

\$ GUIs

Shells für Nicht-Informatiker:innen

Letzte Großübung

Ulm University | Florian Sihler, Raphael Straub und Matthias Tichy | 16. Juli 2024



Software Engineering
Programming Languages



universität
uulm

Hehoo!

Gibt es vorab Wünsche?



Themenübersicht

1. Einfache GUIs

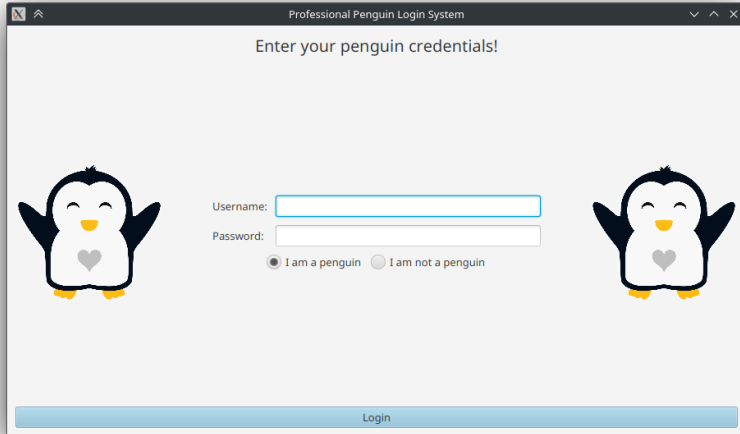
2. Anwendungen Bauen

3. Event! Event? Aaaaah

4. Es git so viel mehr...

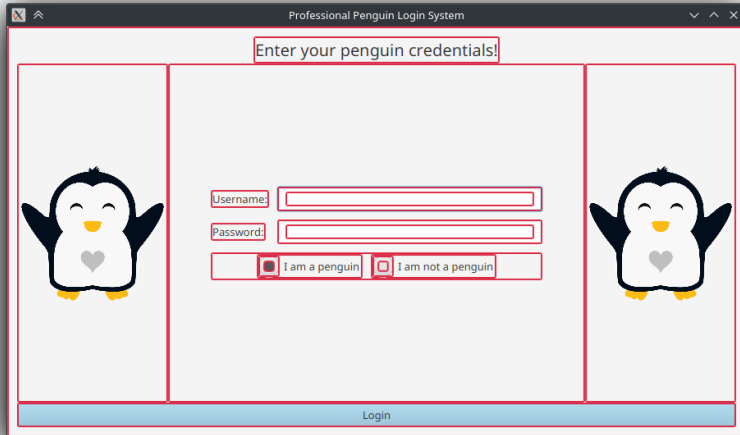
5. Fragen?

Graphische Benutzeroberflächen



- Welche Layouts?
 - Border-Pane
außen
 - Grid-Pane
Login-Felder
 - Flow-Pane
Buttons
- Welche Elemente?
 - Label
 - Image-View
 - TextField
 - RadioButton
 - ToggleGroup
 - Button

Graphische Benutzeroberflächen



- Welche Layouts?
 - Border-Pane
außen
 - Grid-Pane
Login-Felder
 - Flow-Pane
Buttons
- Welche Elemente?
 - Label
 - Image-View
 - TextField
 - RadioButton
 - ToggleGroup
 - Button

Der Scene Graph

Stage

└ Scene

└─ BorderPane

└─ Label

└─ ImageView

└─ ImageView

└─ GridPane

└─ Label

└─ Label

└─ TextField

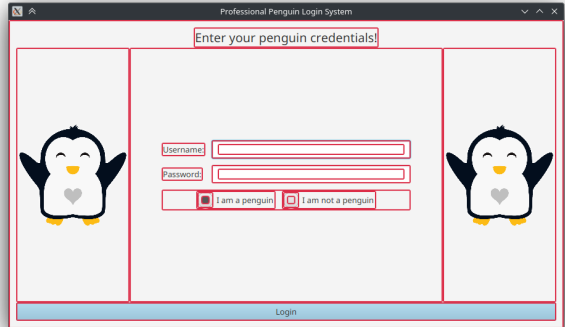
└─ PasswordField

└─ FlowPane

└─ RadioButton

└─ RadioButton

└─ Button



Themenübersicht

1. Einfache GUIs

2. Anwendungen Bauen

3. Event! Event? Aaaah

4. Es git so viel mehr...

5. Fragen?

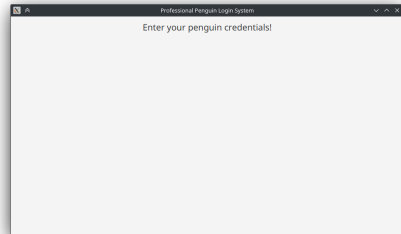
Ein Fenster!

```
public class Login extends Application {  
    public static void main(String[] args) {  
        launch(args);  
    }  
  
    public void start(Stage primaryStage) {  
        final var root = new BorderPane();  
  
        final var scene = new Scene(root, 900, 500);  
        primaryStage.setTitle("Professional_Penguin_Login_System");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
}
```



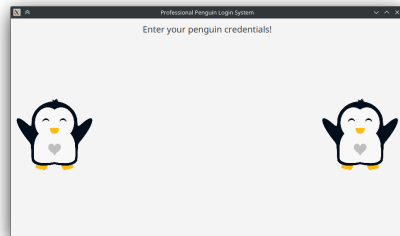
Ein Label

```
public void start(Stage primaryStage) {  
    final var root = new BorderPane();  
  
    final var mainInfoLabel = new Label();  
    mainInfoLabel.setAlignment(Pos.CENTER);  
    mainInfoLabel.setContentDisplay(ContentDisplay.TOP);  
    mainInfoLabel.setFont(Font.font("Arial", 20));  
    mainInfoLabel.setText(currentLabelText);  
  
    root.setTop(mainInfoLabel);  
    BorderPane.setAlignment(mainInfoLabel, Pos.CENTER);  
}
```



Bilder

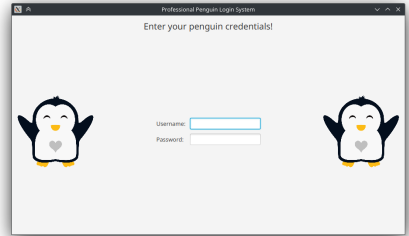
```
public void start(Stage primaryStage) {  
    final var root = new BorderPane();  
  
    final var penguImage = new Image(  
        getClass().getResourceAsStream("/pengu.png")  
    );  
    final var penguView = new ImageView(penguImage);  
    penguView.setPreserveRatio(true);  
    penguView.setFitWidth(0.2 * 900); // width of application  
    root.setLeft(penguView);  
  
    final var penguViewB = new ImageView(penguImage);  
    penguViewB.setPreserveRatio(true);  
    penguViewB.setFitWidth(0.2 * 900);  
    root.setRight(penguViewB);  
    // ...  
}
```



Do not use magic numbers (900)! (*see project*)

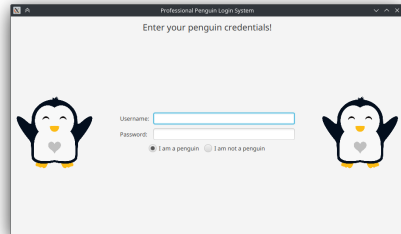
Login Basics

```
public void start(Stage primaryStage) {  
    final var root = new BorderPane();  
    final var grid = new GridPane();  
    grid.setAlignment(Pos.CENTER);  
    grid.setHgap(10);  
    grid.setVgap(10);  
  
    grid.add(new Label("Username:"), 0, 0);  
    grid.add(new Label("Password:"), 0, 1);  
  
    final var usernameInput = new TextField();  
    grid.add(usernameInput, 1, 0);  
    final var passwordInput = new PasswordField();  
    grid.add(passwordInput, 1, 1);  
  
    root.setCenter(grid);  
}
```



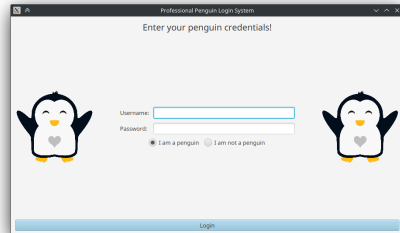
Radio Buttons

```
public void start(Stage primaryStage) {  
    final var root = new BorderPane();  
    final var grid = new GridPane();  
  
    final var group = new ToggleGroup();  
    final var penguinButton = new RadioButton("I am a penguin");  
    penguinButton.setToggleGroup(group);  
    penguinButton.setSelected(true);  
    final var notPenguinButton = new RadioButton("I am not a penguin");  
    notPenguinButton.setToggleGroup(group);  
  
    final var buttons = new FlowPane();  
    buttons.setAlignment(Pos.CENTER);  
    buttons.setHgap(10);  
    buttons.getChildren().addAll(penguinButton, notPenguinButton);  
    grid.add(buttons, 0, 2, 2, 1); // start at 0/2, span 2 columns  
}
```



Der Anmeldeknopf

```
public void start(Stage primaryStage) {  
    final var root = new BorderPane();  
    final var loginButton = new Button("Login");  
  
    loginButton.setMaxSize(Double.MAX_VALUE, Double.MAX_VALUE);  
    loginButton.setDefaultButton(true);  
    root.setBottom(loginButton);  
}
```



Themenübersicht

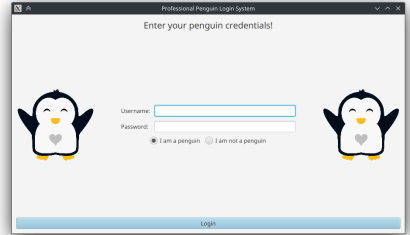
1. Einfache GUIs
2. Anwendungen Bauen
- 3. Event! Event? Aaaah**
4. Es git so viel mehr...
5. Fragen?

Events!

- Wir wollen, dass nur Pinguine sich anmelden können!

```
notPenguinButton.setOnAction(e -> {  
    mainInfoLabel.setText("You are not a penguin!");  
    loginButton.setDisable(true);  
});
```

```
penguinButton.setOnAction(e -> {  
    mainInfoLabel.setText("Enter your penguin credentials!");  
    loginButton.setDisable(false);  
});
```



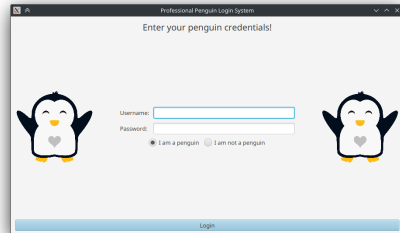
Einloggen!

- Wir möchten uns überhaupt anmelden...

```
loginButton.setOnAction(e -> {  
    if (/* amazing verification */) {  
        mainInfoLabel.setText("Welcome, _penguin!");  
        return;  
    }  
});
```

```
mainInfoLabel.setText("Invalid_credentials!");  
mainInfoLabel.setStyle("-fx-text-fill:_red;_ -fx-font-weight:_bold;");  
loginButton.setDisable(true);
```

```
final var transition = new PauseTransition(Duration.seconds(3));  
transition.setOnFinished(e -> { /* reset settings */ });  
transition.play();  
});
```



Was spricht gegen Threads?

GUI-Manipulationen müssen auf den main thread.

Themenübersicht

1. Einfache GUIs
2. Anwendungen Bauen
3. Event! Event? Aaaah
- 4. Es git so viel mehr...**
5. Fragen?

Man darf ja träumen...

- Komponenten für unsere Anwendung basteln
- Stil und Design allgemein beeinflussen
- Funktionen und Logik auslagern (z.B., MVC)
- Unzählige Weitere Themen beleuchten:
 - Sicherheit & Validierung
 - Ressourcen-Management
 - Internationalisierung
 - ...

Themenübersicht

1. Einfache GUIs
2. Anwendungen Bauen
3. Event! Event? Aaaah
4. Es git so viel mehr...
- 5. Fragen?**

Dies ist die letzte Großübung!
Fragen?