

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Санкт-Петербургский государственный электротехнический университет  
“ЛЭТИ” им.В.И.Ульянова (Ленина) »

Кафедра МОЭВМ

**ОТЧЕТ**  
**по лабораторно-практической работе № 10**  
**по дисциплине «Объектно - ориентированное программирование**  
**на языке Java»**

Выполнил Баймухамедов Р. Р.

Преподаватель Павловский М.Г

Факультет КТИ

Группа № 3311

Подпись преподавателя \_\_\_\_\_

Санкт-Петербург  
2024 г

## Цель работы

Знакомство с методами протоколирования работы приложения с использования библиотеки Log4j.

## Описание задания

1. Перечень используемых типов сообщений, которые выводятся в логфайл.
2. Конфигурационный файл log4j.properties.
3. Исходные тексты классов, где осуществляется протоколирование работы приложения.
4. Текст документации, сгенерированный Javadoc
5. Фрагменты кода, содержащие WARN, INFO, ERROR и DEBUG

## Репозиторий

[https://github.com/brick1ng5654/course-2/tree/main/OOP/lab\\_10](https://github.com/brick1ng5654/course-2/tree/main/OOP/lab_10)

В этом репозитории находятся исходные файлы проекта:

В /src/edu/java/lab10 находится файл CinemaList.java

В /doc находится документация, сгенерированная JavaDoc

Ссылка на видеоотчёт

<https://youtu.be/Vg81QbQb8L4>

## Описание и проверки работоспособности приложения

При запуске программы начинается автоматическое протоколирование. Также добавлены логи при нажатии на кнопки некоторого функционала приложения (например, “Сохранить файл”, “Открыть файл” и т.д.).

Логи программы заносятся в файл как показано на примере 1

```
23:30:49,243 DEBUG main CinemaList:show:30 - App is started
23:31:03,442 DEBUG AWT-EventQueue-0 CinemaList:saveToFile:328 - The method of saving the file was called
23:31:10,847 DEBUG AWT-EventQueue-0 CinemaList:addMovie:224 - Method of adding the movie was called
23:31:17,285 DEBUG AWT-EventQueue-0 CinemaList:loadFromFile:368 - The method of loading from file was called
23:31:51,219 DEBUG main CinemaList:show:30 - App is started
23:32:01,836 DEBUG AWT-EventQueue-0 CinemaList:saveToFile:328 - The method of saving the file was called
```

Пример 1

## Конфигурационный файл log4j.properties

```
log4j.rootLogger=DEBUG, test
log4j.appender.test=org.apache.log4j.FileAppender
log4j.appender.test.file=myproject.log
log4j.appender.test.Encoding=Cp1251
log4j.appender.test.layout=org.apache.log4j.PatternLayout
log4j.appender.test.layout.conversionPattern=%d{ABSOLUTE} %5p %t %c{1}:%M:%L
- %m%n
```

### *Пояснение строк*

Строка (1) устанавливает уровень логирования для корневого логгера на DEBUG и связывает его с аппендером test

Строка (2) определяет, что аппендер test будет использовать класс org.apache.log4j.FileAppender, который записывает логи в файл

Строка (3) указывает имя файла, в который будут записываться логи. В данном случае файл называется myproject.log

Строка (4) указывает имя файла, в который будут записываться логи. В данном случае файл называется myproject.log

Строка (5) устанавливает кодировку для записи логов в файл. В данном случае используется кодировка Cp1251, которая является однобайтовой кодировкой для кириллицы.

Строка (6) указывает, что для форматирования логов будет использоваться класс org.apache.log4j.PatternLayout, который позволяет задавать шаблон для вывода логов.

Строка (7) задает шаблон для форматирования логов. Разберем его подробнее:

%d{ABSOLUTE}: Дата и время в абсолютном формате (например, HH:mm:ss,SSS).

%5p: Уровень логирования

%t: Имя потока, в котором было создано сообщение.

%c{1}: Имя категории (логгера), ограниченное одним символом.

%M: Имя метода, в котором было создано сообщение.

%L: Номер строки в исходном коде, где было создано сообщение.

%m: Сообщение логирования.

%n: Символ новой строки.

### **Вывод**

Во время выполнения данной лабораторной работы, были получены навыки работы с методами протоколирования приложения с использованием

библиотеки Log4j. В пункте «Описание и проверки работоспособности приложения» ошибок выявлено не было.

### Фрагменты кода, содержащие WARN, INFO, ERROR, DEBUG

```
private static final Logger log = Logger.getLogger(CinemaList.class);
public void show() {
    log.debug("App is started");
    ...
}
private void addMovie() {
    log.debug("Method of adding the movie was called");
    ...
}
private void addSession() {
    log.debug("Method of adding the session was called");
    ...
}
private void saveToFile() {
    log.debug("The method of saving the file was called");
    ...
}
```