

# Lab 1 Wprowadzenie do WPF

Mateusz Wiśniewski  
Politechnika Śląska

Zadanie 1 i 2:

Elastyczny interfejs oraz klasa Użytkownik z zaimplementowaną metodą obliczającą wiek użytkownika.

```
namespace Lab1.Model
{
    interface IUser
    {
        string FirstName { get; set; }
        string LastName { get; set; }
        int YearOfBirth { get; set; }
    }
}
```

```
namespace Lab1.Model
{
    class User : IUser
    {
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public int YearOfBirth { get; set; }
        public string City { get; set; }
        public string Street { get; set; }
        public int HouseNumber { get; set; }
        public string PostalCode { get; set; }
        public string PESEL { get; set; }

        public User() { }

        public User(string firstName, string lastName, int yearOfBirth, string city, string street, int houseNumber, string postalCode, string pesel)
        {
            FirstName = firstName;
            LastName = lastName;
            YearOfBirth = yearOfBirth;
            City = city;
            Street = street;
            HouseNumber = houseNumber;
            PostalCode = postalCode;
            PESEL = pesel;
        }

        public int GetAge()
        {
            DateTime date = DateTime.Now;
            int year = date.Year;
            return year - this.YearOfBirth;
        }
    }
}
```

### Zadanie 3:

Walidator numeru PESEL, w przypadku braku numeru PESEL generuje go z wprowadzonych danych.

```
namespace Lab1.Model
{
    class PeselValidator
    {
        public User User { get; set; }
        public PeselValidator(User user)
        {
            this.User = user;
        }

        public bool IsValid()
        {
            return (User.PESEL.Length > 0 && User.PESEL != null);
        }

        public string GeneratePesel()
        {
            string generatedPesel = User.YearOfBirth.ToString().Substring(2, 2);

            generatedPesel += "0801";

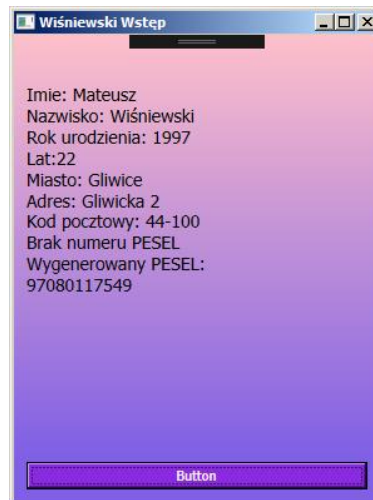
            Random rand = new Random();

            generatedPesel += rand.Next(10000, 99999);

            return generatedPesel;
        }
    }
}
```

### Zadanie 4:

Interfejs graficzny stworzony przy pomocy XAMLa.



Kod XAML:

```
Window x:Class="Lab1.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:Lab1"
    Title="Wiśniewski Wstęp" Height="400" Width="300">
    <Grid>
        <Grid.Background>
            <LinearGradientBrush EndPoint="0.5,0.9" StartPoint="0.5,0">
                <GradientStop Color="Pink"/>
                <GradientStop Color="Blue" Offset="2"/>
            </LinearGradientBrush>
        </Grid.Background>
        <Button Content="Button" HorizontalAlignment="Left" VerticalAlignment="Top" Width="272" Margin="10,341,0,0" Height="22"
            Click="Button_Click" Foreground="FloralWhite" Background="BlueViolet"/>
        <TextBlock x:Name="Message" HorizontalAlignment="Left" TextWrapping="Wrap" VerticalAlignment="Top" Margin="10,39,0,0" FontSize="14"/>
    </Grid>
</Window>
```

Zadanie teoretyczne:

Jak zmodyfikować zadanie by sprawdzić działanie dziedziczenia?

Odpowiedź:

Można stworzyć klasę Administrator, która będzie dziedziczyć po klasie User i będzie miała dodatkowe pole np. data otrzymania uprawnień administratora (DateTime AdminSince).