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 ВТ

**ОТЧЕТ**

**по лабораторно-практической работе № 10**

**«Протоколирование работы приложения»**

**по дисциплине «Объектно - ориентированное программирование на языке Java»**

Выполнил Бондаренко Р.В.

Факультет КТИ

Группа № 1307

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2013 г

**Цель работы**

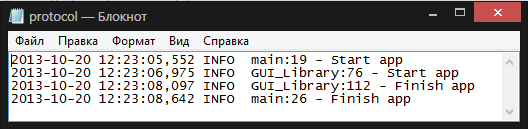
Знакомство c методами протоколирования работы приложения с использования библиотеки Log4j.

**Описание задания**

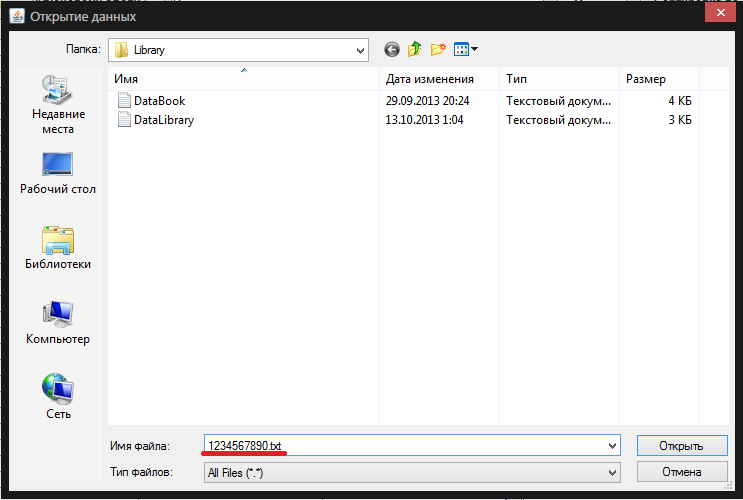
1. Перечень используемых типов сообщений, которые выводятся в логфайл.
2. Конфигурационный файл log4j.properties.
3. Исходные тексты классов, где осуществляется протоколирование работы приложения.
4. Текст документации, сгенерированный Javadoc
5. Фрагменты кода, содержащие WARN, INFO, ERROR и DEBUG

**Описание и проверки работоспособности приложения**

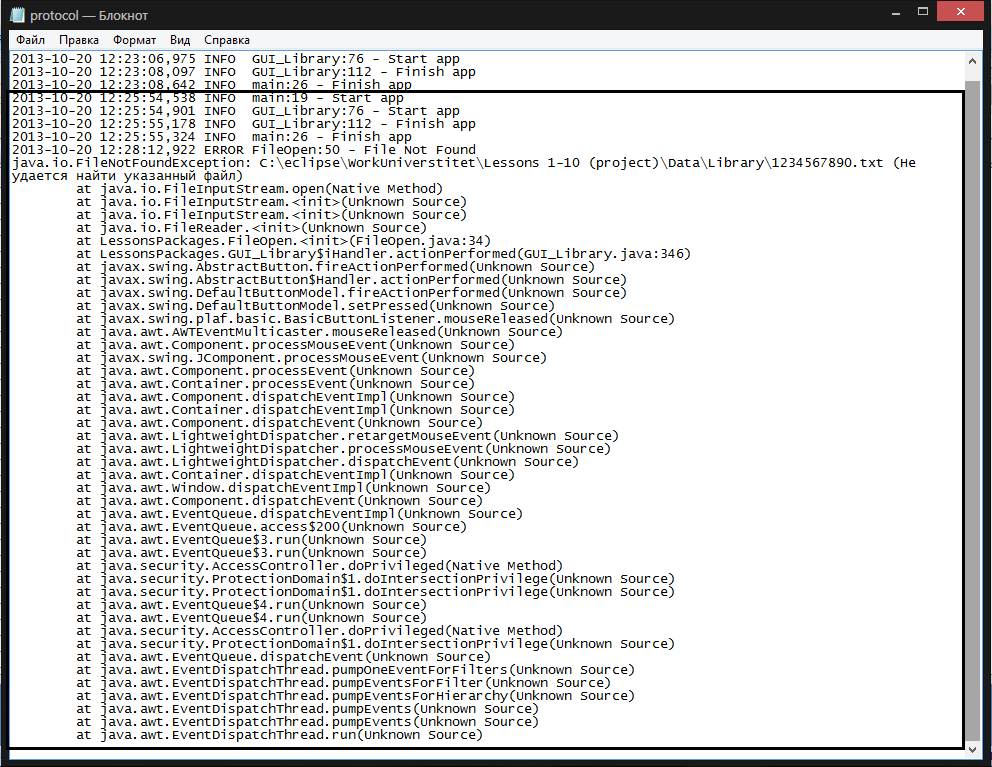
При запуске программы начинается автоматическое протоколирование. Сначала в файл записывается 2013-10-19 20:15:24,967 INFO main:19 - Start app (старт класса main, выводящий методом INFO в 19-ой строчке c приоритетом 967 ), затем 2013-10-19 20:15:25,357 INFO GUI\_Library:76 - Start app, после, если на этом пользователь заканчивает свои действия закрывая программу,2013-10-19 20:15:25,616 INFO GUI\_Library:112 - Finish app (указывающий на завершение) и далее 2013-10-19 20:15:25,822 INFO main:26 - Finish app.



Если попробовать открыть не существующий фай.



То в протоколе будет внесено следующие сообщение об ошибки, которое будет сгенерировано в *logFile*.error("File Not Found", e) класса FileOpen.



Обратите внимание на то, что при записи новых данных, старые данные остаются.

**Конфигурационный файл log4j.properties**

log4j.rootLogger=DEBUG, file

# Direct log messages to a log file

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

log4j.appender.file.File=./Data/Library/protocol.log

log4j.appender.file.MaxFileSize=1MB

log4j.appender.file.MaxBackupIndex=1

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

log4j.appender.file.layout.ConversionPattern=%d{ISO8601} %-5p %c**{1}**:%L - %m%n

Пояснение строк:

1. Описывает уровень сообщения DEBUG c символическим названием file, на которое будет выводиться сообщение.
2. Указывает, что сообщение будет выводиться в файл.
3. Задаёт имя и путь файлу для записи.
4. Определяет максимальный размер файла.
5. Максимальное резервное копирование.
6. Описывает формат вывода текста.
7. Вывод данных.

**Фрагменты кода, содержащие WARN, INFO, ERROR и DEBUG**

**public** **class** main {

**public** **static** **final** Logger *log* = Logger.*getLogger*(main.**class**);

**public** **static** **void** main(String[] args) {

*log*.info("Start app");

/\*

\*/

*log*.info("Finish app");

}

}

**public** **class** GUI\_Library **extends** JFrame{

**public** **static** **final** Logger *logGUI* = Logger.*getLogger*(GUI\_Library.**class**);

**public** GUI\_Library(){

*logGUI*.info("Start app");

/\*

\*

\*/

*logGUI*.info("Finish app");

}

**if**(e.getSource() == сomboBox){

//search\_field.setText("По " + (String) сomboBox.getSelectedItem());

*search\_field*.setText("Введите данные");

**if**(*search\_field*.getText().contains("Введите слово")){

*logGUI*.warn("search\_field != Введите данные");

*logGUI*.debug("Перезагрузите компьютер");

}

}

**public** **class** FileOpen **extends** JFrame{

/\*

\*

\*/

**public** **static** **final** Logger *logFile* = Logger.*getLogger*(FileOpen.**class**);

**try**{

/\*

\*

\*/

}**catch**(FileNotFoundException e) {

e.printStackTrace();

*logFile*.error("File Not Found", e);

}

}

**Вывод**

Во время выполнения данной лабораторной работы, были получены навыки работы с методами протоколирования приложения с использования библиотеки Log4j. В пункте «Описание и проверки работоспособности приложения» ошибок выявлено не было.