183 Kleinprojekt Dokumentation - Florian Naef

Deklaration der FXML-Elemente

```
@FXML
private StackPane root;

@FXML
private Rectangle dropTarget;

@FXML
private TextField pathInput;

@FXML
private TextField passwordInput;

@FXML
private Label passwordStrength;

@FXML
private Button go;

@FXML
private Label pathOutput;
```

Es werden verschiedene FXML-Elemente deklariert, die im Code verwendet werden. Ein StackPane mit dem Namen "root", ein Rechteck mit dem Namen "dropTarget", zwei Textfelder mit den Namen "pathInput" und "passwordInput", ein Label mit dem Namen "passwordStrength", ein Button mit dem Namen "go" und ein weiteres Label mit dem Namen "pathOutput".

initialize-Methode

```
public void initialize() {
    // Set up drag and drop handling for the Rectangle
    dropTarget.setOnDragOver(event -> {
        if (event.getDragboard().hasFiles()) {
            event.acceptTransferModes(TransferMode.COPY);
        }
        event.consume();
    });

    dropTarget.setOnDragDropped(event -> {
        Dragboard db = event.getDragboard();
        boolean success = false;
        if (db.hasFiles()) {
            File file = db.getFiles().get(0);
            pathInput.setText(file.getAbsolutePath());
            success = true;
        }
        event.setDropCompleted(success);
        event.consume();
    });
}
```

Diese Methode wird automatisch aufgerufen, wenn der Controller initialisiert wird. Sie richtet die Drag-and-Drop Funktion für das "dropTarget" ein. Wenn der Benutzer eine Datei in das Rechteck zieht, wird der Pfad der Datei in das Feld "pathInput" eingefügt in welches der Nutzer den Pfad auch von Hand eintragen kann.

confirm-Methode

```
@FXML
private void confirm() throws IOException {
    path = pathInput.getText();
    passwordInput.setEditable(true);
}
```

Diese Methode wird aufgerufen, wenn der Benutzer seine Pfadeingabe bestätigt. Sie speichert den Inhalt des Textfelds "pathInput" in der Variable "path" und ermöglicht die Bearbeitung des Passwortfelds "passwordInput".

encryptFileU-Methode

```
public void encryptFileU() throws Exception {

    String password = passwordInput.getText();

    boolean isStrongPassword = password.matches(passwordPattern);

    passwordStrength.setText(isStrongPassword ? "Password ist stark genug." : "Password ist nicht stark genug.");

    if (isStrongPassword) {
        SecureRandom random = new SecureRandom();
        byte[] salt = new byte[16];
        random.nextBytes(salt);

        int iterations = 10000;
        int keyLength = 128;
        KeySpec spec = new PBEKeySpec(passwordInput.getText().toCharArray(), salt, iterations, keyLength);
        SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
        byte[] keyBytes = factory.generateSecret(spec).getEncoded();
        secretKey = new SecretKeySpec(keyBytes, "AES");

        byte[] fileContent = Files.readAllBytes(Paths.get(path));

        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);

        byte[] encryptedContent = cipher.doFinal(fileContent);

        FileOutputStream outputStream = new FileOutputStream(path);
        outputStream.write(salt);
        outputStream.write(encryptedContent);
        outputStream.close();

        pathOutput.setText("Verschlüsselte Datei hier gespeichert: " + path);
    }
}
```

Diese Methode wird aufgerufen, wenn der Benutzer den Verschlüsseln Button drückt. Hier wird die Verschlüsselung mit dem AES-Algorithmus durchgeführt. Zunächst wird das eingegebene Passwort aus dem Textfeld "passwordInput" abgerufen.

Das Passwort wird mit einem regulären Ausdruck überprüft, um sicherzustellen, dass es den Anforderungen an ein starkes Passwort entspricht. Wenn das Passwort stark genug ist, wird ein zufälliges Salt generiert und mit dem Passwort kombiniert.

Mit Hilfe des Salt und des Passworts wird ein SecretKey-Objekt generiert. Anschließend wird der Inhalt der Datei, die in "path" angegebenen ist, gelesen und in ein Byte-Array geladen.

Ein Cipher-Objekt wird erstellt und im Verschlüsselungsmodus mit dem generierten SecretKey initialisiert und in den Verschlüsselungsmodus gesetzt.

Der Inhalt der Datei wird mit dem Cipher verschlüsselt und das verschlüsselte Ergebnis wird in ein anderen Byte-Array geschrieben.

Eine FileOutputStream wird erstellt, um die verschlüsselte Datei zu speichern. Zuerst wird das Salt in die Datei geschrieben, gefolgt vom verschlüsselten Inhalt.

Zum Ende wird das Label "pathOutput" aktualisiert, um den Pfad der verschlüsselten Datei anzuzeigen.

decryptFileU-Methode

```
public void decryptFileU() throws Exception {
    try {
        byte[] encryptedContent = Files.readAllBytes(Paths.get(path));

        byte[] salt = Arrays.copyOfRange(encryptedContent, 0, 16);

        int iterations = 10000;
        int keyLength = 128;
        KeySpec spec = new PBEKeySpec(passwordInput.getText().toCharArray(), salt, iterations, keyLength);
        SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
        byte[] keyBytes = factory.generateSecret(spec).getEncoded();
        secretKey = new SecretKeySpec(keyBytes, "AES");

        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);

        byte[] decryptedContent = cipher.doFinal(encryptedContent, 16, encryptedContent.length - 16);

        FileOutputStream outputStream = new FileOutputStream(path);
        outputStream.write(decryptedContent);
        outputStream.close();

        passwordStrength.setText("Passwort richtig");
        pathOutput.setText("Entschlüsselte Datei hier gespeichert: " + path);
    }catch(Exception e){
        passwordStrength.setText("Passwort falsch");
    }

}
```

Diese Methode wird aufgerufen, wenn der Benutzer den Button zum Entschlüsseln drückt. Hier wird die Entschlüsselung wieder mit dem AES-Algorithmus durchgeführt. Zuerst wird der Inhalt der verschlüsselten Datei gelesen und in ein Byte-Array geladen.

Dann wird das Salt aus dem Byte-Array extrahiert.

Mit dem eingegebenen Passwort und dem Salt wird ein SecretKey-Objekt generiert.

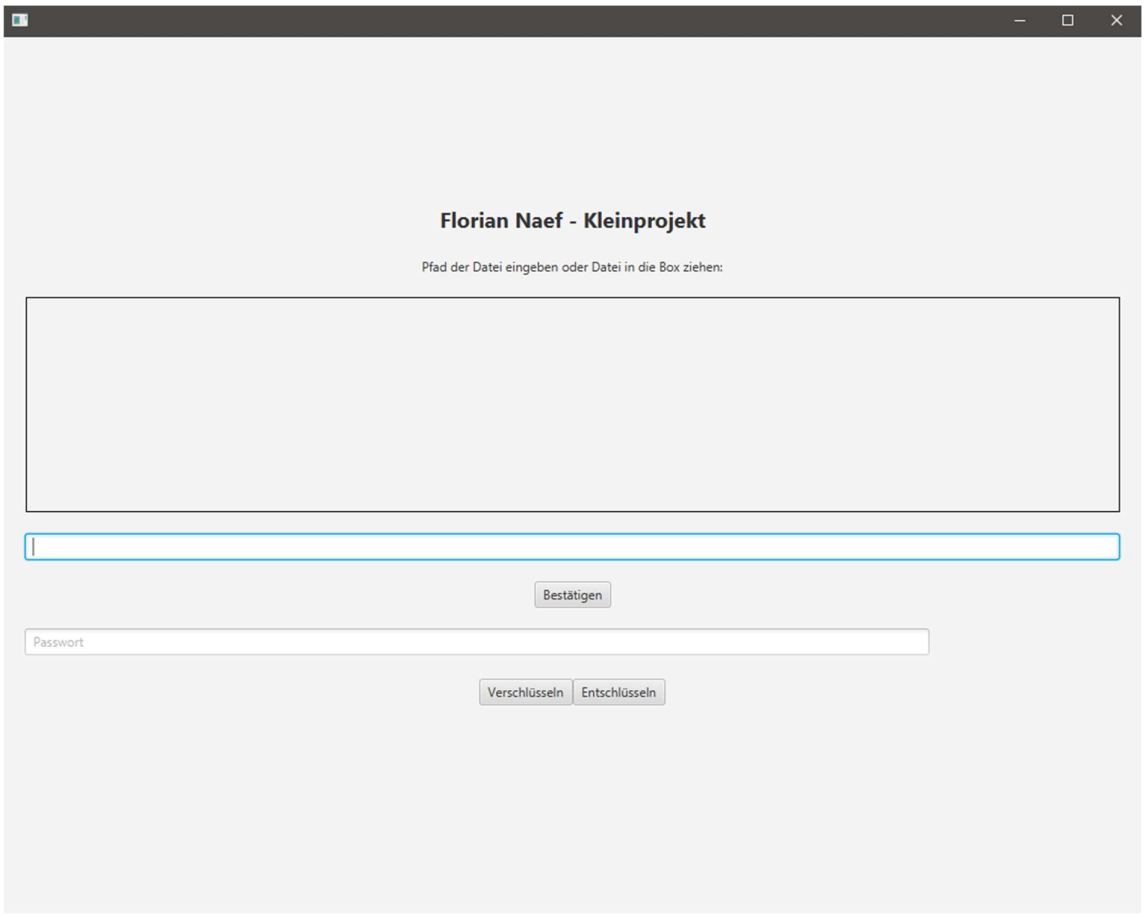
Das Cipher-Objekt wird erstellt und im Entschlüsselungsmodus mit dem SecretKey initialisiert.

Der verschlüsselte Inhalt wird mit dem Cipher entschlüsselt und das entschlüsselte Ergebnis wird in ein Byte-Array geschrieben.

Eine FileOutputStream wird erstellt, um die entschlüsselte Datei zu speichern. Der Inhalt des Byte-Arrays wird in die Datei geschrieben.

Die FileOutputStream wird geschlossen und das Label "pathOutput" wird aktualisiert, um den Pfad der entschlüsselten Datei anzuzeigen. Wenn das Entschlüsseln fehlschlägt, was bedeutet dass das Passwort falsch war, wird das Label "passwordStrength" auf "Passwort falsch" gesetzt.

Endprodukt


Florian Naef - Kleinprojekt
Pfad der Datei eingeben oder Datei in die Box ziehen:
Bestätigen
Passwort
Verschlüsseln
Entschlüsseln