

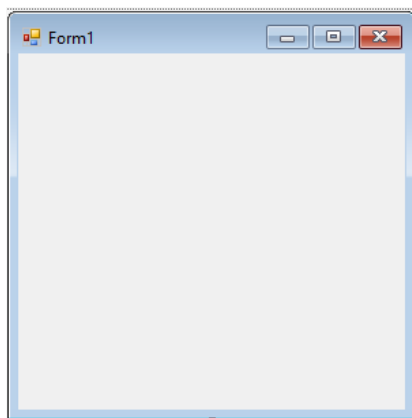


Основни елементи графичког окружења

Једна од предности рада са *Visual Basic* програмским језиком је могућност једноставног развоја апликација са графичким корисничким интерфејсом. У овом поглављу биће представљена четири основна елемента графичког корисничког интерфејса: форма, лабела, поље за унос текста и дугме. Примена осталих елемената, њихова својства и функционалност биће презенована у наредним поглављима.

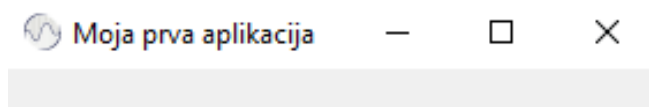
Форма

Први елемент који се аутоматски креира када креирате *Windows Forms Application* пројекат је форма или образац, (*Form*) (слика 1). Површина елемента је празна без додатних елемената, а из палете са алаткама могуће је превући друге елементе на форму.



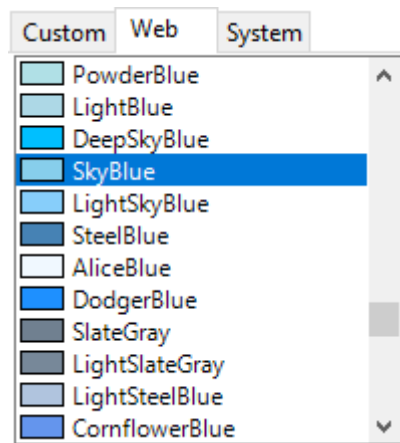
Слика 1. Графички елемент форма - *Form*

Форма има заглавље које се састоји од иконице и текста. То су својства *text* и *icon* која се могу променити у палети са својствима. За текст треба унети следећи садржај *Моја прва апликација* а за иконицу узети *ico* датотеку. За претварање било које слике у *ico* датотеку може се користити било који *online* сервис, који се коришћењем *Google*-а или неког другог претраживача може наћи. Кључне речи које треба унети у претраживач су *JPG to ICO* или *PNG to ICO*. За демонстрацију узет је лого школе који је конвертован из слике у иконицу. Резултат измена можете видети на слици 2.



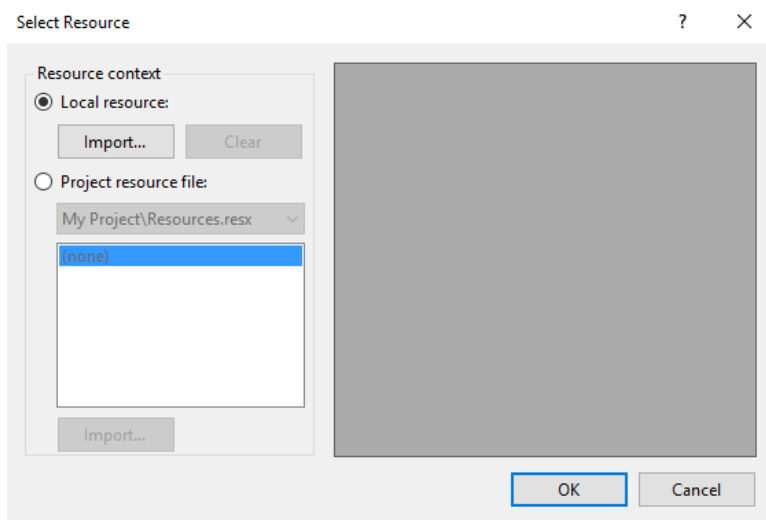
Слика 2. Заглавље *Windows Forms* апликације

Контејнер форме може да има своју боју позадине, или слику постављену као позадину. Својство за измену боје позадине је ***BackColor***. Кликом на опције постоје три могућности избора боја – *Custom*, *Web*, *System*. *Custom* су основне боје, *Web* су боје које се могу наћи у спецификацији веб претраживача, док *System* су предефинисане боје које *Microsoft* користи кроз своје графичке апликације. У *BackColor* својству одаберите боју *SkyBlue* (слика 3).



Слика 3. Палета за избор боја (web)

Поред боје позадине могуће је поставити слику. Слика може заменити позадину или јој бити додаток. Својство за додавање слике позадине је **BackgroundImage**, а начин приказа слике је **BackgroundImageLayout**. Коришћење слика или других мултимедијалних садржаја реализује се кроз управљач ресурсима (*Resource Manager*). Могуће је ресурс увести као датотеку са рачунара или креирањем специјалне датотеке (*Resources.resx*) која води евиденцију о ресурсима који се користе у пројекту (слика 4).



Слика 4. Коришћење управљача ресурсима

BackgroundImageLayout има неколико својстава за приказ слике:

None - приказује слику у природној величини почев од горњег левог угла,

Tile - понавља слику као плочице,

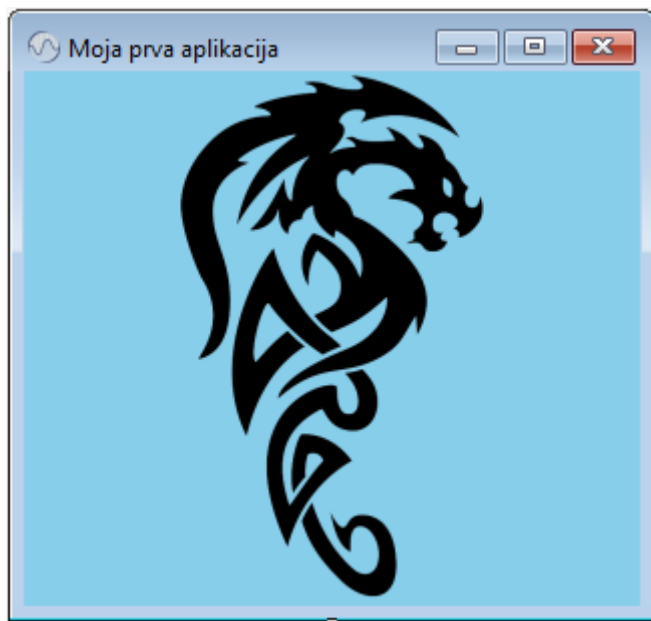
Center – позиционира слику у центар форме (уколико је слика већа од величине форме приказаће само део слике),

Stretch - прилагођава слику (растеже је) да одговара димензијама форме и

Zoom - адаптира слику према величини форме.



Као пример узета је транспарентна слика тетоваже змаја а за *BackgroundImageLayout* изабрана је вредност *Zoom*. Изглед форме са овако подешеним параметрима позадине приказан је на слици 5.



Слика 5. Пример коришћења слике за позадину

За измену величине форме користи се својство **Size** које се састоји од два својства *Width* и *Height* (слика 6). Уместо подразумеваних вредности могу се, на пример унети вредности дате на слици 6.

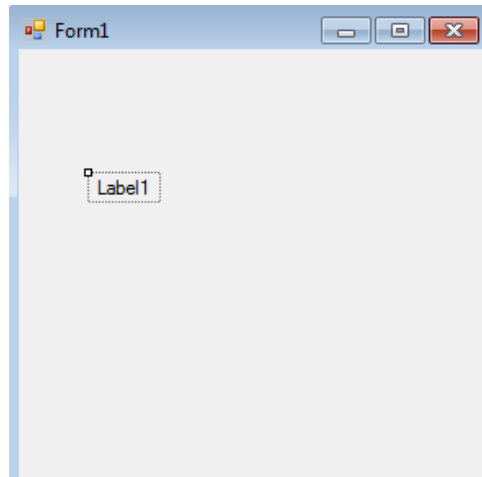
Size	600; 400
Width	600
Height	400

Слика 6. Својство Size и подсвојства – Width и Height

Сва својства форме могуће је изменити у самом коду.

Лабела

Лабела (*Label*) је елемент који се користи за приказ предефинисаног текста. За разлику од поља за унос текста, лабелу се текст може једино изменити писањем кода. Лабела је један од деце елемената форме. Из палете са алатима у секцији *Common Tools* елемент *Label* треба превући на форму. После постављања лабеле на форми она изгледа слично форми датој на слици 7.

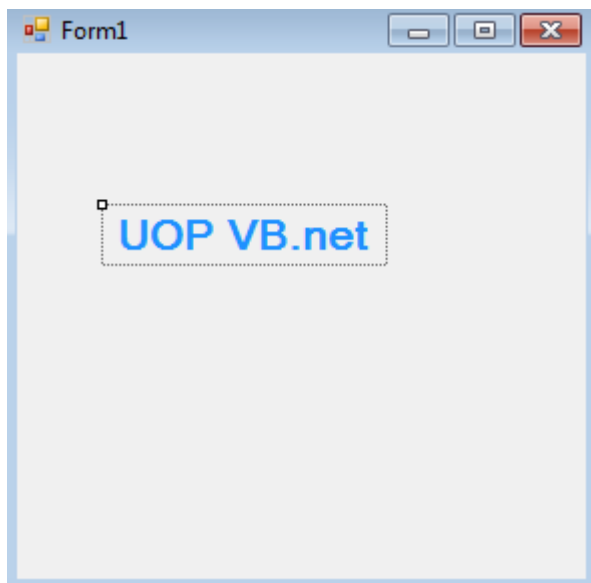


Слика 7. Форма са лабела елементом

Прво својство лабеле које треба изменити је *Text* својство које ће променити текст који се приказује на самој лабели. Уместо *Label1* унети текст у *UOP VB.net*.

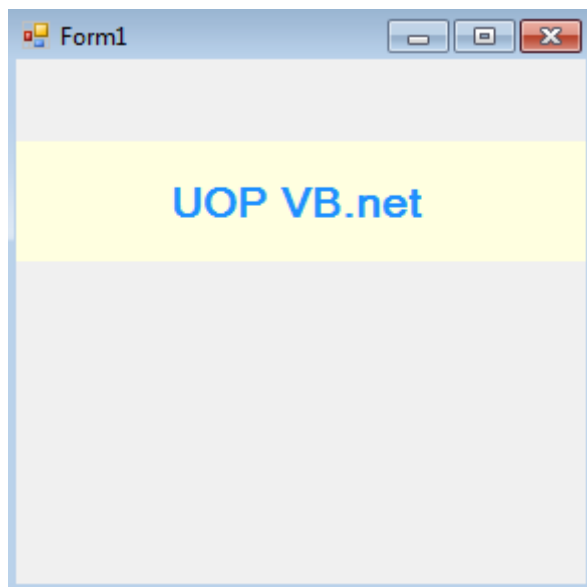
(*Name*) својство је важно својство и не служи за приказ текста већ за јединствени назив елемента и његову идентификацију. Када се елемент превуче на форму *Visual Studio* му даје подразумевано име које се састоји од имена типа елемента и редног броја елемента. Три превучене лабеле аутоматски добијају имена *Label1*, *Label2*, *Label3*. Име елемента је јединствено. Не могу у истој апликацији да постоје два елемента која имају исту вредност у (*Name*) својству. У сваком тренутку мора бити јасно које име се на који елемент односи јер када је потребно нешто изменити програмским путем навођењем имена тог објекта се врши обраћање елементу са којим је потребно остварити комуникацију тј. извршити неку промену његових својстава или тражити да се обави нека од функционалности коју тај елемент има уграђену или програмирану.

Својство **Font** омогућава да избор породице слова, тип фонта и његову величину. За боју слова текста користи се својство **ForeColor**. Избор палете боја идентичан је оном код форме. На пример, ако би се променила величина на 14px, а за боју фонта из *Web* палете изабрала боја *Dodger Blue* добио би се изглед лабеле као на слици 8.



Слика 8. Предефинисана лабела за корисничка подешавања

Лабели постављеној на форму није могуће променити величину зато што је подразумевано укључено својство *AutoSize*. Ако је потребно променити величину лабеле треба искључити ово својство, тј. изабрати вредност *False*, потом је могуће променити величину лабеле да одговара захтевима задатка тј. потребама корисника. Текст у оквиру контејнера лабеле може се позиционирати помоћу својства лабеле *TextAlign*. На пример својство *AutoSize* постављено је на *False*, својство *TextAlign* постављено је на *MiddleCenter*, а *BackColor* на *LightYellow*. Резултат ових промена приказан је на слици 9.



Слика 9. Лабела са искљученим *AutoSize* својством

Поље за унос текста

Поље за унос текста (*TextBox*) је елемент који дозвољава корисницима да уносе текстуални запис. Налази се у групи елемената секције *Common Tools* и као и лабела користи се као дете



елемент. Има слична својства као и лабела, па ће у наставку бити речи о својствима која су јединствена за овај елемент.

Multiline својство одређује да ли ће бити једнолинијских или вишелинијских корисничких уноса. Уколико је вредност *True* унос је вишелинијски, уколико не унос је једнолинијски. **Lines** својство нам омогућава да се унесе предефинисани вишелинијски текст ако је *Multiline* својство активно. У супротном свака линија ће бити спојена на претходну.

ReadOnly својство онемогућава корисника да унесе текст у поље, текст је само за читање.

Scrollbars својство служи за додавање хоризонталног и/или вертикалног клизача. Ово својство се користи када је својство *Multiline* активно.

Контејнер за слику

Контејнер за слику (*PictureBox*) служи за додавање слике у апликацију. Од својстава која су значајна, три су везана за сам ресурс слике: *Initial image*, *Image* и *Error Image*. **Initial image** је иницијална слика која се приказује док се права слика учитава. Препорука је да се за ову слику користи GIF слика *spinner*-а (слика 10).

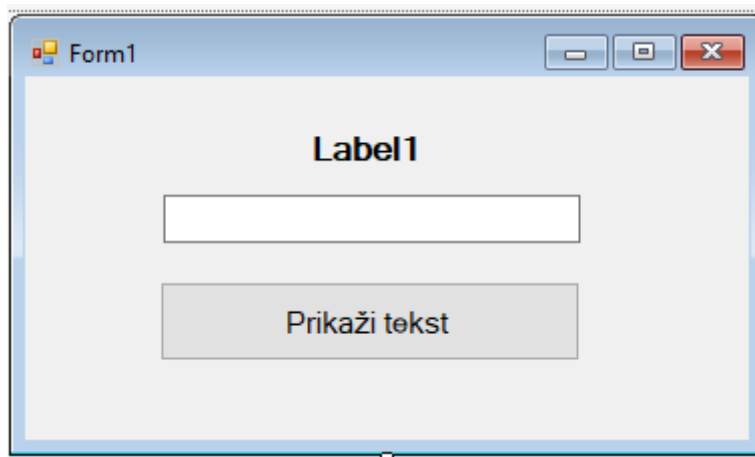


Слика 10. Пример *spinner*-а за *Initial image* својство

Својство **Image** представља слику коју треба приказати, док својство **Error Image** приказује слику која се отвара када жељена слика не може да се прочита.

Догађаји и динамичка измена својстава

У претходном тексту је описано како се коришћењем *Properties* прозора могу мењати својства, међутим, често је потребно да се у току извршавања програма промени неко својство графичког елемента то се постиже писањем кода. Нека је дата форма приказана на слици 11.





Слика 11. Пример форме са три графичка елемента

У циљу реализације комуникације између корисника и апликације битно је знати како функционише ток извршања *event-driven* апликација. **Event-driven** апликације су апликације где се интеракција реализује извршавањем догађаја. Сваки догађај дефинисан је следећим током:

1. **Корисник покрене догађај над апликацијом или апликација креира сама догађај.** Догађај који корисник може да покрене над апликацијом је на пример клик на дугме или прелаз миша преко лабеле. Догађај који апликација може да изазове или креира над собом је на пример измена података када се апликација покрене или извршавање неке радње у задатом временском интервалу.
2. **Догађај утиче на елемент/е.** На пример када корисник кликне на дугме, лабела промени боју. Такође је могуће да се вредност својства једног елемента додели својству другог елемента. На пример из поља за унос текста, унети текст корисника доделимо другом елементу на пример лабели.
3. **Корисник добија неку врсту повратне информације.** Корисник види визуелно измене или добија неку врсту поруке о томе да је догађај успешно реализован.

Именовање догађаја и елемената

У наведеном току извршавања, догађај мора бити прецизно дефинисан. Да би се догађај реализовао онако како је програмер замислио, потребно је знати јасне одговоре на питања:

1. Који је елемент захтевао догађај?
2. Који је догађај позван?
3. Који је елемент (ако се извршавају промене над елементом) измењен. Који је исход написаног кода?
4. Како корисника обавестити о изменама насталим позивањем догађаја?

Управо одговори на претходно наведена питања захтевају давање јединствених имена елементима и догађајима, па би било добро водити рачуна о следећем:

1. Када се елемент превуче на форму он добија подразумевано име. За дугме то ће бити *Button1*, следеће дугме биће *Button2* итд., за лабелу *Label1*, *Label2*, за поље за унос текста *TextBox1*, *TextBox2*. Проблем овог именовања које подразумевано даје *Visual Studio* је да на форми може постојати више десетина елемената који су дугмад или лабеле или поља за унос текста и са подразумеваним именима је тешко наћи жењени елемент, поготово и сложеним апликацијама.
2. Елементима треба дати смислено име које га описује. То име треба да прати следећа правила:
 - Име елемента мора бити јединствено и не може на истој форми да постоји други елемент или догађај са датим именом.
 - Име елемента мора да садржи само слова и бројеве. Име не може да има размак или друге специјалне знакове.

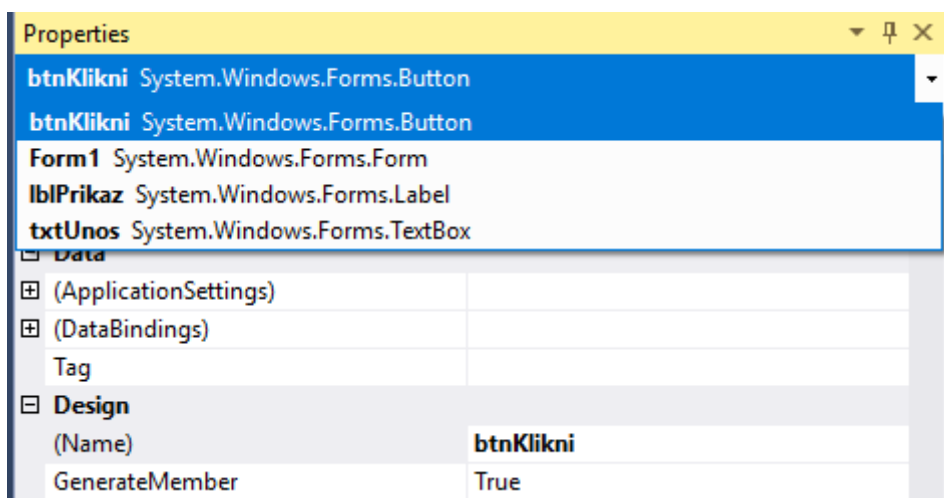


- Име елемента не може да почиње са бројем. Име елемента мора почети са малим или великом словом.
- Уколико је елемент графички елемент, препорука је да се испред елемента поставе одговарајући префикси. Листа префикса за неке учестале елементе дата је у табели 1.

Графички елемент	Префикс	Пример
Button	btn	btnOpen
Check box	chk	chkReadOnly
Combo box, drop-down list box	cbo	cboEnglish
Command button	cmd	cmdExit
Common dialog	dlg	dlgFileOpen
Date picker	dtp	dtpPublished
Form	frm	frmEntry
Frame	fra	fraLanguage
Image	img	imgIcon
ImageList	ils	ilsAllIcons
Label	lbl	lblHelpMessage
List box	lst	lstPolicyCodes
Menu	mnu	mnuFileOpen
Option button	opt	optGender
Picture box	pic	picVGA
ProgressBar	prg	prgLoadFile
RichTextBox	rtf	rtfReport
StatusBar	sta	staDateTime
TabStrip	tab	tabOptions
Text box	txt	txtLastName
Timer	tmr	tmrAlarm

Табела 1. Учестали префикси и примери за именовање елемената

У претходно приказаном примеру. Лабели са текстом *Label1* према конвенцији треба дати име *lblPrikaz*, пољу за унос текста *txtUnos* а дугмету *btnKlikni*. У *Properties* прозору потребно је изменити (*Name*) својство. У падајућој листи испод *Properties* заглавља могу се видети новоименовани елементи (слика 12).



Слика 12. Падајућа листа са новоименованим елементима

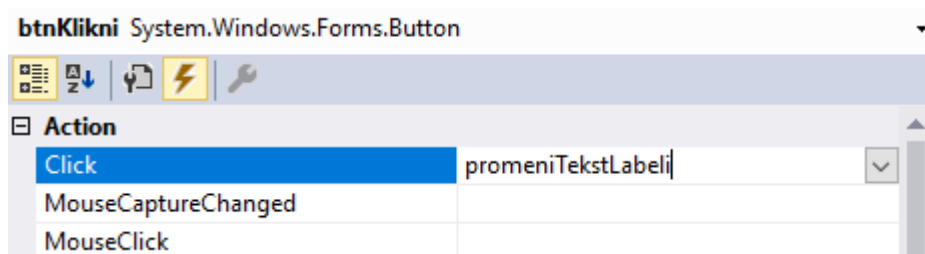
Дефинисање основних догађаја



Задатак: Креирати форму са слике 11. Када корисник унесе текст у поље за унос текста, кликом на дугме потребно је да се текст у лабели са текстом *Label1* замени са текстом који је унео корисник.

Из поставке задатка је познат одговор на питање „Који је елемент позвао догађај?“ то дугме на које је корисник кликнуо - *btnKlikni*. Одговор на питање „Који је догађај позван?“. Позван је догађај клик дугмета, који програмер треба да напише.

У *Properties* прозору налази се иконица за догађаје (⚡). Кликом на њу појавиће се листа догађаја које елемент може реалзовати. Догађај који је потребно креирати је везан за акцију *Click*. Из листе понуђених догађаја за дугме треба изабрати овај догађај и попунити поље десно од имена догађаја које представља назив догађаја. Догађаје је пожељно смислено именовати тако да описују радњу коју догађај треба да уради. За именовање догађаја важе иста правила као и за именовање графичких елемената. Такође, догађај и било који графички елемент на форми не смеју имати исто име. Име догађаја у решењу постављеног задатка биће *promeniTekstLabeli* (слика 13).



Слика 13. Давање имена догађаја за клик акцију дугмета „btnKlikni“

Отвара се едитор за уређивање кода (*Code Editor*) са иницијалним кодом за догађај. Може се приметити да овом догађају није дата никаква функционалност, што се може видети на слици 14, где је жутом бојом обележен код који представља почетак и крај догађаја. Између ових линија програмер пише делове кода којим реализује жељени догађај.



0 references

```
Private Sub promeniTekstLabel1(sender As Object, e As EventArgs)  
    naredbe koje se izvršavaju za događaj  
End Sub
```

Слика 14. Код генерисан за догађај „promeniTekstLabel1“

Следећи корак је написати код који кореспондира одговору. Елемент који ће бити измењен је *lblPrikaz* а елемент од ког ће бити узета вредност за измену је *txtUnos*. Кориснику није потребно приказати обавештење о измени јер ће визуелна промена бити индикатор да се догађај реализовао.

Када је потребно приступити неком својству објекта користи се следећа синтакса: **име објекта.име својства**. На пример *Text* својству елемента *lblPrikaz* приступа се писањем *lblPrikaz.Text*. Приступање својству боје фонта - *lblPrikaz.ForeColor*.

У демонстрационом примеру потребно је променити *Text* својство лабеле *lblPrikaz* тако што ће бити додељена вредност *Text* својства поља за унос текста *txtUnos*. Када је потребно изменити својство, односно доделити му неку вредност мора се користити **оператор доделе** знак = (једнако). Наредба за доделу има формат име објекта **име објекта.име својства = вредност**. Вредност може бити или својство од неког другог објекта или директна вредност у коду (хардкодowana, *hardcoded*).

Решење примера је дато следећим кодом.

```
1. Public Sub promeniTekstLabel1(sender As Object, e As EventArgs) Handles  
   btnKlikni.Click  
2.     lblPrikaz.Text = txtUnos.Text  
3. End Sub
```

Комбинацијом тастера *Ctrl* и *F5* покреће се креирана апликација.

Корисно је напоменути да је за повратак у режим за дизајн пречица на тастатури комбинаца тастера *Shift* и *F7*. За прелазак из режима дизајна назад у едитор користи се тастер *F7*.