**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ВТ**

отчет

**по лабораторной работе №10**

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

Тема: **Организация многопоточных приложений**

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

Санкт-Петербург

2024

**Цель работы.**

Знакомство с построением параллельных приложений в языке Java.

**Описание задания.**

Описать 3 параллельных потока, один из которых будет загружать данные из XML-файла, второй − редактировать данные и формировать XML-файл для отчета, а третий − строить отчет в HTMLформате. Второй поток не должен формировать XML-файл для отчета, пока первый не загрузит данные в экранную форму, а третий поток не должен формировать отчет, пока второй поток редактирует данные и записывает их в XML-файл.

**Код потоков.**

// 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 PDF 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: ");

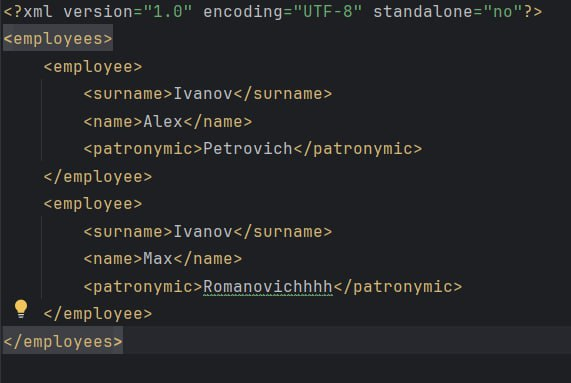
}

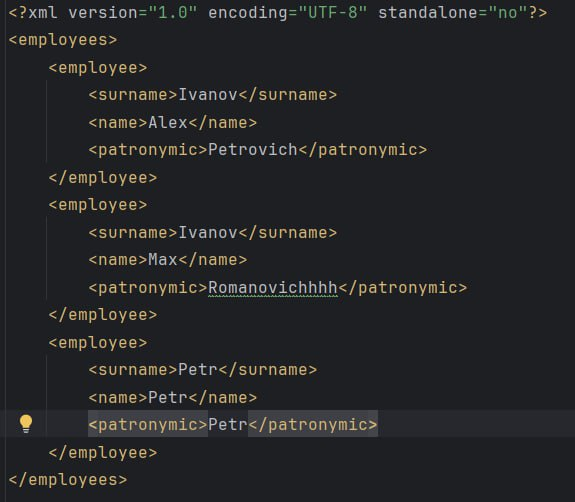
}

});

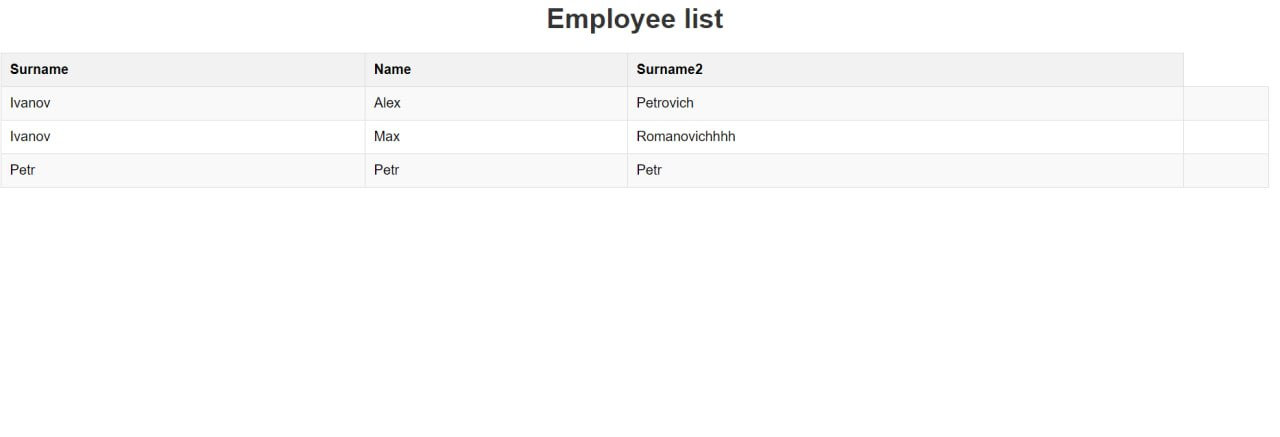
}

**XML.**





**HTML**



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 PDF 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.**

В процессе выполнения задания мы изучили концепцию многопоточности в Java-приложениях. Мы изучили различные методы работы с потоками (Thread) и создали три потока для выполнения операций загрузки данных из XML файла, их редактирования, сохранения и создания PDF отчета в рамках нашего проекта курсовой работы.