

**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**Факултет „Компютърни системи и технологии“**

**Курсова работа по ПС**

**Преподавател: ас. П. Маринов**

Изработил: Иван Ангелов Иванов

Група: 39

Факултетен № 121217003

Подпис:

**Задание**

**Тема 10**: Предложете реализация на проекта StudentInfoSystem с възможност за избор между поне 3 различни стила за цялото приложение.

**Описание на използваните класове и методи**

**Class Styles** – използва се за изграждането на методите за стилизация на приложението;

**static public void DefaultStyle(Login login)** – променя стила на прозореца Login по подразбиране;

**static public void GrayStyle(Login login)** – променя стила на прозореца Login в сиви и бели нюанси;

**static public void DarkSlateStyle(Login login)** – променя стила на прозореца Login в тъмно-зелени и бели нюанси;

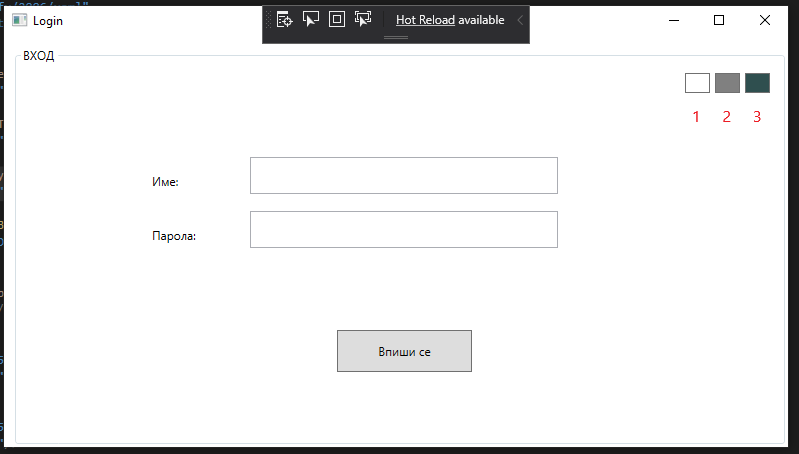
**static public void MainWindowGrayStyle(MainWindw mainWindow)** – променя стила на прозореца MainWindow в сиви и бели нюанси;

**static public void MainWindowDarkSlateStyle(MainWindow mainWindow)** – променя стила на прозореца MainWindow в сиви и бели нюанси;

**static public MainWindow ChosenStyle(Login login, MainWindow mainWindow)** – прави проверка в зависимост какъв стил е избран за прозореца Login, за да зададе съответния и за прозореца MainWindow;

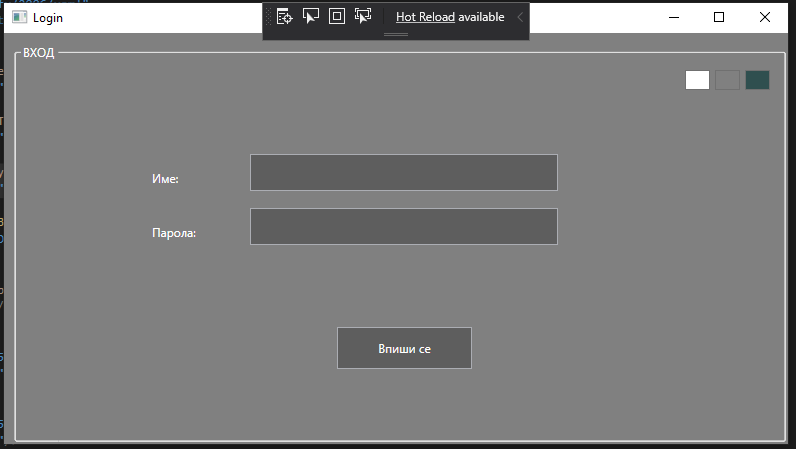
**Код и описание на функционалността**

Със стартирането на приложението ни се отваря прозореца за вход, където имаме три опции за избор на стил на приложението.

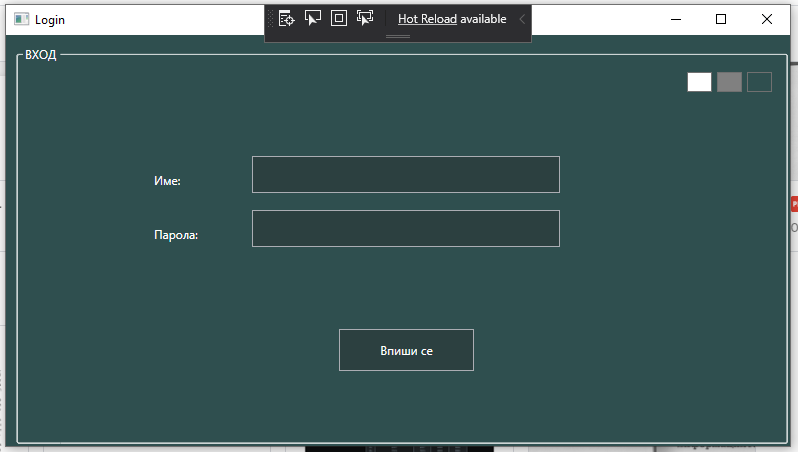


*Фиг.1 Тема по подразбиране*

С натискането на първия бутон избираме стила по подразбиране като в снимката по-горе. При натискането на втория и третия бутон, на прозореца **Login** му се задава сива или тъмно-зелена тема.



*Фиг. 2 Сива тема*



*Фиг. 3 Тъмно-зелена тема*

За промяната на атрибутите на всеки елемент е използван шаблон за стилизация във файла **App.xaml.**

Source код:

<Application.Resources>

<Style x:Key="DefaultStyle" TargetType="{x:Type Grid}">

<Setter Property="Background" Value="White"/>

</Style>

<Style x:Key="GroupBoxDefault" TargetType="{x:Type GroupBox}">

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style x:Key="DefaultTextBox" TargetType="{x:Type TextBox}">

<Setter Property="Background" Value="White"/>

</Style>

<Style x:Key="DefaultBtn" TargetType="{x:Type Button}">

<Setter Property="Background" Value="#DDDDDD"/>

</Style>

<Style x:Key="GrayGridStyle" TargetType="{x:Type Grid}">

<Setter Property="Background" Value="Gray"/>

</Style>

<Style x:Key="GrayTextBox" TargetType="{x:Type TextBox}">

<Setter Property="Background" Value="#5e5e5e"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<Style x:Key="GrayListBox" TargetType="{x:Type ListBox}">

<Setter Property="Background" Value="#5e5e5e"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<Style x:Key="GrayBtn" TargetType="{x:Type Button}">

<Setter Property="Background" Value="#5e5e5e"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="BorderBrush" Value="#ABADB3"/>

</Style>

<Style x:Key="DarkGridStyle" TargetType="{x:Type Grid}">

<Setter Property="Background" Value="DarkSlateGray"/>

</Style>

<Style x:Key="DarkTextBox" TargetType="{x:Type TextBox}">

<Setter Property="Background" Value="#2c4040"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<Style x:Key="DarkListBox" TargetType="{x:Type ListBox}">

<Setter Property="Background" Value="#2c4040"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<Style x:Key="DarkBtn" TargetType="{x:Type Button}">

<Setter Property="Background" Value="#2c4040"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="BorderBrush" Value="#ABADB3"/>

</Style>

<Style x:Key="WhiteTextStyle" TargetType="{x:Type Label}">

<Setter Property="Foreground" Value="White"/>

</Style>

<Style x:Key="BlackTextStyle" TargetType="{x:Type Label}">

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style x:Key="GroupBoxStyle" TargetType="{x:Type GroupBox}">

<Setter Property="Foreground" Value="White"/>

<Setter Property="BorderBrush" Value="White"/>

<Setter Property="BorderThickness" Value="0.5"/>

</Style>

</Application.Resources>

Като във всеки елемент **Style** e зададен ключ, който да се отнася до съответните елементи при извикването на методите за промяна на стила.

За трите бутона (номерирани на фиг.1) са създадени Click методи в code-behind на **Login**, като съответно всеки един от тях извика съответния метод, който е необходим за промяна на стила.

Source код:

private void styleOne\_Click(object sender, RoutedEventArgs e)

{

Styles.DefaultStyle(this);

}

private void styleTwo\_Click(object sender, RoutedEventArgs e)

{

Styles.GrayStyle(this);

}

private void styleThree\_Click(object sender, RoutedEventArgs e)

{

Styles.DarkSlateStyle(this);

}

В **клас Styles** са описани и трите метода:

static public void DefaultStyle(Login login)

{

login.MainGrid.Style = (Style)Application.Current.Resources["DefaultStyle"];

login.GroupBox.Style = (Style)Application.Current.Resources["GroupBoxDefault"];

login.loginBtn.Style = (Style)Application.Current.Resources["DefaultBtn"];

foreach (Label l in login.SubGrid.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["BlackTextStyle"];

}

foreach (TextBox tb in login.SubGrid.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["DefaultTextBox"];

}

}

static public void GrayStyle(Login login)

{

login.MainGrid.Style = (Style)Application.Current.Resources["GrayGridStyle"];

login.GroupBox.Style = (Style)Application.Current.Resources["GroupBoxStyle"];

login.loginBtn.Style = (Style)Application.Current.Resources["GrayBtn"];

foreach (Label l in login.SubGrid.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["WhiteTextStyle"];

}

foreach (TextBox tb in login.SubGrid.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["GrayTextBox"];

}

}

static public void DarkSlateStyle(Login login)

{

login.MainGrid.Style = (Style)Application.Current.Resources["DarkGridStyle"];

login.GroupBox.Style = (Style)Application.Current.Resources["GroupBoxStyle"];

login.loginBtn.Style = (Style)Application.Current.Resources["DarkBtn"];

foreach (Label l in login.SubGrid.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["WhiteTextStyle"];

}

foreach (TextBox tb in login.SubGrid.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["DarkTextBox"];

}

}

Трите метода притежават по един параметър от тип **Login,** който сочи към текущия прозорец. За да можем да променим стойностите на атрибутите на даден елемент използваме следният ред:

(Style)Application.Current.Resources[Key] – търси в **App.xaml** елемент(ресурс) от тип Style със съответния ключ Key.

Така за всеки елемент задаваме конкретния стил, който желаем. В случай, че някои от елементите се съдържа два или повече пъти, използваме foreach цикъл като търсим елементите от даден тип (Children.OfType<Label>()) в друг елемент (SubGrid).

Правим същото и за стила на **MainWindow** в **класа Styles**, като методите приемат параметър от тип **MainWindow**.

Source код:

static public void MainWindowGrayStyle(MainWindow mainWindow)

{

mainWindow.MainGrid.Style = (Style)Application.Current.Resources["GrayGridStyle"];

mainWindow.status.Style = (Style)Application.Current.Resources["GrayListBox"];

foreach (Label l in mainWindow.Names.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["WhiteTextStyle"];

}

foreach (Label l in mainWindow.StudentInfo.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["WhiteTextStyle"];

}

foreach (TextBox tb in mainWindow.Names.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["GrayTextBox"];

}

foreach (TextBox tb in mainWindow.StudentInfo.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["GrayTextBox"];

}

foreach (GroupBox gb in mainWindow.MainGrid.Children.OfType<GroupBox>())

{

gb.Style = (Style)Application.Current.Resources["GroupBoxStyle"];

}

foreach (Button b in mainWindow.MainGrid.Children.OfType<Button>())

{

b.Style = (Style)Application.Current.Resources["GrayBtn"];

}

}

static public void MainWindowDarkSlateStyle(MainWindow mainWindow)

{

mainWindow.MainGrid.Style = (Style)Application.Current.Resources["DarkGridStyle"];

mainWindow.status.Style = (Style)Application.Current.Resources["DarkListBox"];

foreach (Label l in mainWindow.Names.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["WhiteTextStyle"];

}

foreach (Label l in mainWindow.StudentInfo.Children.OfType<Label>())

{

l.Style = (Style)Application.Current.Resources["WhiteTextStyle"];

}

foreach (TextBox tb in mainWindow.Names.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["DarkTextBox"];

}

foreach (TextBox tb in mainWindow.StudentInfo.Children.OfType<TextBox>())

{

tb.Style = (Style)Application.Current.Resources["DarkTextBox"];

}

foreach (GroupBox gb in mainWindow.MainGrid.Children.OfType<GroupBox>())

{

gb.Style = (Style)Application.Current.Resources["GroupBoxStyle"];

}

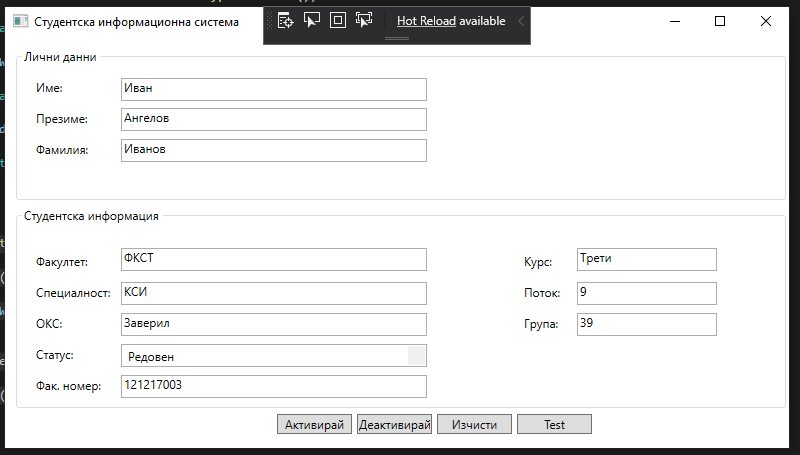
foreach (Button b in mainWindow.MainGrid.Children.OfType<Button>())

{

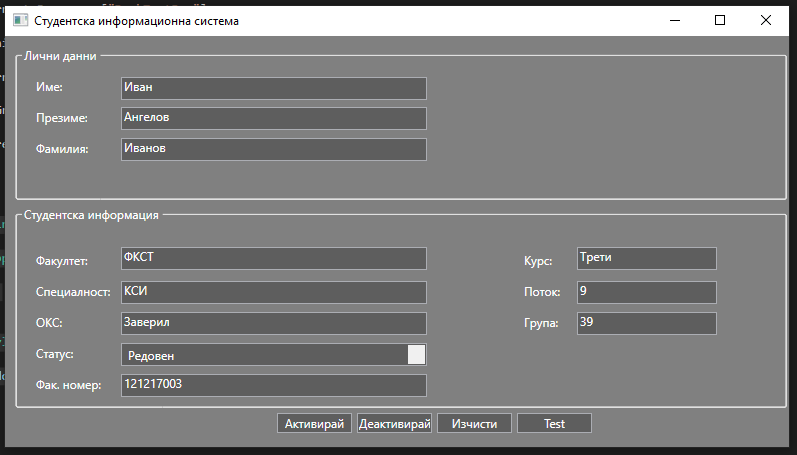
b.Style = (Style)Application.Current.Resources["DarkBtn"];

}

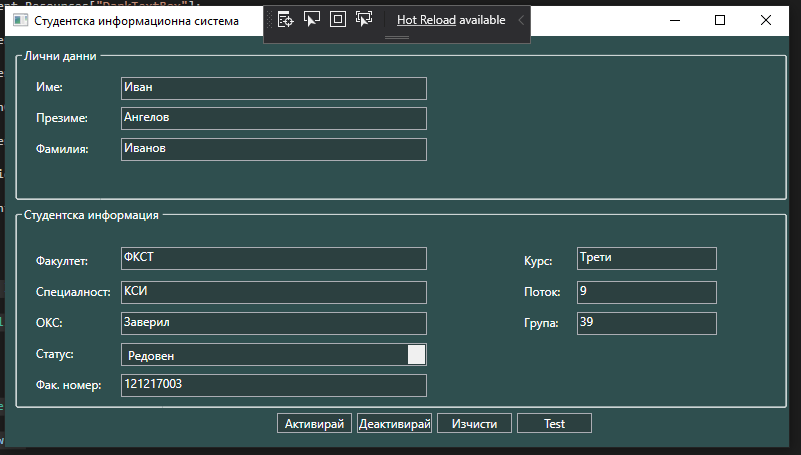
}



*Фиг. 4 Tема по подразбиране*



*Фиг. 5 Сива тема*



*Фиг. 6 Тъмно-зелена тема*

И тъй като и в **Login**, и в **MainWindow** искаме да имаме един същ стил, ще използваме метода **ChosenStyle(…).**

Source код:

static public MainWindow ChosenStyle(Login login, MainWindow mainWindow)

{

if (login.MainGrid.Style == (Style)Application.Current.Resources["GrayGridStyle"])

{

MainWindowGrayStyle(mainWindow);

return mainWindow;

}

else if (login.MainGrid.Style == (Style)Application.Current.Resources["DarkGridStyle"])

{

MainWindowDarkSlateStyle(mainWindow);

return mainWindow;

}

else

{

return mainWindow;

}

}

Методът приема два параметъра - от тип **Login** и **MainWindow** проверява какъв е стилът на **MainGrid** в **Login,** и в зависимост от това какъв е избран се извиква съответния метод (освен за стила по подразбиране) за стилизиране на **МainWindow** и връща като резултат прозореца **MainWindow** с избрания стил.