

Uvod u pravljenje Windows Form Aplikacija

- Pravimo novi projekat sa "Create a Project"
- Biramo C# - Windows Form Application
- Form
 - Appearance->Text: ShopApp
 - Appearance->FormBorderStyle: Sizable -> None
 - Ne želimo da imamo podrazumevani Windows okvir za našu aplikaciju već ćemo praviti svoj
 - Layout -> Size: 900, 550
 - Layout -> MinimumSize: 900, 550
 - Želimo da radimo na minimalnoj Formi, da ne bi morali stalno da proveravamo i merimo
 - WindowStyle->Icon: ...
 - Možemo da promenimo ikonu koja predstavlja našu aplikaciju
 - Layout->StartPosition: CenterScreen
- Buttons (za gašenje i spuštanje prozora u taskbar)
 - Appearance->FlatStyle: Flat
 - Appearance->FlatAppearance->BorderColor: 0
 - Appearance->Text:
 - Design->Name: btnExit / btnMinimize
 - Appearance->BackgroundImage: ...
 - Appearance->BackgroundImageLayout: Stretch
 - Pišemo kod za ova dva dugmeta (dvoklik na buttons na formi da bi se stvorile funkcije btnExit_Click i btnMinimize_Click u Form1.cs)

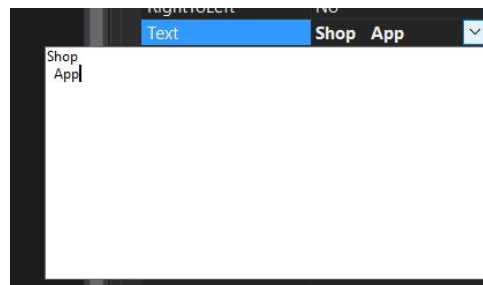


```
private void btnExit_Click(object sender, EventArgs e)
{
    //this.Close();
    //ne mozemo da koristimo this.Close() jer ce se tada zatvoriti samo ova forma a ne cela aplikacija
    //to je problem ako u aplikaciji imamo više od jedne forme
    Application.Exit();
}

private void btnMinimize_Click(object sender, EventArgs e)
{
    WindowState = FormWindowState.Minimized;
}
```

- SidePanel
 - U ovaj panel stavićemo meni pomoću kojeg će se koristiti aplikacija
 - Layout->Dock: Left
 - Appearance->BackColor: Black
 - Design->Name: panelSide

- Label Logo
 - Labela je kontrola koja služi da prikazuje neki tekst na koji ne može da utiče korisnik
 - Design->Name: labelLogo
 - Appearance->Text: Shop App
 - Ako kliknemo strelicu pored teksta možemo da pišemo labelu u više redova
 - Appearance->Font: OCR, bold, 28
 - Appearance->ForeColor: White



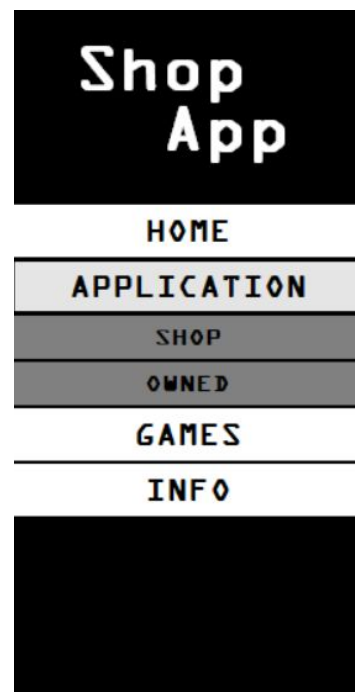
- Panel za meni
 - Nalazi se ispod Logo labele
 - Design->Name: panelMenu
 - Layout->Dock: Bottom
 - Razvučemo ga sve do Logo labele
- Menu buttons
 - Appearance->Font: OCR, bold 16
 - Appearance->FlatStyle: Flat
 - Appearance->BackColor: White
 - Layout->Dock: Top
 - Appearance->Text: HOME / APPLICATION / GAME / INFO
 - Za APPLICATION i GAME želimo da imamo meni koji ce da se spusti ispod dugmeta ako kliknemo na njega
 - Da bismo ovo mogli, nakon što napravimo to dugme, pravimo Panel koji stavljamo ispod dugmeta; U taj panel stavimo 2 manja dugmeta (možemo da kopiramo već napravljeno dugme i samo promenimo veličinu i BackColor)
 - Želimo da se ovaj pod-meni ne vidi kada se tek pokrene aplikacija
 - Behavior->Visible: False
 - Panel oblikujemo da bude veličine dva mala dugmeta
 - Da bi ovo pravilno radilo, **obavezno je da se radi po pravilnom redosledu jer će jedino tako svi delovi moći lepo da se pozicioniraju** (odnosno da im pravilo radi Dock osobina) - to znači da nove dugmiće ubacujemo u panel isključivo po redosledu po kojem želimo da budu poređani u samoj aplikaciji
 - ...->Button->(submenu panel->submenu Button->...->submenu Button)->Button->...
 - Želimo da nam se ovaj pod-meni otvori/zatvori kada kliknemo na dugme APPLICATION, odnosno GAMES



Form1.cs [Design]



Kada pokrenemo aplikaciju



Kada pritisnemo dugme

```
private void btnApplication_Click(object sender, EventArgs e)
{
    if (panelApp.Visible == false) //ako pod-meni nije prikazan, prikazi ga
    {
        panelApp.Visible = true;
    }
    else
    //ako izraz u if zagradama nije tacan (pod-meni jeste prikazan) onda cemo ga sakriti
    {
        panelApp.Visible = false;
    }
}

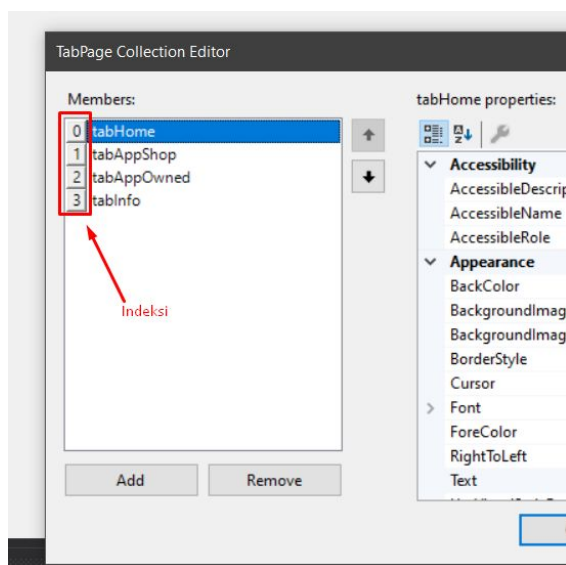
private void btnGames_Click(object sender, EventArgs e)
{
    if (panelGames.Visible == false)
    {
        panelGames.Visible = true;
    }
    else
    {
        panelGames.Visible = false;
    }
}
```

- TabControl - Main Tabovi

- Ovde će da nam budu Tabovi za Home, Applications, Games i Info
- Vidimo da ima gore tab dugmići, ali njih cemo kasnije da sklonimo
- Design->Name: tabControlMain
- Tab1 - HOME



- Design->Name: tabHome
- Appearance->Text: Home
- Dodati jos 3 taba sa istim settings za App shop, App owned i Info
 - Games necemo jer je isto kao apps
- Za menu dugmiće sa levog panela stavimo da mogu da promene tab koji se trenutno prikazuje
 - Za to nam je potrebno da znamo koji je indeks taba na koji želimo da promenimo
 - To možemo da vidimo u tabControl->TabPage->(Collection)



```
private void btnHome_Click(object sender, EventArgs e)
{
    tabControlMain.SelectTab(0);
    //menjamo trenutno prikazan tab na onaj sa indeksom 0
}

private void btnShopA_Click(object sender, EventArgs e)
{
    tabControlMain.SelectTab(1);
}

private void btnOwnedA_Click(object sender, EventArgs e)
{
    tabControlMain.SelectTab(2);
}

private void btnInfo_Click(object sender, EventArgs e)
{
    tabControlMain.SelectTab(3);
}
```

- tab INFO

- Buttons - Fb, Insta, Twitter
 - Možemo kopirati naše dugme za gašenje ili minizaciju
 - Appearance->BackgroundImage: ...
 - Selektujemo sva 3 dugmeta i stavimo da su podjednako odaljeni
 - Format->HorizontalSpacing->MakeEqual
 - Format->Align->Centers
 - Dodajemo im kod za onClick da odu na dati sajt

```
private void buttonFB_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("https://www.facebook.com");
}
```

- Sign In

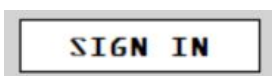
- Želimo da postoji mogućnost da se korisnik prijavi na svoj profil
- Pravimo Button za Sign In
 - Možemo da kopiramo Info button i malo ga izmenimo (Veličina i teks)
 - Layout->Dock: None
 - Design->Name: btnSign
 - Appearance->Text: SIGN IN
 - Želimo da dugme otvori novu formu u kojoj bismo uneli podatke za prijavljivanje
 - Moramo da napravimo novu formu
 - Project->AddForm: FormLog
 - Appearance->FormBorderStyle: None
 - Layout->StartPosition: CenterScreen
 - Postavimo željene dimenzije (ili u properties ili u Design)
 - Appearance->BackColor: LightGray
 - Želimo da se boja razlikuje od glavne forme
 - Mozemo da prekopiramo X dugme iz Form1 i damo mu istu funkcionalnost (da zatvori datu formu pri kliktanju)

```
private void btnExit_Click(object sender, EventArgs e)
{
    this.Close(); //ovde ne gasimo celu aplikaciju pa je potrebno samo formu da zatvorimo
}
```

- Želimo da se Forma upali pri kliktanju dugmeta na glavnoj formi

```
private void btnSign_Click(object sender, EventArgs e)
{
    FormLog logForm = new FormLog(); //stvaramo novu formu tipa FormLog sa nazivom logForm
    //logForm.Show(); //osvime prikazujemo formu, ali korisnik i dalje moze da se vrati na glavnu formu
    logForm.ShowDialog(); //sa ShowDialog() korisnik ne moze da koristi glavnu formu sve dok ne zatvori ovu
}
```

- Dodajemo 2 labela: Username i Password
 - Appearance->Font: OCR
 - Appearance->Text: ...
 - Appearance->Name: ...
- Dodajemo 2 textBoxa gde će se unositi informacije
 - Appearance->Font: OCR
 - Kod Password textBoxa stavljamo da se prikazuju * umesto teksta
 - Behavior->PasswordChar: *
- Dodajmo dugme Sign In kojime se proveravaju upisane informacije



- Želimo da ova forma salje glavnoj informaciju da se neko uspešno ulogovao
 - Ovo radimo tako što u konstruktoru Forme za prijavljivanje damo informaciju o tome koja ju je forma stvorila



```
FormLog logForm = new FormLog();
```

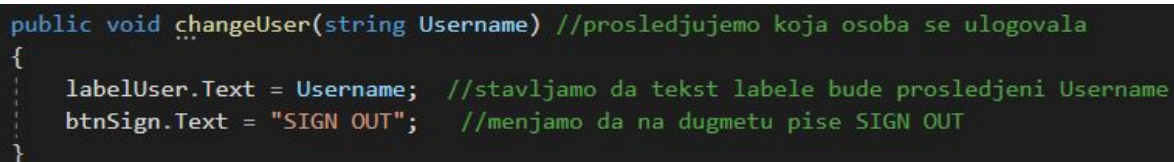
- Konstruktor se poziva odmah čim se stvori instance te klase
- Mi pozivamo ovaj konstruktor kada se klikne dugme Sign In na glavnoj formi
- Pišemo novi konstruktor ispod već postojećeg koji ima sposobnost da prihvati jedan argument - podatak o tome koje forma ga je pozvala

```
//originalni konstruktor koji se sam stvorio prilikom dodavanje forme
public FormLog()
{
    InitializeComponent();
}
//nas novi konstruktor
public FormLog(Form1 callingForm) //kao argument konstuktora zelimo da se prosledi glavna(pozivajuca) forma
{
    InitializeComponent(); //Obavezno, isto kao i u originalnom konstruktoru
    mainForm = callingForm;
}
private Form1 mainForm; //dodajemo Formi za SignIn promenljivu u kojoj se cuva koja forma ju je stvorila
```



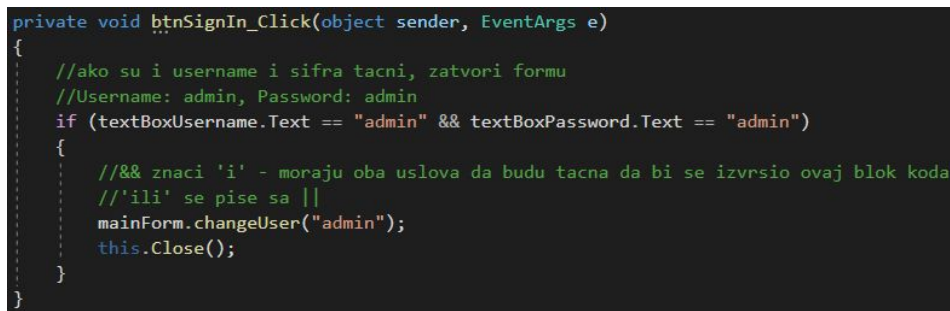
```
Form2 frm = new Form2(this);
```

- Zbog ovoga moramo malo da promenimo funkciju za kliktanje na Log In dugme u glavnoj formi
- Želimo da se pri ispravnom logovanju na glavnoj formi ispiše ko se ulogovao (u novoj labeli), a i promeni da na Sign In dugmetu piše Sign out
 - Da bismo ovo postigli, treba nam fukcija u **glavnoj Formi**
 - Ovu funkciju pozivamo kada se uspesno ulogujemo



```
public void changeUser(string Username) //prosledjujemo koja osoba se ulogovala
{
    labelUser.Text = Username; //stavljamo da tekst labele bude prosledjeni Username
    btnSign.Text = "SIGN OUT"; //menjamo da na dugmetu piše SIGN OUT
}
```

- Sada možemo da napišemo šta se desi kada se klikne SIGN IN dugme na novoj formi - proveravamo da li su ispravni Username i šifra



```
private void btnSignIn_Click(object sender, EventArgs e)
{
    //ako su i username i sifra tacni, zatvori formu
    //Username: admin, Password: admin
    if (textBoxUsername.Text == "admin" && textBoxPassword.Text == "admin")
    {
        //&& znaci 'i' - moraju oba uslova da budu tacna da bi se izvrsio ovaj blok koda
        //'ili' se piše sa ||
        mainForm.changeUser("admin");
        this.Close();
    }
}
```

- Konacno, zelimo da dugme LogIn/LogOut, odloguje korisnika ako je vec ulogovan

```
private void btnSign_Click(object sender, EventArgs e)
{
    if (labelUser.Text == "Not Signed In") //ako niko nije prijavljen, onda otvaramo formu za prijavu
    {
        FormLog logForm = new FormLog(); //stvaramo novu formu tipa FormLog sa nazivom logForm
        //logForm.Show(); //osvime prikazujemo formu, ali korisnik i dalje moze da se vrati na glavnu formu
        logForm.ShowDialog(); //sa ShowDialog() korisnik ne moze da koristi glavnu formu sve dok ne zatvori ovu
    }
    else
    {
        //u slucaju da je korisnik vec ulogovan, mi cemo ga odlogovati
        labelUser.Text = "Not Signed In";
        btnSign.Text = "SIGN IN";
    }
}
```

- Na tabControl zelimo konacno da namestimo da se tabovi ne vide:
 - Appearance->Text: ""
 - Behavior->ItemSize: 0, 1
 - Behavior->Appearance: FlatButtons
 - Behavior->SizeMode: Fixed