

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: Протоколирование работы приложения

Студент гр. 9308

Яловега Н.В.

Преподаватель

Гречухин М.Н.

Санкт-Петербург

2021

Цель работы

Знакомство с методами протоколирования работы приложения.

Протоколирование

Для протоколирования были добавлены логгеры почти во все классы с gui. Пример:

```
add.addActionListener((e) ->
{
    log.info("Старт Add listener");
    addDialogWorker = new AddDialogWorker(window,
WorkerWindow.this, "Добавление записи");
    addDialogWorker.setVisible(true);
});

add.setMnemonic(KeyEvent.VK_A);
delete.addActionListener((e) -> {
    log.info("Старт Delete listener");
    if (dataWorkers.getRowCount() > 0) {
        if (dataWorkers.getSelectedRow() != -1) {
            try {

employeeService.delete(Integer.parseInt(dataWorkers.getVa
lueAt(dataWorkers.getSelectedRow(), 0).toString()));

model.removeRow(dataWorkers.convertRowIndexToModel(dataWo
rkers.getSelectedRow()));
                JOptionPane.showMessageDialog(window, "Вы
удалили строку");
                log.info("Была удалена строка данных");
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(null,
"Ошибка");
                log.log(Level.SEVERE, "Исключение: ",
ex);
            }
        } else {
            JOptionPane.showMessageDialog(null, "Вы не
выбрали строку для удаления");
            log.log(Level.WARNING, "Исключение: не
выбрана строка для удаление");
        }
    } else {
        JOptionPane.showMessageDialog(null, "В данном
окне нет записей. Нечего удалять");
        log.log(Level.WARNING, "Исключение: нет записей.
нечего удалять");
    }
});
```

```

delete.setMnemonic(KeyEvent.VK_D);

edit.addActionListener((e)-> {
    log.info("Старт Edit listener");
    if (model.getRowCount() != 0) {
        if (dataWorkers.getSelectedRow() != -1) {
            t1 = new Thread(() -> {
                JOptionPane.showMessageDialog(null, "1
поток запущен");
                editDialogWorker = new
EditDialogWorker(window, WorkerWindow.this,
"Редактирование");
                editDialogWorker.setVisible(true);
            });
            t1.start();
        } else {
            JOptionPane.showMessageDialog(null, "Не
выбрана строка. Нечего редактировать");
            log.log(Level.WARNING, "Исключение: не
выбрана строка для удаление");
        }
    } else {
        JOptionPane.showMessageDialog(null, "В данном
окне нет записей. Нечего редактировать");
        log.log(Level.WARNING, "Исключение: нет записей.
нечего удалять");
    }
});
edit.setMnemonic(KeyEvent.VK_E);

print.addActionListener((e)->{
    log.info("Старт Print listener");
    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 = "reportWorkers.pdf";
            int returnVal =
fileChooser.showSaveDialog(null);
            if(returnVal == JFileChooser.APPROVE_OPTION)
            {
                File file =

```

```

fileChooser.getSelectedFile();
        resultpath = file.getAbsolutePath();
    }
    checkList();
    makeXml();
    ReportUtil.print("dataWorkers.xml",
"window/dataWorkers", "workers.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();
});

```

Пример протоколирования

```
июн. 11, 2021 4:38:24 PM App main
INFO: Запуск программы
июн. 11, 2021 4:38:33 PM Factory.gui.ManagerWindow show
INFO: Открытие окна ManagerWindow
июн. 11, 2021 4:38:33 PM Factory.gui.ManagerWindow show
INFO: Добавление кнопок к окну ManagerWindow
июн. 11, 2021 4:38:33 PM Factory.gui.ManagerWindow show
INFO: Добавление таблицы с данными к окну ManagerWindow
июн. 11, 2021 4:38:33 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 5.4.24.Final
июн. 11, 2021 4:38:34 PM
org.hibernate.annotations.common.reflection.java.JavaReflectionManage
r <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
июн. 11, 2021 4:38:35 PM
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectio
nProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for
production use!)
июн. 11, 2021 4:38:35 PM
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectio
nProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL
[jdbc:mysql://localhost/kurs]
июн. 11, 2021 4:38:35 PM
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectio
nProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
июн. 11, 2021 4:38:35 PM
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectio
nProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
июн. 11, 2021 4:38:35 PM
org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectio
nProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
июн. 11, 2021 4:38:36 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
июн. 11, 2021 4:38:38 PM
org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransacti
onIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess
[org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$Conn
ectionProviderJdbcConnectionAccess@712a42bd] for (non-JTA) DDL
execution was not in auto-commit mode; the Connection 'local
transaction' will be committed and the Connection will be set into
auto-commit mode.
июн. 11, 2021 4:38:41 PM Factory.gui.ManagerWindow lambda$show$1
INFO: Старт Add listener
июн. 11, 2021 4:38:44 PM Factory.gui.ManagerWindow lambda$show$4
INFO: Старт Edit listener
июн. 11, 2021 4:38:47 PM Factory.gui.ManagerWindow lambda$show$6
INFO: Старт Print listener
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by
net.sf.jasperreports.engine.util.ClassUtils
(file:/home/musicfreakt/.m2/repository/net/sf/jasperreports/jasperrep
```

```
orts/6.17.0/jasperreports-6.17.0.jar) to constructor  
com.sun.org.apache.xerces.internal.util.XMLGrammarPoolImpl()  
WARNING: Please consider reporting this to the maintainers of  
net.sf.jasperreports.engine.util.ClassUtils  
WARNING: Use --illegal-access=warn to enable warnings of further  
illegal reflective access operations  
WARNING: All illegal access operations will be denied in a future  
release
```

Выводы

При выполнении лабораторной работы была изучена технология протоколирования приложений java.