**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ВТ**

отчет

**по лабораторной работе №9**

**по дисциплине «Объектно-ориентированное программирование»**

Тема: **Протоколирование работы приложения**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 3315 |  | Красавцев Н.В. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

2024

**Цель работы.**

знакомство с методами протоколирования работы приложения с использованием библиотеки Log4j.

**Описание задания.**

Используя методы debug(), info(), warn(), error(), fatal() созданного объекта включить в описание класса операции вывода сообщений с соответствующим уровнем.

**Код класса.**

// Поток загрузки данных из XML

Thread load = new Thread(() -> {

logger.info("Запуск потока загрузки данных из XML...");

JOptionPane.showMessageDialog(null, "Загрузка данных из XML...");

loadFromXML("employee.xml", station); // Здесь замените путь на свой

logger.info("Данные успешно загружены из XML.");

JOptionPane.showMessageDialog(ServiceList, "Данные загружены");

try {

Thread.sleep(3000);

} catch (InterruptedException ex) {

logger.error("Ошибка при загрузке данных из XML: ");

ex.printStackTrace();

}

});

load.start();

// Поток добавления данных в XML

Thread addXML = new Thread(() -> {

if (load.isAlive()) {

try {

load.join();

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}

if (station.getRowCount() != 0) {

logger.info("Загрузка завершена, запуск потока добавления данных в XML...");

JOptionPane.showMessageDialog(null, "Редактирование XML...");

logger.debug("Начинаем добавление данных в XML...");

addEmployee();

logger.info("Данные успешно добавлены в XML.");

JOptionPane.showMessageDialog(null, "Редактирование XML завершено");

try {

Thread.sleep(3000);

} catch (InterruptedException ex) {

ex.printStackTrace();

logger.error("Ошибка в потоке добавления данных: ");

}

}

});

addXML.start();

// Поток создания HTML отчета

Thread reportHTML = new Thread(() -> {

if (addXML.isAlive()) {

try {

addXML.join();

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}

if (station.getRowCount() != 0) {

logger.info("Добавление данных завершено, запуск потока создания отчета HTML...");

JOptionPane.showMessageDialog(null, "Создание HTML отчёта...");

logger.debug("Начинаем создание HTML отчета...");

saveToHTML();

logger.info("HTML отчет успешно сгенерирован.");

JOptionPane.showMessageDialog(null, "Создание HTML отчёта завершено");

try {

Thread.sleep(1000);

} catch (InterruptedException ex) {

ex.printStackTrace();

logger.error("Ошибка в потоке создания отчета HTML: ");

}

}

});

}

**Сообщения лог-файлов.**

*2024-11-29 21:59:11 - INFO - org.example.ServiceList - Запуск потока загрузки данных из XML...*

*2024-11-29 21:59:13 - INFO - org.example.ServiceList - Данные успешно загружены из XML.*

*2024-11-29 21:59:19 - INFO - org.example.ServiceList - Загрузка завершена, запуск потока добавления данных в XML...*

*2024-11-29 21:59:22 - DEBUG - org.example.ServiceList - Начинаем добавление данных в XML...*

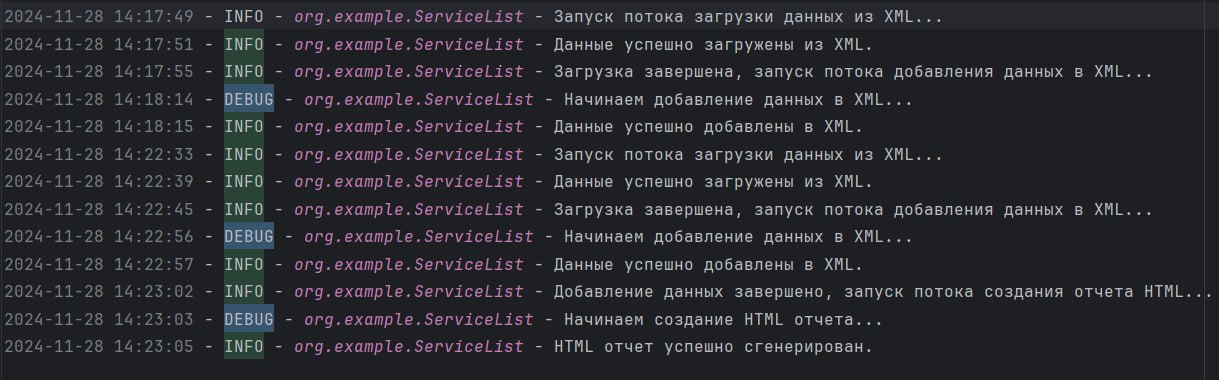
*2024-11-29 21:59:49 - INFO - org.example.ServiceList - Данные успешно добавлены в XML.*

*2024-11-29 21:59:53 - INFO - org.example.ServiceList - Добавление данных завершено, запуск потока создания отчета HTML...*

*2024-11-29 21:59:54 - DEBUG - org.example.ServiceList - Начинаем создание HTML отчета...*

*2024-11-29 21:59:55 - INFO - org.example.ServiceList - HTML отчет успешно сгенерирован.*

**Log-файл.**



Javadoc.

// Поток загрузки данных из XML

Thread load = new Thread(() -> {

logger.info("Запуск потока загрузки данных из XML...");

JOptionPane.showMessageDialog(null, "Загрузка данных из XML...");

loadFromXML("employee.xml", station); // Здесь замените путь на свой

logger.info("Данные успешно загружены из XML.");

JOptionPane.showMessageDialog(ServiceList, "Данные загружены");

try {

Thread.sleep(3000);

} catch (InterruptedException ex) {

logger.error("Ошибка при загрузке данных из XML: ");

ex.printStackTrace();

}

});

load.start();

// Поток добавления данных в XML

Thread addXML = new Thread(() -> {

if (load.isAlive()) {

try {

load.join();

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}

if (station.getRowCount() != 0) {

logger.info("Загрузка завершена, запуск потока добавления данных в XML...");

JOptionPane.showMessageDialog(null, "Редактирование XML...");

logger.debug("Начинаем добавление данных в XML...");

addEmployee();

logger.info("Данные успешно добавлены в XML.");

JOptionPane.showMessageDialog(null, "Редактирование XML завершено");

try {

Thread.sleep(3000);

} catch (InterruptedException ex) {

ex.printStackTrace();

logger.error("Ошибка в потоке добавления данных: ");

}

}

});

addXML.start();

// Поток создания HTML отчета

Thread reportHTML = new Thread(() -> {

if (addXML.isAlive()) {

try {

addXML.join();

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}

if (station.getRowCount() != 0) {

logger.info("Добавление данных завершено, запуск потока создания отчета HTML...");

JOptionPane.showMessageDialog(null, "Создание HTML отчёта...");

logger.debug("Начинаем создание HTML отчета...");

saveToHTML();

logger.info("HTML отчет успешно сгенерирован.");

JOptionPane.showMessageDialog(null, "Создание HTML отчёта завершено");

try {

Thread.sleep(1000);

} catch (InterruptedException ex) {

ex.printStackTrace();

logger.error("Ошибка в потоке создания отчета HTML: ");

}

}

});

}

**Вывод.**

При выполнении задания мы изучили протоколирование Java-приложений с использованием библиотеки log4j. Мы освоили различные методы этой библиотеки и настроили вывод сообщений с разными уровнями важности для нашего проекта курсовой работы.