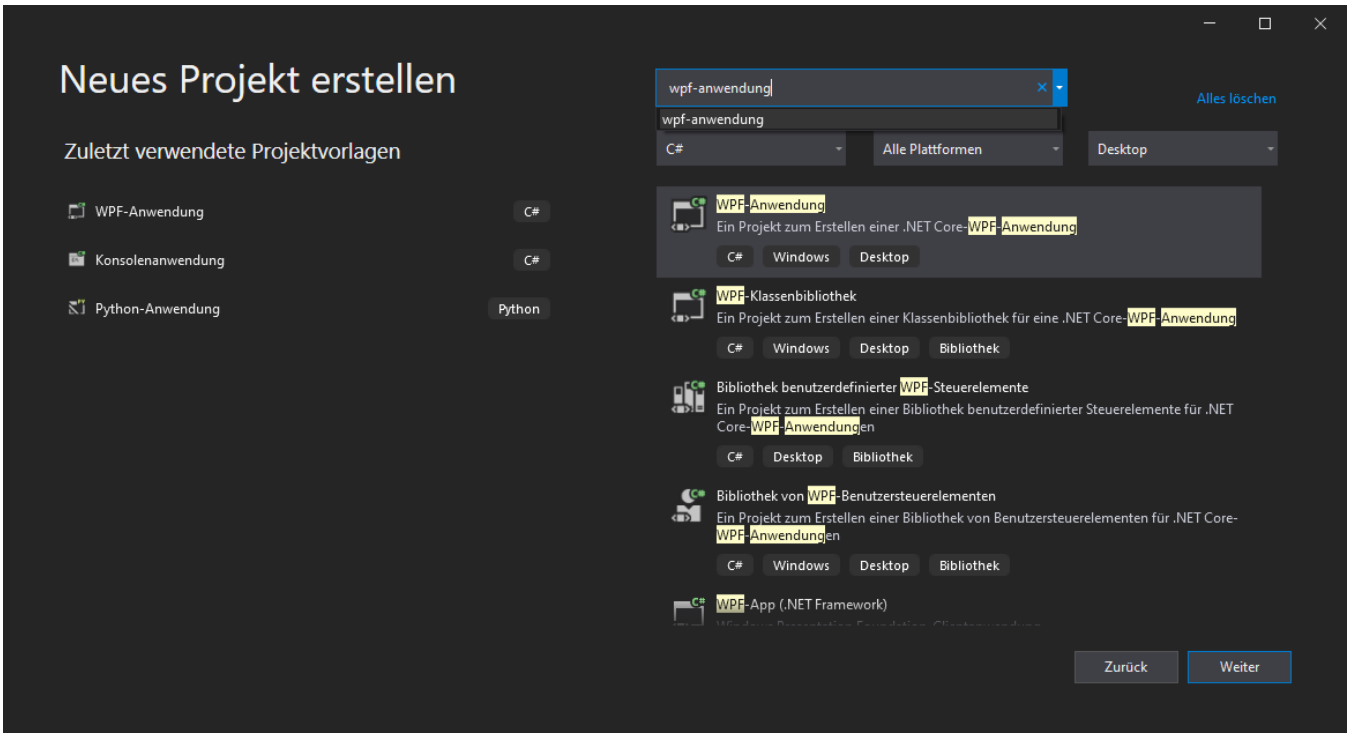
Beispielprojekt: Passwortgenerator

Visual-Studio-WPF-Projekt für einen zufallsgesteuerten Passwortgenerator

Inhaltsverzeichnis

Schritt für Schritt Anleitung

Neue WPF-Anwendung erstellen



- WPF-Anwendung
- Name z.B.: "Passwortgenerator"
- Speicherort wählen

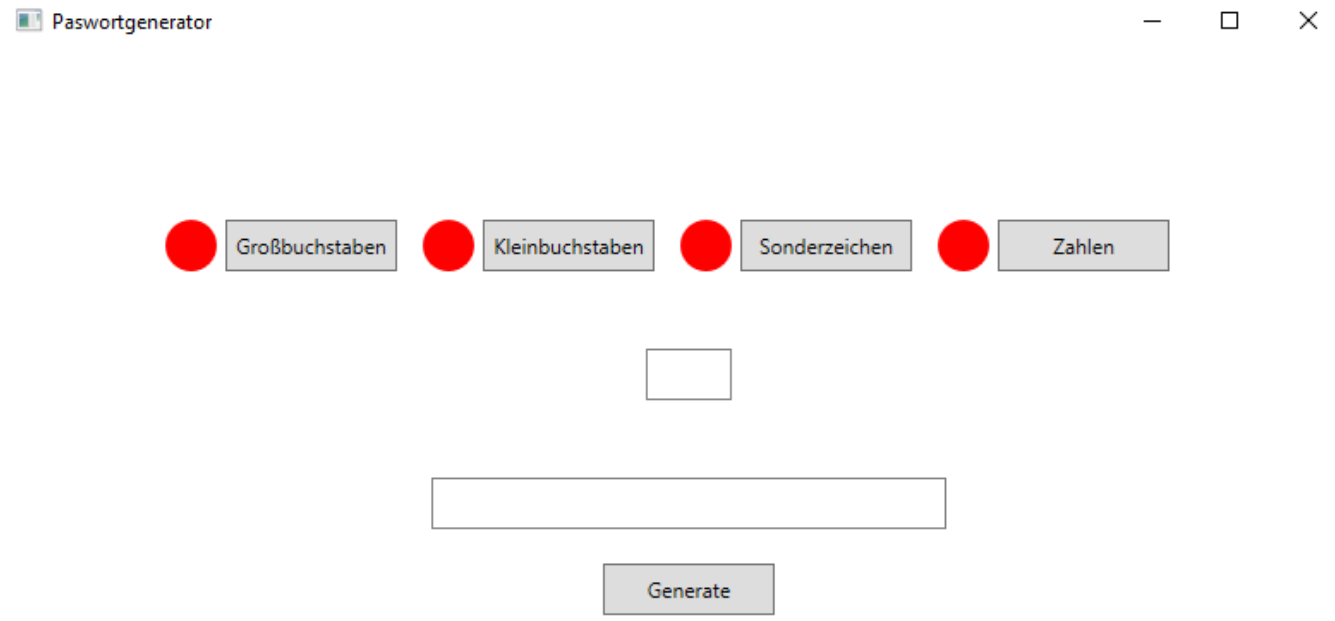
XAML-Code

XAML-Code

```
<Window x:Class="Projektname_hier_einfügen.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Projektname_hier_einfügen"
        mc:Ignorable="d"
        Title="Paswortgenerator" Height="450" Width="800">

    <Canvas>
        <Button x:Name="Großbuchstaben" Content="Großbuchstaben" Height="30" Width="100" Canvas.Left="130"
Canvas.Top="100" Click="Großbuchstaben_Click"></Button>
        <Ellipse x:Name="Ellipse_Großbuchstaben" Height="30" Width="30" Fill="Red" Canvas.Left="95" Canvas.Top="
100" ></Ellipse>
        <Button x:Name="Kleinbuchstaben" Content="Kleinbuchstaben" Height="30" Width="100" Canvas.Left="280"
Canvas.Top="100" Click="Kleinbuchstaben_Click"></Button>
        <Ellipse x:Name="Ellipse_Zahlen" Height="30" Width="30" Fill="Red" Canvas.Left="545" Canvas.Top="100" ><
/Ellipse>
        <Button x:Name="Sonderzeichen" Content="Sonderzeichen" Height="30" Width="100" Canvas.Left="430" Canvas.
Top="100" Click="Sonderzeichen_Click"></Button>
        <Ellipse x:Name="Ellipse_Sonderzeichen" Height="30" Width="30" Fill="Red" Canvas.Left="395" Canvas.Top="
100" ></Ellipse>
        <Button x:Name="Zahlen" Content="Zahlen" Height="30" Width="100" Canvas.Left="580" Canvas.Top="100"
Click="Zahlen_Click"></Button>
        <Ellipse x:Name="Ellipse_Kleinbuchstaben" Height="30" Width="30" Fill="Red" Canvas.Left="245" Canvas.
Top="100" ></Ellipse>
        <TextBox x:Name="Länge" BorderBrush="Gray" BorderThickness="1px" Height="30" Width="50" Canvas.Left="
375" Canvas.Top="175" VerticalContentAlignment="Center"/>
        <TextBox x:Name="Ausgabe" Height="30" Width="300" Canvas.Left="250" Canvas.Top="250" BorderBrush="Gray"
BorderThickness="1px"></TextBox>
        <Button x:Name="Generate" Content="Generate" Height="30" Width="100" Canvas.Top="300" Canvas.Left="350"
Click="Generate_Click"></Button>
    </Canvas>
</Window>
```

- Hier erstellen wir den Code für die Benutzeroberfläche
- Die Platzhalter Projektname_hier_einfügen dürft ihr mit eurem Projektnamen austauschen
- In diesem Code werden 4 Buttons erstellt um zu bestimmen aus was das Passwort bestehen soll
- Dazu kommen 4 Kreise die anzeigen welche Funktion an ist und welche aus
- Eine "TextBox" für die Bestimmung der Länge
- Eine "TextBox" welches Fehler und das generierte Passwort anzeigt
- Und einen Button für die Funktion zum Generieren des Passwortes

Das Ganze sollte dann wie folgt aussehen:



Code Behind

- Noch kann unser Programm nicht viel
- Dafür fügen wir jetzt unseren Code ein. Über den Tab "MainWindow.xaml.cs" oder über die Tastenkombination Strg + F6 gelangt ihr in den Code Behind
    - Wir deklarieren die Variablen
    - Wir geben unseren Knöpfen eine Funktion
    - Wir kreieren unsere Funktion für die Zufallsgeneration des Passwortes

Deklarieren der Variablen:

Variablen

```
    /// <summary>
    /// Interaction logic for MainWindow.xaml
/// </summary>
public partial class MainWindow : Window
{
    private bool großbuchstaben = false;
    private bool kleinbuchstaben = false;
    private bool sonderzeichen = false;
    private bool zahlen = false;
    private int length;

    public MainWindow()
    {
        InitializeComponent();
    }
    }
```

- Wir benötigen vier Variablen des Typs Bool für Großbuchstaben, Kleinbuchstaben, Zahlen und Sonderzeichen
- Dazu benötigen wir noch einen Integer für die Länge

Nun brauchen unsere Buttons auch noch eine Funktion:

Konfiguration der Buttons für die Bestimmung der Zeichenkette:

- Bei dem Klicken eines Buttons wechselt der boolesche Ausdruck von True auf False und auch wieder von False auf True
- Zusätzlich machen wir eine If-Abfrage um unsere Ellipse zu füllen
    - Wir füllen die Ellipse grün wenn der Bool gleich True ist
    - Oder füllen die Ellipse rot wenn der Bool gleich False ist

Zeichenkette Button's

```
        private void Großbuchstaben_Click(object sender, RoutedEventArgs e)
        {
                        //tauscht den boolschen ausdruck
            großbuchstaben = !großbuchstaben;
            if (großbuchstaben == false)
            {
                            //Färbt die Ellipse Rot sollte die Funktion ausgeschaltet sein
                Ellipse_Großbuchstaben.Fill = Brushes.Red;
            }
            else
            {
                            //Färbt die Ellipse Grün wenn due Funktion eingeschaltet ist
                Ellipse_Großbuchstaben.Fill = Brushes.Green;
            }
        }

        private void Kleinbuchstaben_Click(object sender, RoutedEventArgs e)
        {
                        //tauscht den boolschen ausdruck
            kleinbuchstaben = !kleinbuchstaben;
            if (kleinbuchstaben == false)
            {
                            //Färbt die Ellipse Rot sollte die Funktion ausgeschaltet sein
                Ellipse_Kleinbuchstaben.Fill = Brushes.Red;
            }
            else
            {
                            //Färbt die Ellipse Grün wenn due Funktion eingeschaltet ist
                Ellipse_Kleinbuchstaben.Fill = Brushes.Green;
            }
        }

        private void Sonderzeichen_Click(object sender, RoutedEventArgs e)
        {
                        //tauscht den boolschen ausdruck
            sonderzeichen = !sonderzeichen;
            if (sonderzeichen == false)
            {
                            //Färbt die Ellipse Rot sollte die Funktion ausgeschaltet sein
                Ellipse_Sonderzeichen.Fill = Brushes.Red;
            }
            else
            {
                            //Färbt die Ellipse Grün wenn due Funktion eingeschaltet ist
                Ellipse_Sonderzeichen.Fill = Brushes.Green;
            }
        }

        private void Zahlen_Click(object sender, RoutedEventArgs e)
        {
                        //tauscht den boolschen ausdruck
            zahlen = !zahlen;
            if (zahlen == false)
            {
                            //Färbt die Ellipse Rot sollte die Funktion ausgeschaltet sein
                Ellipse_Zahlen.Fill = Brushes.Red;
            }
            else
            {
                            //Färbt die Ellipse Grün wenn due Funktion eingeschaltet ist
                Ellipse_Zahlen.Fill = Brushes.Green;
            }
        }
```

Konfiguration des Buttons für das Ausführen der Funktion:

- Beim Klick auf den Button speichert er die Eingabe unserer TextBox für die Länge in unsere Variable "length"
- Zusätzlich wollen wir dem Benutzer ein Feedback geben - sollte er einen Fehler gemacht haben - das machen wir mit Hilfe einer If-Abfrage
    - Für den Fall, dass der Benutzer keine Option für die Zeichenkette ausgewählt hat, lassen wir in unserem Label einen Text ausgeben um ihn darauf hin zu weisen, dass er mindestens eine Auswahl treffen muss
    - Für den Fall, dass die eingegebene Länge gleich null ist oder keine Eingabe gemacht wurde, lassen wir in unserem Label wieder einen Text ausgeben um ihn darauf hinzuweisen
    - Sollten aber alle Bedingungen erfüllt sein, befüllen wir das Label mit dem Rückgabewert unserer Funktion welche wir im nächsten Schritt erstellen werden

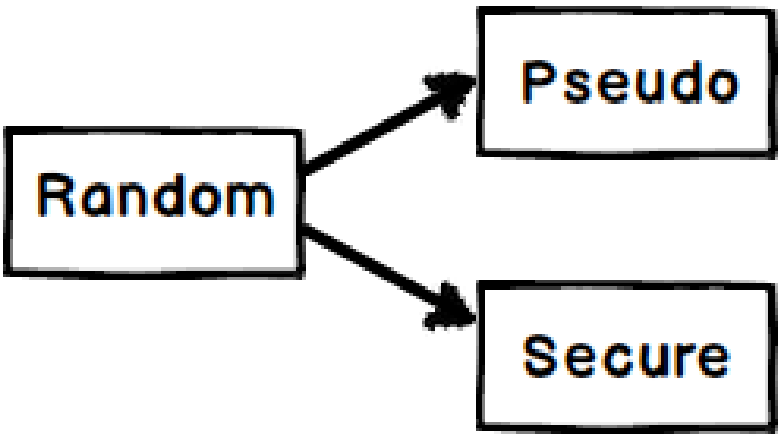
Button zum Generieren des Passwortes

```
private void Generate_Click(object sender, RoutedEventArgs e)
        {
                        //Speichert die in die TextBox eingegebene zahl in unserer lenght Variable
            length = Convert.ToInt32(Länge.Text);

            if (kleinbuchstaben == false && großbuchstaben == false && sonderzeichen == false && zahlen ==
false)
            {
                Ausgabe.Text = "Bitte mindestens eine Auswahl treffen";
            }
            else if (length == 0)
            {
                Ausgabe.Text = "Bitte eine Passwortlänge angeben";
            }
            else
            {
                Ausgabe.Text = CreatePassword(kleinbuchstaben, großbuchstaben, zahlen, sonderzeichen, length);
            }
        }
```

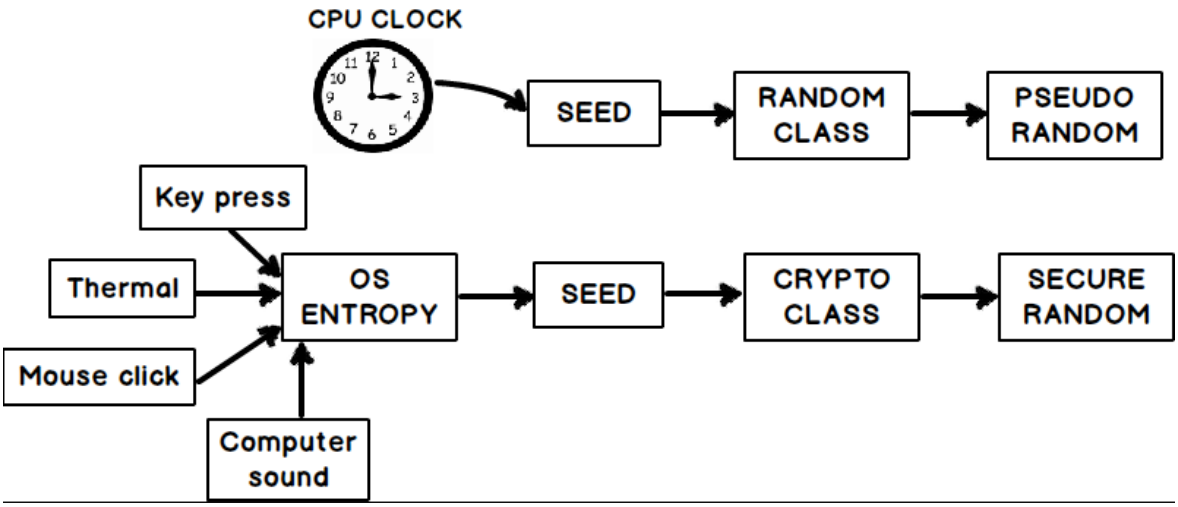
Erstellen der Funktion für die Zufallsgeneration eines Passwortes:

Was ist eigentlich Zufall in der Programmierung?

- Es gibt kein richtiges Random in der Programmierung sondern es gibt zwei arten von Random es gibt "Pseudo-" und es gibt "Secure-" random



- Um etwas zufällig zu machen brauchen wir einen Trigger (Seed)
- Bei Menschen "generiert" sich dieser Seed in Bruchteilen einer Sekunde
  - Bei der Frage Grün, Rot oder Gelb spielen viele Faktoren eine Rolle wie z.B.: Lieblingsfarbe, Glücksfarbe, etc.
- Bei einem Computer wird der Seed, in unserem Beispiel, mit Hilfe der Random Klasse generiert
  - Diese generiert einen Seed über die CPU Clock
- Für unser Beispiel ist das vollkommen ausreichend, in gewissen Spielen oder anderen Anwendungen muss aber ein Secure Random gewährleistet sein, dass könnte man mit Hilfe der "RNGCryptoServiceProvider" Klasse sicherstellen diese wird für sachen wie z.B.: Verschlüsselung verwendet



Programmierung der Methode:

- Hierfür erstellen wir und eine Methode mit dem Namen GeneratePasswort und übermitteln dieser unsere vier bool Variablen und unsere int Variable für die Länge
- Wir deklarieren eine leere string Variable mit dem Namen "alphabet"
- Danach fragen wir unseren booleschen Ausdrücke ab und je nach True oder False werden die Zeichen in unsere Zeichenkette hinzugefügt
- Wir erstellen ein neues StringBuilder Objekt mit dem Namen "result" und erstellen eine neues Random Objekt mit dem Namen "r"
- Mit Hilfe einer While Schleife werden so lange neue Zahlen zufallsgeneriert bis unsere Zeichenkette die eingegebene Länge hat
  - Bei jedem Generieren einer Zahl wird die Stelle von unserem "alphabet"-String zu unseren "result"-String hinzugefügt

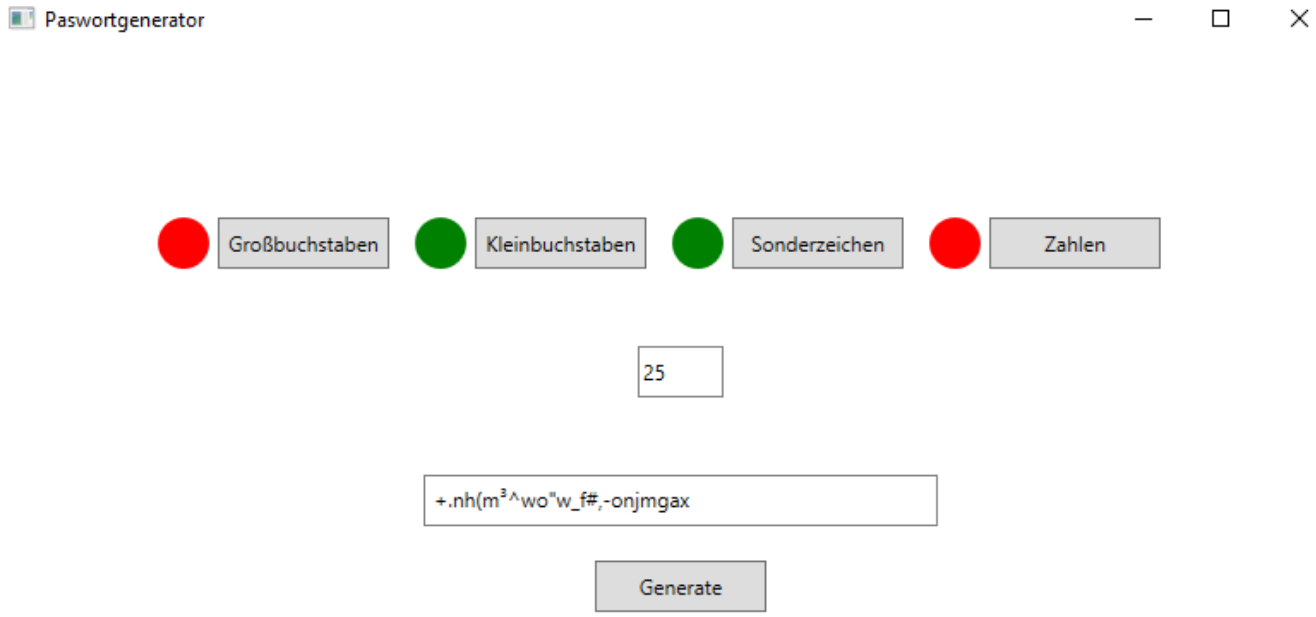
- Anschließend wird uns ein String zurückgegeben, welchen wir wie vorher im Button bestimmt in unser Label ausgeben

CreatePassword Methode

```
private static string CreatePassword(bool lc, bool up, bool dg, bool sc, int len)
        {
            string alphabet = "";

            if (lc)
                alphabet += "abcdefghijklmnopqrstuvwxyz";
            if (up)
                alphabet += "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
            if (dg)
                alphabet += "0123456789";
            if (sc)
                alphabet += "^!\"§$%&/()=?²³{[]}\\`´+*~#',.-;:_<>|";

            if (alphabet.Length == 0)
                return "error";

            StringBuilder result = new StringBuilder();
            Random r = new Random();

            while (result.Length < len)
                result.Append(alphabet[r.Next(alphabet.Length)]);

            return result.ToString();
        }
```

Das fertige Projekt

- Wenn alles fertig ist und alles richtig gemacht wurde sollte unser Projekt wie folgt aussehen:


Paswortgenerator
Großbuchstaben
Kleinbuchstaben
Sonderzeichen
Zahlen
25
+.nh(m³^wo"w_f#,-onjmgax
Generate