

Aufgaben für WPF Layout

Allgemeine Hinweise (für alle Aufgaben)

Texte anzeigen

- Ein Label zeigt Text über die Eigenschaft Content an.

Beispiel:

```
<Label Content="Hallo Welt"/>
```

Elemente benennen

Damit man ein Element im C# Code benutzen kann, muss es einen Namen haben:

Beispiel:

```
<Label Name="lblText"/>
```

Im Code kann man dann schreiben:

```
lblText.Content = "Neuer Text";
```

Button Klick Events

Ein Button kann eine Funktion ausführen.

Dazu nutzt man die Eigenschaft:

```
Click="MethodeName"
```

Beispiel:

```
<Button Content="Prüfen" Click="Check_Click"/>
```

Wenn man im Designer doppelt auf den Button klickt, wird die Methode automatisch erstellt.

Layout Manager

In dieser Übung werden verschiedene Layout Manager verwendet:

- Grid → Tabelle
- StackPanel → Elemente untereinander / nebeneinander
- WrapPanel → Elemente umbrechen automatisch
- DockPanel → Elemente an Seiten andocken

Vorgegebene Grundstruktur

Die folgende TabControl ist bereits erstellt und darf nicht verändert werden:

```
<TabControl>
  <TabItem Header="Registrierung">
    <!-- Hier Aufgabe 1 bearbeiten -->
  </TabItem>

  <TabItem Header="Einkauf">
    <!-- Hier Aufgabe 2 bearbeiten -->
  </TabItem>

  <TabItem Header="Mail App">
    <!-- Hier Aufgabe 3 bearbeiten -->
  </TabItem>
</TabControl>
```

Aufgabe 1 - Passwortprüfung

Erstelle eine Passwortprüfung mit folgenden Regeln:

Das Passwort muss:

- mindestens 8 Zeichen lang sein
- mindestens eine Zahl enthalten
- mindestens einen Großbuchstaben enthalten
- mindestens einen Kleinbuchstaben enthalten

MainWindow

Registrierung Einkauf Mail App

Passwort festlegen

Passwort:

.....

Passwort prüfen

Passwort gültig

Vorgehen

Nutze eine GroupBox für das Formular.

Verwende ein Grid, um die Elemente untereinander anzuordnen.

Füge hinzu:

- Label „Passwort“
- PasswordBox
- Button „Passwort prüfen“
- Label für Ergebnis

Tipps

Einzelne Zeichen prüfen

Ein String kann Buchstabe für Buchstabe geprüft werden:

```
char c = string[i];
```

Prüfen auf Eigenschaften

```
if (char.IsDigit(c))
{
    // enthält Zahl
}
if (char.IsUpper(c))
{
    // enthält Großbuchstabe
}
if (char.IsLower(c))
{
    // enthält Kleinbuchstabe
}
```

Ergebnis anzeigen

Den Text eines Labels kann man ändern mit:

```
lblResult.Content = "Text";
```

Aufgabe 2 - Einkaufswagen

Erstelle eine kleine Einkaufsliste.

Es soll angezeigt werden, wie viele Artikel im Einkaufswagen sind.



Vorgehen

Verwende ein StackPanel für die Hauptanordnung.

Ganz oben:

```
<Label Name="lblCount" Content="Artikel im Einkaufswagen: 0"/>
```

Darunter ein WrapPanel, damit Produkte nebeneinander angezeigt werden.

Für jedes Produkt:

- eine GroupBox
- darin ein StackPanel
- zwei Buttons:
 - Kaufen
 - Entfernen

Tipps

Gesamtzahl berechnen

Alle Artikel zusammen ergeben die Gesamtzahl:

```
cartCount = breadCount + waterCount;
```

Anzeige aktualisieren

```
lblCount.Content = "Artikel im Einkaufswagen: " + cartCount;
```

Aufgabe 3 - Mail App Layout

Erstelle ein einfaches Mail-App Layout mit DockPanel.

Beim Klick auf einen Button soll der Inhalt angezeigt werden.



Du bist im Bereich: Posteingang

Vorgehen

Verwende ein DockPanel.

Oben:

- TextBlock mit Titel „Meine Mail App“

Links:

- StackPanel mit Buttons :
 - Posteingang
 - Gesendet
 - Papierkorb

Mitte:

- TextBlock mit Name txtContent

Dieser zeigt später den ausgewählten Bereich an.

Text formatieren

Größe ändern:

FontSize="18"

Schriftart ändern:

FontWeight="Bold"

Text zentrieren:

HorizontalAlignment="Center"

VerticalAlignment="Center"

Tipps für den C# Code

Den geklickten Button ermitteln

Button btn = sender as Button;

Text des Buttons auslesen

btn.Content.ToString()