**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной Техники**

**ОТЧЕТ**

**по лабораторной работе № 8**

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

**Тема: Организация многопоточных приложений**

| Студентка гр. 3312 |  | Фирст М.В. |
| --- | --- | --- |
| Преподаватель |  | Павловский М.Г. |

Санкт-Петербург

2024

**Цель работы**

Знакомство с правилами и классами построения параллельных приложений в языке Java.

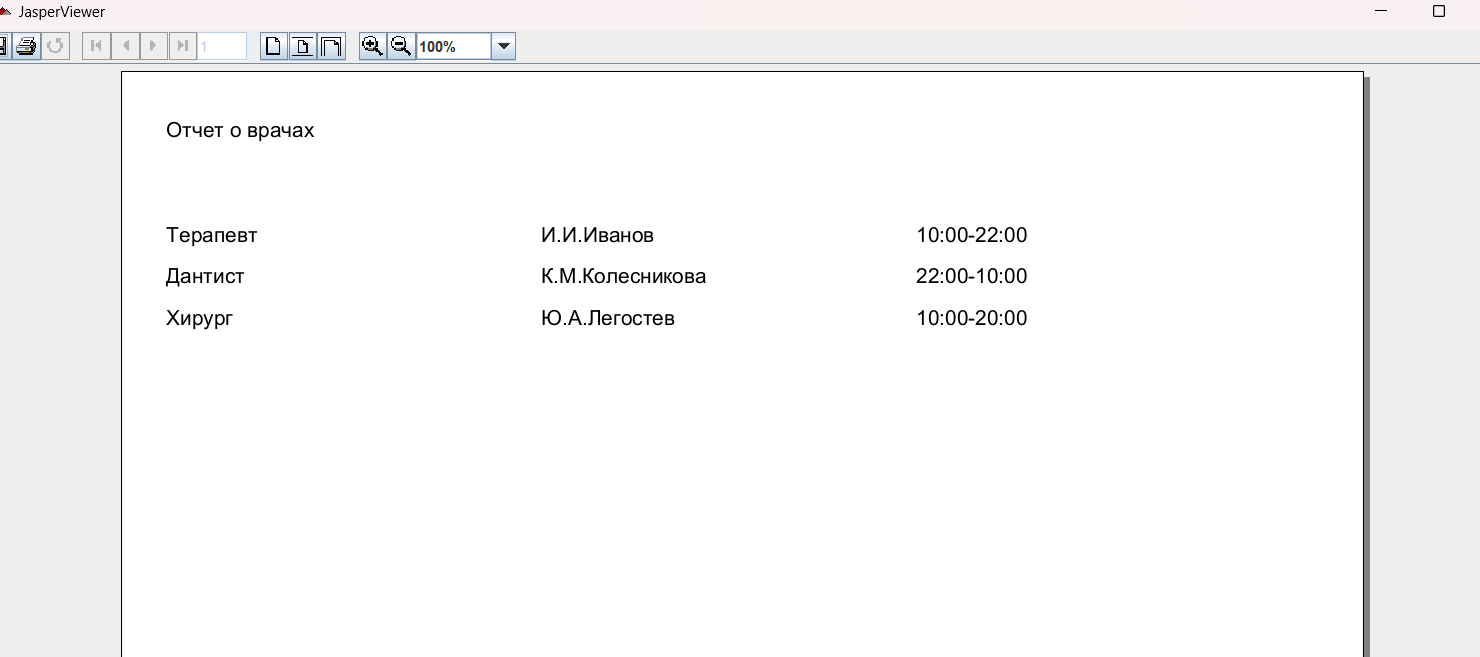
**Исходный XML файл**

****

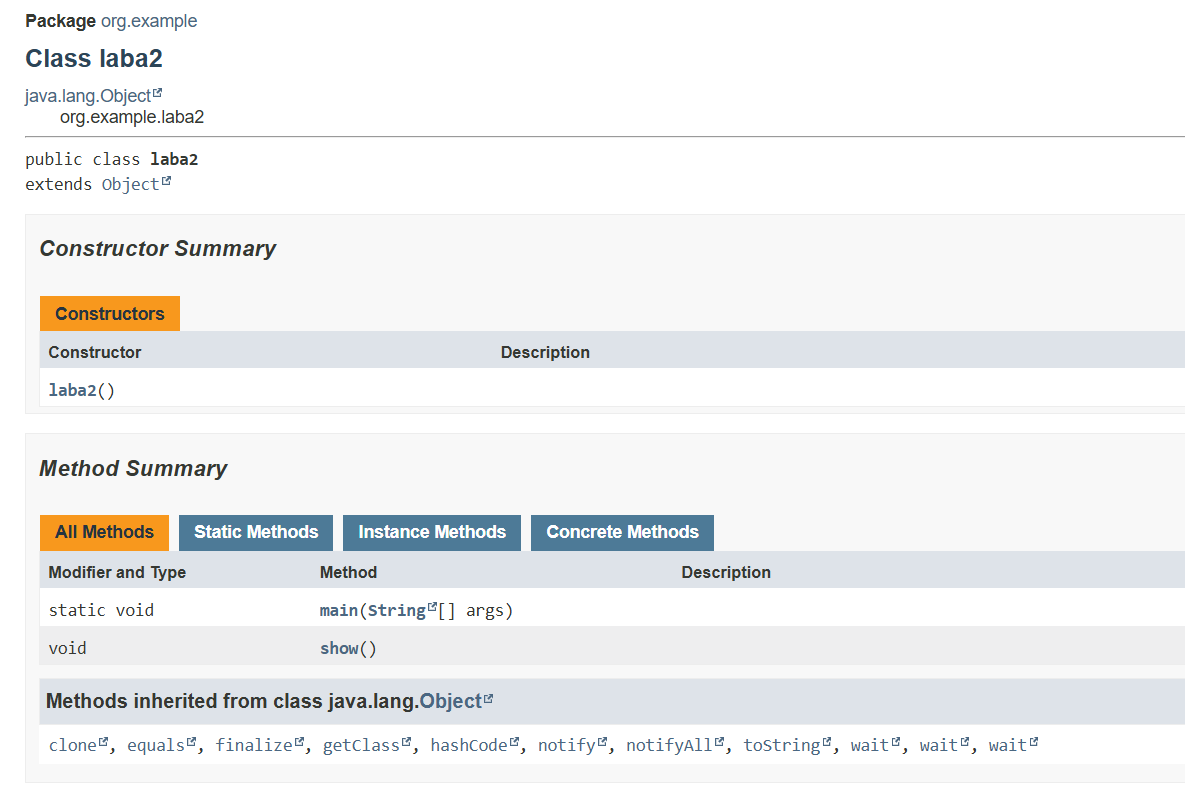
**Отредактированный XML файл**

****

**Сгенерированный отчет**

****

**Javadoc**

****

****

**Часть кода, отвечающая за многопоточность**

<...>

private CountDownLatch latch1 = new CountDownLatch(1);

private CountDownLatch latch2 = new CountDownLatch(1);

<...>

<...>

open\_file.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Запускаем первый поток для загрузки данных

new Thread(() -> {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Открыть файл");

int userSelection = fileChooser.showOpenDialog(doc\_list);

if (userSelection == JFileChooser.APPROVE\_OPTION) {

File fileToOpen = fileChooser.getSelectedFile();

readDataFromXML(fileToOpen);

}

latch1.countDown(); // Сигнализируем о завершении загрузки

}).start();

}

});

// New info button action listener

new\_info.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Запускаем второй поток для редактирования данных

new Thread(() -> {

try {

latch1.await(); // Ждем завершения первого потока

int selectedRow = docs.getSelectedRow();

if (selectedRow != -1) {

String[] data = new String[model.getColumnCount()];

for (int i = 0; i < model.getColumnCount(); i++) {

data[i] = model.getValueAt(selectedRow, i).toString();

}

editData(data);

} else {

showMessage("Выберите строку для редактирования");

}

latch2.countDown(); // Сигнализируем о завершении редактирования

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}).start();

}

});

generateReportButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

// Запускаем третий поток для генерации отчета

new Thread(() -> {

try {

latch2.await(); // Ждем завершения второго потока

generateReport();

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}).start();

}

});

<...>

**Ссылка на папку:** [3312\_First\_laba8](https://drive.google.com/drive/folders/1WlKZkCqTm0ti7p7nW_2PXxfcm_ijv-Ts?usp=sharing)