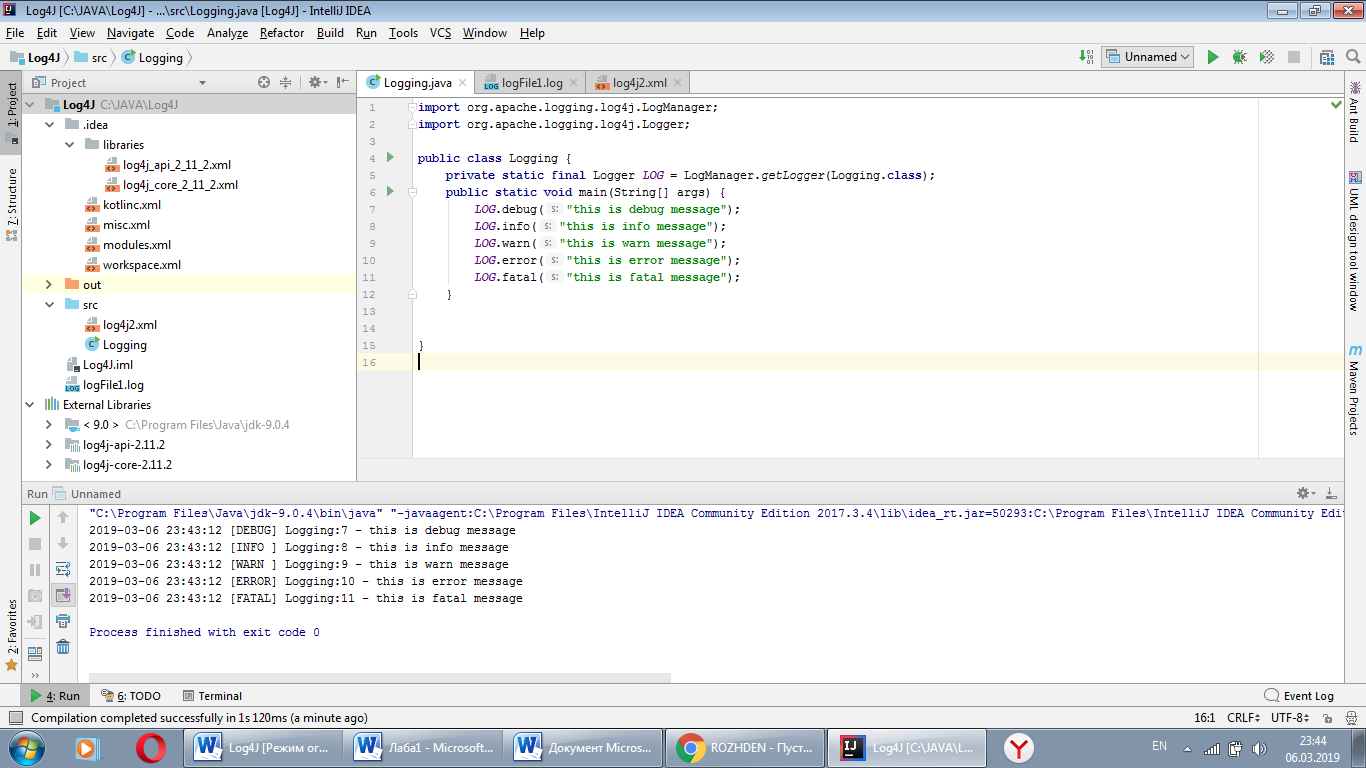
Потім заповнюємо його:

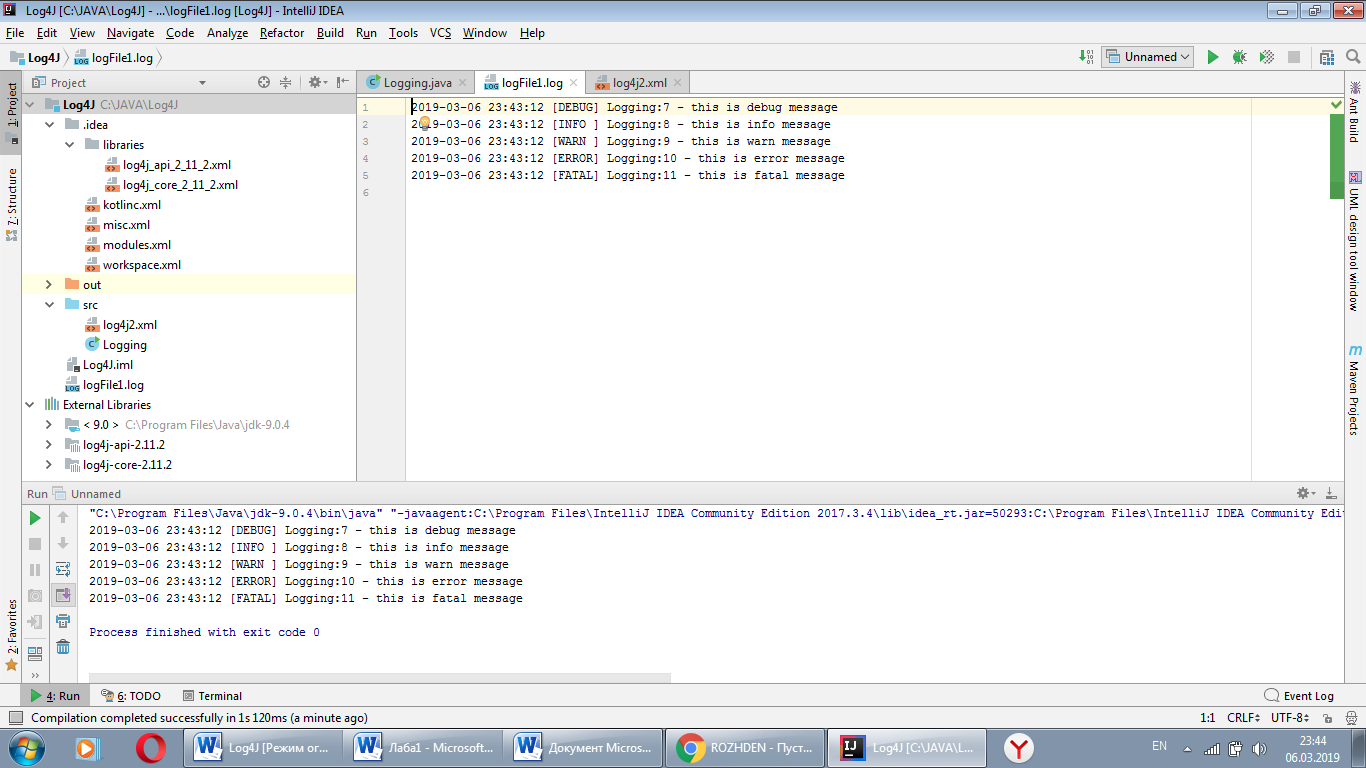
*<?***xml version="1.0" encoding="UTF-8"** *?>*<**Configuration status="INFO"**>  
 -  
 <**Appenders**>  
 <**Console name="Console" target="SYSTEM\_OUT"**>  
 <**PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%-5p] %c{1}:%L - %m%n"**/>  
 <**LevelRangeFilter minLevel="DEBUG" maxLevel="All" onMatch="ACCEPT" onMismatch="DENY"**/>  
 </**Console**>  
 <**RollingFile name="RollingFile1" fileName="logFile1.log" filePattern="calculator-%i.log" append="false"**>  
 <**PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss} [%-5p] %c{1}:%L - %m%n"**/>  
 <**SizeBasedTriggeringPolicy size="1MB"**/>  
 <**DefaultRolloverStrategy max="10"**/>  
 </**RollingFile**>  
 </**Appenders**>  
 <**Loggers**>  
 <**Root level="debug"**>  
 <**AppenderRef ref="Console" level="debug"**/>  
 <**AppenderRef ref="RollingFile1" level="debug"**/>  
 </**Root**>  
 </**Loggers**>  
</**Configuration**>.



Для подальшої роботи та перевірки коректності виконання програми створюємо новий Java-клас - *Logging* та заповнюємо його:

**import** org.apache.logging.log4j.LogManager;  
**import** org.apache.logging.log4j.Logger;  
  
**public class** Logging {  
 **private static final** Logger ***LOG*** = LogManager.*getLogger*(Logging.**class**);  
 **public static void** main(String[] args) {  
 ***LOG***.debug(**"this is debug message"**);  
 ***LOG***.info(**"this is info message"**);  
 ***LOG***.warn(**"this is warn message"**);  
 ***LOG***.error(**"this is error message"**);  
 ***LOG***.fatal(**"this is fatal message"**);  
 }  
   
}



Виконавши вищеописані дії, компілюємо програму та перевіряємо правильність даних, виведених на консоль та у файл.

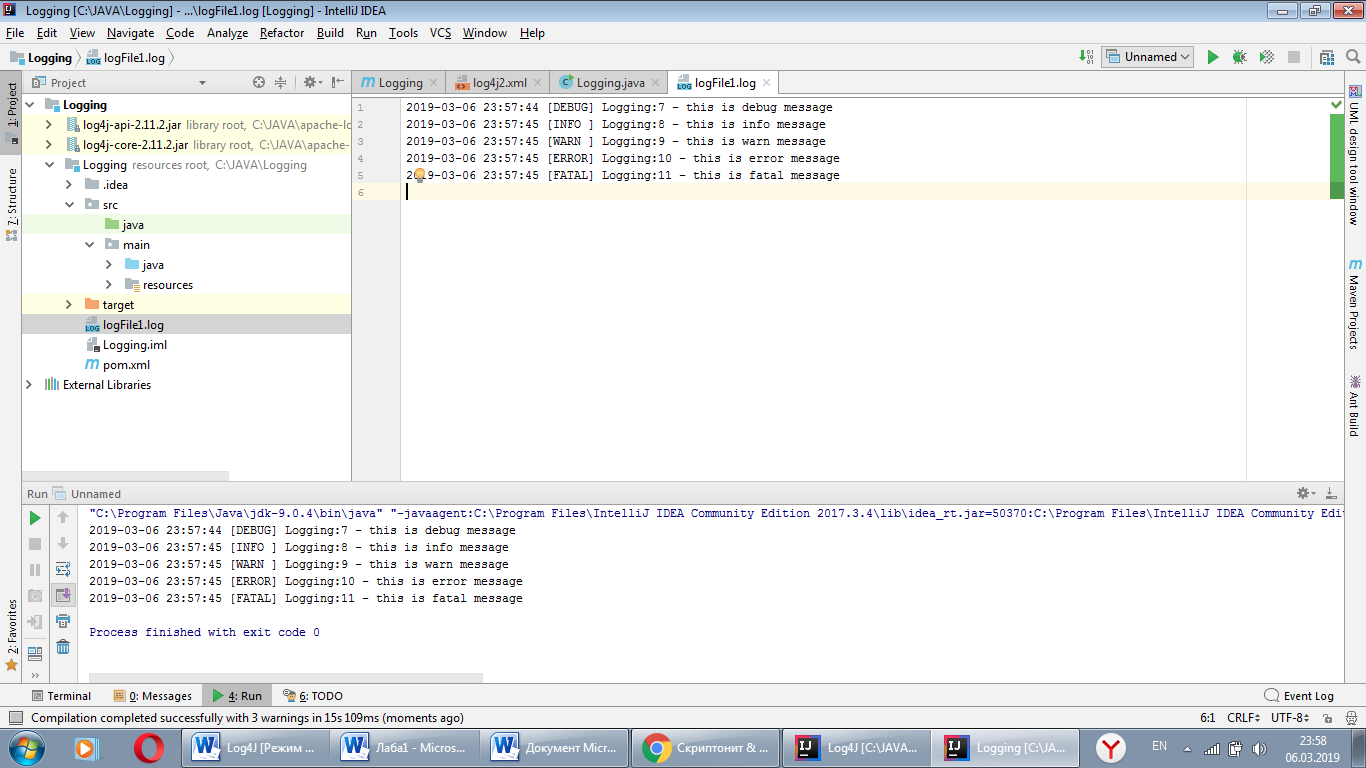
Далі проводимо аналогічні дії для Maven-проекту: створюємо новий проект, додаємо Log4J до списку ресурсів (resources).

До файлу *pom.xml* записуємо залежності:

<**dependencies**>  
 *<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->* <**dependency**>  
 <**groupId**>org.apache.logging.log4j</**groupId**>  
 <**artifactId**>log4j-core</**artifactId**>  
 <**version**>2.11.2</**version**>  
 </**dependency**>  
 *<!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api -->* <**dependency**>  
 <**groupId**>org.apache.logging.log4j</**groupId**>  
 <**artifactId**>log4j-api</**artifactId**>  
 <**version**>2.11.2</**version**>  
 </**dependency**>  
</**dependencies**>

Далі аналогічно створюємо Java-клас – *Logging* та копіюємо в нього зміст такого ж класу з попереднього проекту.

Запускаємо програму та перевіряємо коректність, виведених на консоль та у файл, даних:



**Висновки**

Отже, Log4j – фреймворк для приховування реалізації рутинних операцій логування деяких подій, які відбуваються під час роботи програми. Рутинні операції - хід виконання методів, класів, блоків обробки критичних ситуацій.

Під час логування користувач вирішує такі питання:

- вибір сховища (консоль, файл);

- конфігурація сховища - найменування сховища, обсяг сховища, шлях, збереження конфігурації;

- форматування записів журналу - дата, час, клас, метод і т.д.

Таким чином, у ході виконання лабораторної роботи було вивчено фреймворк запису легування – Log4J, а також створено два проекти, у яких реалізовані класи та файли,що дозволяють проводити процес легування.

У результаті виконання обох програм, було отримано коректні вихідні дані, як на консолі, так і у файлах.