**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по лабораторной работе №9**

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

Тема: Многопоточность

|  |  |  |
| --- | --- | --- |
| Студент гр. 9308 |  | Яловега Н.В. |
| Преподаватель |  | Гречухин М.Н. |

Санкт-Петербург

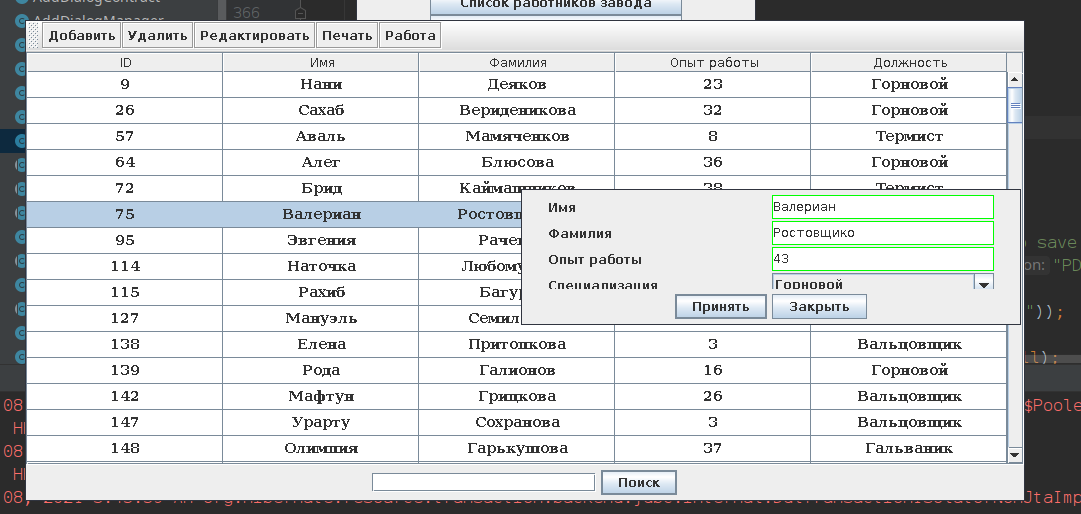
2021

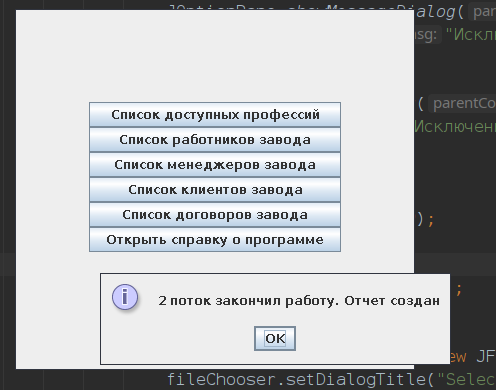
# Цель работы

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

# Параллельное выполнение и синхронизация

Теперь после синхронизации методов доступа к полям классов (где это необходимо), можно одновременно редактировать записи и просматривать их.

 Также генерация отчетов pdf создается в отдельном потоке. После выполения операции будет выведена информация о завершении потока.



t2 = new Thread(() -> {  
 try {  
 JFileChooser fileChooser = new JFileChooser();  
 fileChooser.setDialogTitle("Select where you want to save the report");  
 FileFilter pdf = new FileNameExtensionFilter("PDF file(.pdf)", "pdf");  
 fileChooser.addChoosableFileFilter(pdf);  
 fileChooser.setCurrentDirectory(new File("."));  
 String resultpath = "reportContracts.pdf";  
 int returnVal = fileChooser.showSaveDialog(null);  
 if(returnVal == JFileChooser.*APPROVE\_OPTION*)  
 {  
 File file = fileChooser.getSelectedFile();  
 resultpath = file.getAbsolutePath();  
 }  
 checkList();  
 makeXml();  
 ReportUtil.*print*("dataContracts.xml", "window/dataContracts", "contracts.jrxml", resultpath);  
 JOptionPane.*showMessageDialog*(null, "2 поток закончил работу. Отчет создан");  
 } catch (Exception ex) {  
 JOptionPane.*showMessageDialog*(null, "Ошибка: " + ex.toString());  
 *log*.log(Level.*SEVERE*, "Исключение: ", ex);  
 }  
});  
t2.start();

# Выводы

Во время выполнения данной лабораторной работы мы ознакомились с правилами построения параллельных приложений в языке Java.