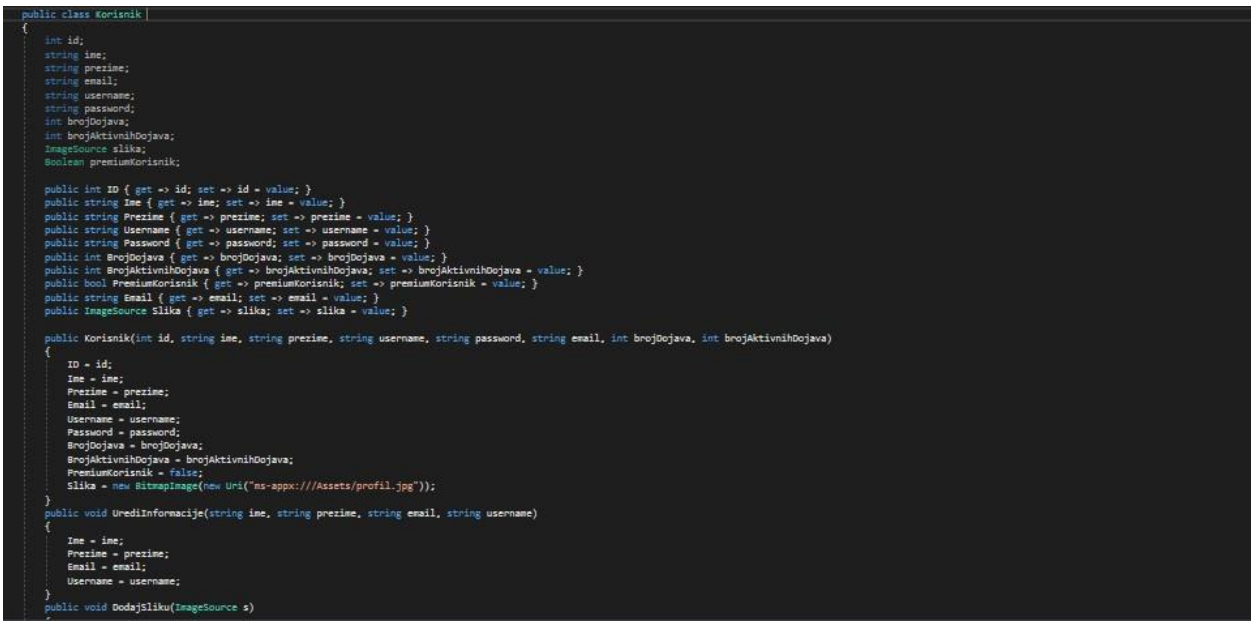
1. **DECORATOR PATTERN**

Napravljena je klasa PremiumKorisnik na kojoj je primjnjen ovaj patern, tj ona je i klasa Korisnik nasljeđuju isti interface, i klasa PremiumKorisnik ima instancu klase Ikorisnik i u njoj su dodani dodatni atributi i metode.

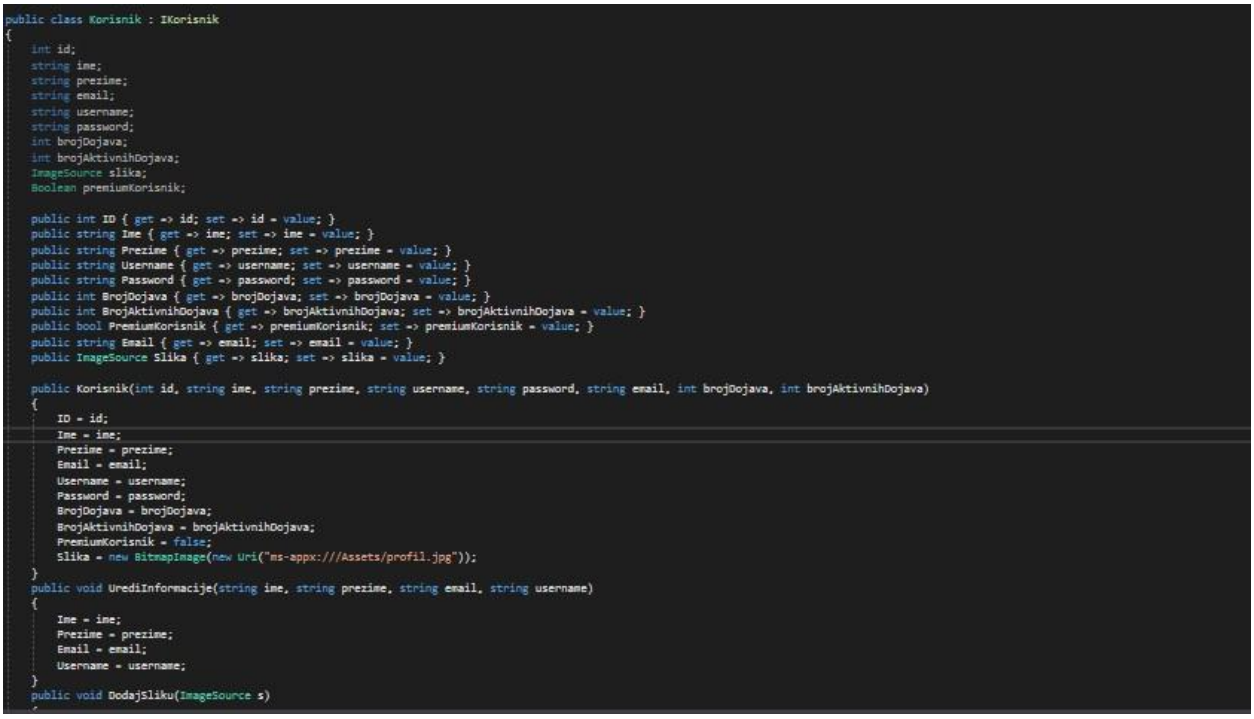


```
public class Korisnik
{
    int id;
    string ime;
    string prezime;
    string email;
    string username;
    string password;
    int brojDojava;
    int brojAktivnihDojava;
    ImageSource slika;
    Boolean premiumKorisnik;

    public int ID { get => id; set => id = value; }
    public string Ime { get => ime; set => ime = value; }
    public string Prezime { get => prezime; set => prezime = value; }
    public string Username { get => username; set => username = value; }
    public string Password { get => password; set => password = value; }
    public int BrojDojava { get => brojDojava; set => brojDojava = value; }
    public int BrojAktivnihDojava { get => brojAktivnihDojava; set => brojAktivnihDojava = value; }
    public bool PremiumKorisnik { get => premiumKorisnik; set => premiumKorisnik = value; }
    public string Email { get => email; set => email = value; }
    public ImageSource Slika { get => slika; set => slika = value; }

    public Korisnik(int id, string ime, string prezime, string username, string password, string email, int brojDojava, int brojAktivnihDojava)
    {
        ID = id;
        Ime = ime;
        Prezime = prezime;
        Email = email;
        Username = username;
        Password = password;
        BrojDojava = brojDojava;
        BrojAktivnihDojava = brojAktivnihDojava;
        PremiumKorisnik = false;
        Slika = new BitmapImage(new Uri("ms-appx:///Assets/profil.jpg"));
    }
    public void UrediInformacije(string ime, string prezime, string email, string username)
    {
        Ime = ime;
        Prezime = prezime;
        Email = email;
        Username = username;
    }
    public void DodajSliku(ImageSource s)
```

Slika 1 - Klasa Korisnik prije primjene paterna



```
public class Korisnik : IKorisnik
{
    int id;
    string ime;
    string prezime;
    string email;
    string username;
    string password;
    int brojDojava;
    int brojAktivnihDojava;
    ImageSource slika;
    Boolean premiumKorisnik;

    public int ID { get => id; set => id = value; }
    public string Ime { get => ime; set => ime = value; }
    public string Prezime { get => prezime; set => prezime = value; }
    public string Username { get => username; set => username = value; }
    public string Password { get => password; set => password = value; }
    public int BrojDojava { get => brojDojava; set => brojDojava = value; }
    public int BrojAktivnihDojava { get => brojAktivnihDojava; set => brojAktivnihDojava = value; }
    public bool PremiumKorisnik { get => premiumKorisnik; set => premiumKorisnik = value; }
    public string Email { get => email; set => email = value; }
    public ImageSource Slika { get => slika; set => slika = value; }

    public Korisnik(int id, string ime, string prezime, string username, string password, string email, int brojDojava, int brojAktivnihDojava)
    {
        ID = id;
        Ime = ime;
        Prezime = prezime;
        Email = email;
        Username = username;
        Password = password;
        BrojDojava = brojDojava;
        BrojAktivnihDojava = brojAktivnihDojava;
        PremiumKorisnik = false;
        Slika = new BitmapImage(new Uri("ms-appx:///Assets/profil.jpg"));
    }
    public void UrediInformacije(string ime, string prezime, string email, string username)
    {
        Ime = ime;
        Prezime = prezime;
        Email = email;
        Username = username;
    }
    public void DodajSliku(ImageSource s)
```

Slika 2 – Klasa korisnik nakon primjene paterna

```
using Windows.UI.Xaml.Media;

namespace eRouting2
{
    public interface IKorisnik
    {


        void DodajSliku(ImageSource s);
        void UrediInformacije(string ime, string prezime, string email, string username);
    }
}
```

Slika 2 – Interface IKorisnik

```
using System.Threading.Tasks;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Media.Imaging;

namespace eRouting2
{
    public class PremiumKorisnik : IKorisnik
    {
        IKorisnik k;
        double ocjena;

        public double Ocjena { get => ocjena; set => ocjena = value; }

        public PremiumKorisnik(int id, string ime, string prezime, string username, string password, string email, int brojDojava, int brojAktivnihDojava)
        {
            k = new Korisnik(id, ime, prezime, username, password, email, brojDojava, brojAktivnihDojava);

        }
        public void UrediInformacije(string ime, string prezime, string email, string username)
        {
            k.UrediInformacije(ime, prezime, email, username);
        }
        public void DodajSliku(ImageSource s)
        {
            k.DodajSliku(s);
        }
        public void Ocijeni (double o)
        {
            Ocjena = o;
        }
```

Slika 3 – Klasa PremiumKorisnik

2. REPLACE A MAGIC NUMBER WITH A NAMED CONSTANT

Vrijednost na koju se inicijalizira ocjena za Premium korisnika je zamjenjena sa konstantom

```
public class PremiumKorisnik : IKorisnik
{

    IKorisnik k;
    double ocjena;
    int brOcjena;

    public double Ocjena { get => ocjena; set => ocjena = value; }
    public int BrOcjena { get => brOcjena; set => brOcjena = value; }

    public PremiumKorisnik(int id, string ime, string prezime, string username, string password, string email, int brojDojava, int brojAktivnihDojava)
    {
        k = new Korisnik(id, ime, prezime, username, password, email, brojDojava, brojAktivnihDojava);
        Ocjena = 5.0;
        BrOcjena = 1;

    }
    public void UrediInformacije(string ime, string prezime, string email, string username)
    {
        k.UrediInformacije(ime, prezime, email, username);
    }
```

Slika 4 – Klasa prije primjene refaktoringa

```
public class PremiumKorisnik : IKorisnik
{
    const double oc = 5.0;
    const int  b = 1;
    IKorisnik k;
    double ocjena;
    int brOcjena;

    public double Ocjena { get => ocjena; set => ocjena = value; }
    public int BrOcjena { get => brOcjena; set => brOcjena = value; }

    public PremiumKorisnik(int id, string ime, string prezime, string username, string password, string email, int brojDojava, int brojAktivnihDojava)
    {
        k = new Korisnik(id, ime, prezime, username, password, email, brojDojava, brojAktivnihDojava);
        Ocjena = oc;
        BrOcjena = b;

    }
    public void UrediInformacije(string ime, string prezime, string email, string username)
    {
        k.UrediInformacije(ime, prezime, email, username);
    }
```

Slika 5 – Klasa poslije refaktoringa

3. DECOMPOSE A BOOLEAN EXPRESSION

Pojednostavljivanje boolean izraza sa dobro imenovanim među varijablama

```
private void Button_Click_3(object sender, RoutedEventArgs e)
{
    if (TextBoxIme.Text == string.Empty || TextBoxPrezime.Text == string.Empty || TextBoxEmail.Text == string.Empty || TextBoxUsername.Text == string.Empty)
    {
        TextError.Text = "Polja ne smiju biti prazna";
        return;
    }
    else if (ViewModel.DaLiJeDostupanUsername(TextBoxUsername.Text) == false && korisnik.Username!=TextBoxUsername.Text)
    {
        TextError.Text = "Ovaj username vec postoji, izaberite neki drugi username";
        return;
    }
    else {
        korisnik.UrediInformacije(TextBoxIme.Text, TextBoxPrezime.Text, TextBoxEmail.Text, TextBoxUsername.Text);
        ViewModel.UredjivanjeKorisnika(korisnik);
        TextError.Text = "";
        MessageDialog msgbox = new MessageDialog("Uspješno ste uredili podatke");
        msgbox.ShowAsync();
    }
```

Slika 6 – Metoda prije refaktoringa

```
private void Button_Click_3(object sender, RoutedEventArgs e)
{
    bool dostupan = ViewModel.DaLiJeDostupanUsername(TextBoxUsername.Text);
    bool isti = (korisnik.Username != TextBoxUsername.Text);
    if (TextBoxIme.Text == string.Empty || TextBoxPrezime.Text == string.Empty || TextBoxEmail.Text == string.Empty || TextBoxUsername.Text == string.Empty)
    {
        TextError.Text = "Polja ne smiju biti prazna";
        return;
    }
    else if ( !dostupan && isti)
    {
        TextError.Text = "Ovaj username vec postoji, izaberite neki drugi username";
        return;
    }
    else {
        korisnik.UrediInformacije(TextBoxIme.Text, TextBoxPrezime.Text, TextBoxEmail.Text, TextBoxUsername.Text);
        ViewModel.UredjivanjeKorisnika(korisnik);
        TextError.Text = "";
        MessageDialog msgbox = new MessageDialog("Uspješno ste uredili podatke");
        msgbox.ShowAsync();
    }
```

Slika 7 – Metoda poslije refaktoringa