

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Новоуральский технологический институт –

филиал федерального государственного автономного образовательного учреждения
высшего образования «Национальный исследовательский ядерный университет «МИФИ»

(НТИ НИЯУ МИФИ)

Колледж НТИ

Цикловая методическая комиссия информационных технологий

ОТЧЕТ №11

ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ НА ТЕМУ

«ПРОГРАММНАЯ РЕАЛИЗАЦИЯ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ»

ПМ.05 «Разработка программного обеспечения компьютерных сетей»

МДК.05.01 «Защита информации в КС»

Специальность СПО 09.02.03

«Программирование в компьютерных системах»

очная форма обучения

на базе основного общего образования

Выполнил

студент группы КПр–47 Д

Егорушкин И.А.

26.01.2021

дата

подпись

Проверил

преподаватель

Горницкая И.И.

дата

подпись

Новоуральск 2020

Цель работы: Анализ рисков информационной безопасности

Оборудование:

AMD Ryzen 5 3550U

ОЗУ 8 Гб

Программное обеспечение:

Windows 10 Professional 64 бит;

Ход работы:

Задание

Вариант		Задание	Входные данные процедуры	Выходные данные процедуры	Дополнительные условия
1	Алгоритм работы антивирусной программы-ревизора	Ревизоры запоминают исходное состояние файлов/каталогов, тогда, когда компьютер еще не заражен вирусом, а затем периодически сравнивают текущее состояние файла/каталога с исходным. Если обнаружены изменения, то на экран дисплея выводятся сообщения. Разработать процедуру поиска заданных (см. доп. усл.) изменений в файле/каталоге.	Имя файла (файлов)/ Имя каталога (каталогов)	Сообщение о наличии/отсутствии изменений	Поиск изменений в дате и времени создания файла
2					Поиск изменений в атрибутах и размере файла
3					Поиск изменений в содержании файла
4					Поиск изменений в содержании каталога

Алгоритм работы программы

Пользователь вводит файл или каталог после чего все файлы заносятся в хранилище, которое сверяет каждые 10 секунд файлы.

В случаи повреждения или измены файла будет предложено переместить в карантин.

Поиск ищет совпадения в хранилище файлов.

Листинг программы

`package org.example;`

`import java.io.File;`

`import java.io.IOException;`

`import java.util.*;`

`import javafx.application.Platform;`

`import javafx.collections.FXCollections;`

`import javafx.collections.ObservableList;`

`import javafx.fxml.FXML;`

`import javafx.scene.control.Button;`

```
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.VBox;
import javafx.stage.DirectoryChooser;
import javafx.stage.FileChooser;
import org.apache.commons.io.FileUtils;
```

```
public class PrimaryController {
```

```
    @FXML
```

```
    public Button fileInput;
```

```
    @FXML
```

```
    public TextField inputFile;
```

```
    public CustomTableView<Table> table;
```

```
    public static Set<SaveFile> saveFiles = new
```

```
TreeSet<>(Comparator.comparing(SaveFile::getUrl));
```

```
    public VBox root;
```

```
    public Button packageInput;
```

```
    private static final int tikConst = 10;
```

```
    public static int tik = tikConst;
```

```
    public static Thread thread;
```

```
    public Label chekTime;
```

```
    @FXML
```

```
    private void initialize() throws IOException {
```

```
        table = new CustomTableView<Table>();
```

```
        TableColumn<Table, String> colName = new TableColumn<Table, String>("Имя");
```

```
        TableColumn<Table, String> colUrl = new TableColumn<Table, String>("Url");
```

```
        TableColumn<Table, String> colStat = new TableColumn<Table, String>("Статус");
```

```
        TableColumn<Table, String> tableStringTableColumn = crateTable("Дата  
создания", "timeCrate", 100);
```

```
TableColumn<Table, String> tableStringSize = crateTable("Дата создания", "size",  
100);
```

```
TableColumn<Table, String> colDataEdit = new TableColumn<Table, String>("Дата  
изменения");
```

```
colName.setCellValueFactory(new PropertyValueFactory<>("name"));  
colUrl.setCellValueFactory(new PropertyValueFactory<>("url"));  
colStat.setCellValueFactory(new PropertyValueFactory<>("circle"));  
colDataEdit.setCellValueFactory(new PropertyValueFactory<>("time"));
```

```
colStat.setMinWidth(100);  
colUrl.setMinWidth(100);  
colDataEdit.setMinWidth(100);  
colStat.setMaxWidth(100);
```

```
colName.setStyle(" -fx-alignment: CENTER");  
colUrl.setStyle(" -fx-alignment: CENTER");  
colStat.setStyle(" -fx-alignment: CENTER");  
colDataEdit.setStyle(" -fx-alignment: CENTER");
```

```
colName.setEditable(false);  
colUrl.setEditable(false);  
colStat.setEditable(false);  
colDataEdit.setEditable(false);
```

```
table.getColumns().addAll(colName, colUrl, tableStringTableColumn, colDataEdit,  
tableStringSize, colStat);
```

```
root.getChildren().add(root.getChildren().size() - 2, table);  
try {  
    thread.stop();  
} catch (Exception e) {
```

```

    }

    thread = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                Platform.runLater(new Runnable() {
                    @Override
                    public void run() {
                        chekTime.setText("Проверка через:" + tik);
                    }
                });
                if (tik <= 1) {
                    check();
                    tik = tikConst;
                }
                try {
                    tikAdd();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    });

    thread.setDaemon(true);
    thread.start();
    check();
}

private void tikAdd() throws InterruptedException {
    tik--;
    Thread.sleep(1000);
}

```

```

private TableColumn<Table, String> crateTable(String s, String name, int w) {
    TableColumn<Table, String> col = new TableColumn<Table, String>(s);
    col.setCellValueFactory(new PropertyValueFactory<>(name));
    col.setStyle("-fx-alignment: CENTER");
    col.setEditable(false);
    return col;
}

```

@FXML

```

public void failGet(MouseEvent mouseEvent) {

    FileChooser fileChooser = new FileChooser();

    File file = fileChooser.showOpenDialog(App.getStage());
    if (file != null) {
        inputFile.setText(file.getPath());
    }
}

```

```

public void packageGet(MouseEvent mouseEvent) {
    DirectoryChooser directoryChooser = new DirectoryChooser();
    File file = directoryChooser.showDialog(App.getStage());
    if (file != null) {
        inputFile.setText(file.getPath());
    }
}

```

@FXML

```

public void addFile(MouseEvent mouseEvent) throws IOException {
    new Thread(new Runnable() {
        @Override
        public void run() {
            File file = new File(inputFile.getText());
            if (!file.isDirectory()) {
                try {

```

```

        saveFiles.add(new SaveFile(file, file.getPath(),
FileUtils.readFileToByteArray(file)));
    } catch (IOException e) {
        e.printStackTrace();
    }
    ObservableList<Table> list = FXCollections.observableArrayList();
    saveFiles.forEach(saveFile -> list.add(new
Table(saveFile.getName().getName(), saveFile.getName())));

    table.setItems(list);
} else {
    processFilesFromFolder(file).forEach(file1 -> {
        try {
            saveFiles.add(new SaveFile(file1, file1.getPath(),
FileUtils.readFileToByteArray(file1)));
        } catch (IOException e) {
            e.printStackTrace();
        }
        ObservableList<Table> list = FXCollections.observableArrayList();
        saveFiles.forEach(saveFile -> list.add(new
Table(saveFile.getName().getName(), saveFile.getName())));

        table.setItems(list);
    });
}
}).start();
}

```

```

public Set<File> processFilesFromFolder(File folder) {
    Set<File> files = new HashSet<>();
    File[] folderEntries = folder.listFiles();
    for (File entry : folderEntries) {
        if (entry.isDirectory()) {

```

```

        processFilesFromFolder(entry);
        continue;
    }
    files.add(entry);
}
return files;
}

```

```

public void check(MouseEvent mouseEvent) {
    check();
}

```

```

private void check() {
    ObservableList<Table> list = FXCollections.observableArrayList();
    saveFiles.forEach(
        saveFile -> {
            try {
                saveFile.setChek(Arrays.equals(FileUtils.readFileToByteArray(new
File(saveFile.getUrl()), saveFile.bytes));
                Table table = new Table(saveFile.name.getName(), saveFile.getName());
                if (!saveFile.isChek()) {
                    table.setCircle("He корректен");
                } else {
                    table.setCircle("Ok");
                }
                list.add(table);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    );
    Platform.runLater(new Runnable() {
        @Override
        public void run() {

```



```

        table.setItems(list);
    }
});

}

public void search(MouseEvent mouseEvent) throws IOException {
    App.setRoot("serch");
}
}

```

Пример работы программы

Добавить файл, папку

Выбрать файл
Выбрать папку

Добавить

Проверка через:6

Имя	Url	Дата соз...	Дата изменения	Дата соз...	Статус
1 — коп...	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 12:42:25 YEKT	16.0 kb	Ok
1.xlsm	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 12:42:25 YEKT	16.0 kb	Ok
1.xlsx	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.14 12:24:35 YEKT	6.0 kb	Ok
2 — коп...	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 11:19:16 YEKT	21.0 kb	Ok
2.xlsm	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 11:19:16 YEKT	21.0 kb	Ok
5 — коп...	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.18 10:10:11 YEKT	16.0 kb	Ok
5.xlsm	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.18 10:10:11 YEKT	16.0 kb	Ok
6 — коп...	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 14:21:17 YEKT	14.0 kb	Ok
6.xlsm	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 14:21:17 YEKT	14.0 kb	Ok
7 — коп...	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 14:29:54 YEKT	12.0 kb	Ok
7.xlsm	C:\Users\48562\OneDrive\...	2021.01.2...	2021.01.11 14:29:54 YEKT	12.0 kb	Ok

Проверить

Поиск

Поиск

11.01.2021

по дате созданию файла

0

↑ bytes ↓

по размеру файла

поиск по названию или каталогу

Имя	Url	Дата соз...	Дата изменения	Дата соз...	Статус
desktop.ini	C:\Users\48562\OneDrive\...	2021.01.1...	2021.01.11 11:36:45 YEKT	418.0 bytes	Ok
Записная...	C:\Users\48562\OneDrive\...	2021.01.1...	2020.04.13 20:28:45 YEKT	119.0 bytes	Ok

вернуться

Вывод : Было изучено сравнение файлов и их поиск.