

Korišćenje teksta i vidžeta unosa i menadžera mreže prikaza (grid layout manager)

U GUI programiranju, potrebno je i da korisnik unese neki tekst. Ponekad je potrebno da se prikaže tekst korisniku. Za oba slučaja, koriste se vidžeti za rad sa tekstom. Postoje dve vrste: vidžeti unosa (entry widget) se koriste za jednu liniju teksta dok tekst vidžeti se koriste za veći broj linija teksta. Može se čitati sadržaj oba za dobijanje korisničkog ulaza. Takođe se može uneti tekst u njih čime se daju informacije korisniku.

Kada se jednom unose velike količine vidžeta na ekran, potrebno je da se oni organizuju. U dosadašnjim programima sa koristio Grid menadžer prikaza, ali veoma limitirano. Grid prikaz pruža mnogo više kontrole u načinu na koji GUI izgleda. Menadžer dozvoljava smeštanje vidžeta na posebne lokacije pri čemu tretira okvir kao mrežu (grid).

program Dugovecnost.py

Ovaj program otkriva tajnu dugog života do 100 godina, ako korisnik unese tajnu lozinku („tajna“). Korisnik unosi lozinku kroz tekst unos a zatim klikne na dugme. Ako je lozinka tačna, program prikazuje tajnu dugovečnosti u tekst boksu:

```
#Dugovecnost
#prikazuje tekst vidzete unosa i menadzer mreznog prikaza
from tkinter import *
class Aplikacija(Frame):
    """ GUI aplikacija koja moze otkriti tajnu dugovecnosti. """
    def __init__(self, master):
        """ Inicijalizuje okvir. """
        super(Aplikacija, self).__init__(master)
        self.grid()
        self.kreira_vidzete()

    def kreira_vidzete(self):
        """ Kreira dugme, tekst i vidzete unosa. """
        #kreira lejbel instrukcija
        self.lejbel1 = Label(self, text = "Unesi lozinku za tajnu dugovecnosti")
        self.lejbel1.grid(row = 0, column = 0, columnspan = 2, sticky = W)
        #kreira lejbel za lozinke
        self.lejbel2 = Label(self, text = "Lozinka: ")
        self.lejbel2.grid(row = 1, column = 0, sticky = W)
        #kreira vidzet unosa za prihvatanje lozinke
        self.unos = Entry(self)
        self.unos.grid(row = 1, column = 1, sticky = W)
        #kreira dugme za slanje podataka
        self.slanje = Button(self, text = "Slanje", command = self.otkrivanje)
        self.slanje.grid(row = 2, column = 0, sticky = W)
        #kreira tekst vidzeta za prikaz poruke
        self.tajni_tekst = Text(self, width = 35, height = 5, wrap = WORD)
        self.tajni_tekst.grid(row = 3, column = 0, columnspan = 2, sticky = W)

    def otkrivanje(self):
        """ Prikaz poruke u zavisnosti od lozinke. """
        sadrzaj = self.unos.get()
        if sadrzaj == "tajna":
            poruka = "Ovo je tajna doziveti 100 godina: dozivi 99 a onda budi VEOMA oprezan. "
        else:
            poruka = "To nije tacna lozinka, pa nema otkrivanja tajne za tebe."
```

```
self.tajni_tekst.delete(0.0, END)
self.tajni_tekst.insert(0.0, poruka)
```

```
#main
osnovni = Tk()
osnovni.title("Dugovecnost")
osnovni.geometry("300x150")
okvir = Aplikacija(osnovni)
osnovni.mainloop()
```

001 Postavka programa

```
#Dugovecnost
#prikazuje tekst vidzete unosa i menadzer mreznog prikaza
from tkinter import *
class Aplikacija(Frame):
    """ GUI aplikacija koja moze otkriti tajnu dugovecnosti. """
    def __init__(self, master):
        """ Inicijalizuje okvir. """
        super(Aplikacija, self).__init__(master)
        self.grid()
        self.kreira_vidzete()
```

Importuje se **tkinter** modul i definiše klasa **Aplikacija**. U metodi konstruktora, inicijalizuje se nov objekat **Aplikacija**, postavi da bude vidljiv i podigne objektova metoda **kreira_vidzete()**.

002 Postavljanje vidžeta sa menadžerom mrežnog prikaza

Startuje se metoda **kreira_vidzete()** i kreira lejbela koji daje instrukcije korisniku:

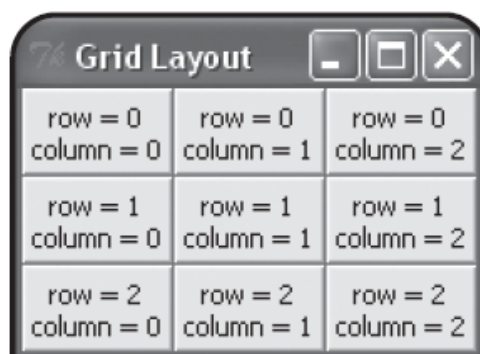
```
def kreira_vidzete(self):
    """ Kreira dugme, tekst i vidzete unosa. """
    #kreira lejbela instrukcija
    self.lejbel1 = Label(self, text = "Unesi lozinku za tajnu dugovecnosti")
```

U sledećoj liniji, koristi se Grid menadžer prikaza da bi se specificirala pozicija lejbela:

```
self.lejbel1.grid(row = 0, column = 0, columnspan = 2, sticky = W)
```

Vidžet objektova metoda **grid()** može primiti vrednosti za rezne parametre, ali ovde se koriste samo četiri: **row** (red), **column** (kolona), **columnspan** i **sticky**.

Parametri **row** i **column** uzimaju cele brojeve i definišu gde je smešten objekat unutar njegovog master vidžeta. U ovom programu, može se zamisliti okvir u osnovnom prozoru kao mreža, podeljena na redove i kolone. U svakom preseku reda i kolone je ćelija, gde se može smestiti vidžet. Na slici je primer devet smeštenih **Button** vidžeta, u devet različitih ćelija, korišćenjem brojeva za redove i kolone.



row = 0 column = 0	row = 0 column = 1	row = 0 column = 2
row = 1 column = 0	row = 1 column = 1	row = 1 column = 2
row = 2 column = 0	row = 2 column = 1	row = 2 column = 2

Za **Label** vidžet u kodu, date su 0 za row i 0 za column, što smešta lejbela na gornji levi ugao okvira.

Ako je vidžet veoma širok, možda je bolje omogućiti vidžetu da proširi (**span**) više od jedne ćelije tako da drugi vidžeti budu korektno smešteni. Parametar **columnspan** omogućava proširenje vidžeta preko više od jedne kolone. Dat je parametar 2 kojim se omogućava

dugačka lejbel proširen na dve kolone. To znači da lejbel zauzima dve ćelije, jednu u redu 0 kolona 0 a drugu u redu 0 kolona 1 (takođe se može koristiti parametar **rowspan** kojim se omogućava vidžetu da se proširi na više od jednog reda).

Čak i kada je uređeno koje ćelije vidžet zauzima, postoji fleksibilnost uređenja vidžeta unutar ćelija upotrebom parametra **sticky**, koji uzima smerove kao vrednosti, uključujući N, S, E i W. Vidžet se pomera po kvadrantu ćelije prema ovim smerovima. Pošto je dat W za **sticky** objektu Label, lejbel je smešten prema zapadu (levo). Drugi način da se ovo objasni: lejbel je uređen prema levom kraju (left-justified) ćelije.

Zatim, kreira se lejbel koji se pojavljuje u sledećem redu, levo-uređen:

```
#kreira lejbel za lozinke
self.lejbel2 = Label(self, text = "Lozinka: ")
self.lejbel2.grid(row = 1, column = 0, sticky = W)
```

003 Kreiranje vidžeta unosa

```
#kreira vidžet unosa za prihvatanje lozinke
self.unos = Entry(self)
```

Kod kreira tekst unosa gde korisnik može uneti lozinku.

Pozicija **Entry** vidžeta je u ćeliji pored lejbela sa lozinkom:

```
self.unos.grid(row = 1, column = 1, sticky = W)
```

Zatim, kreira se dugme koje omogućava korisniku unos lozinke:

```
#kreira dugme za slanje podataka
self.slanje = Button(self, text = "Slanje", command = self.otkrivanje)
```

Povezana je aktivacija dugmeta sa metodom **otkrivanje()**, koja otkriva tajnu dugovečnosti, ako je korisnik uneo tačnu lozinku.

U sledećem redu je smešteno dugme, skroz levo:

```
self.slanje.grid(row = 2, column = 0, sticky = W)
```

004 Kreiranje tekst vidžeta

Kreira se nov tip vidžeta, **Text** vidžet:

```
self.tajni_tekst = Text(self, width = 35, height = 5, wrap = WORD)
```

Dodaju se vrednosti za **width** i **height** za postavljanje dimenzija tekst boksa. Zatim se dodaju vrednosti za parametar **wrap**, koji odlučuje kako je prelomljen tekst u boksu. Moguće vrednosti za parametre su WORD, CHAR i NONE. WORD čini da se reči prelamaju (pri dostizanju desne ivice boksa), CHAR čini da se prelom radi između slova, dok NONE znači da nema preloma. To znači da se može pisati tekst samo na prvoj liniji tekst boksa.

Zatim, postavi se tekst boks tako da je u sledećem redu i proširen na dve kolone:

```
self.tajni_tekst.grid(row = 3, column = 0, columnspan = 2, sticky = W)
```

005 Dobijanje i unos teksta sa vidžetima baziranim na tekstu

Koristi se metodom **otkrivanje()**, koja testira da li je korisnik uneo tačnu lozinku. Ako jeste, metod prikazuje tajnu dugog života. U suprotnom, korisniku se kaže da je lozinka pogrešna.

Prvo se dobija tekst u **Entry** vidžetu pokretanjem metode **get()**:

```
def otkrivanje(self):
    """ Prikaz poruke u zavisnosti od lozinke. """
    sadrzaj = self.unos.get()
```

Metoda **get()** vraća tekst u vidžet. I **Entry** i **Text** objekti imaju **get()** metodu.

Proverava se da li je tekst identičan sa "tajna". Ako jeste, postavlja se **poruka** na string opisujući tajnu života do 100. Inače, postavlja se **poruka** na string koji govori korisniku da je lozinka pogrešna.

```
if sadrzaj == "tajna":
    poruka = "Ovo je tajna doživeti 100 godina: doživeti 99 a onda budi VEOMA
```

```

        oprezan. "
else:
    poruka = "To nije tacna lozinka, pa nema otkrivanja tajne za tebe."

```

Sada kada postoji string koji se prikazuje korisniku, treba ga uneti u Text vidžet. Prvo, obrisati tekst koji je već u Text vidžetu upotrebom `delete()` metode:

```
self.tajni_tekst.delete(0.0, END)
```

Metoda `delete()` može obrisati tekst iz vidžeta baziranom na tekstu. Metoda može uzeti jedan indeks ili početnu i krajnju tačku. Dodaje se realan broj za predstavljanje para reda i kolone gde levi broj od decimalne tačke je red a cifra sa desne strane decimalne tačke je kolona. Npr, u prošlom primeru, dodeljeno je 0.0 kao početna tačka, što znači da metod treba da izbriše tekst počevši od reda 0, kolone 0 (apsolutni početak) u tekst boks.

Modul `tkinter` omogućava nekoliko konstanti za pomoć sa ovim tipom metoda, poput `END`, koja označava kraj teksta. Tako da, prethodna linija koda briše sve od prve pozicije do kraja tekst boks. I `Text` i `Entry` vidžeti imaju `delete()` metodu.

Sledeće, ubacuje se string koji treba da se prikaže na `Text` vidžetu:

```
self.tajni_tekst.insert(0.0, poruka)
```

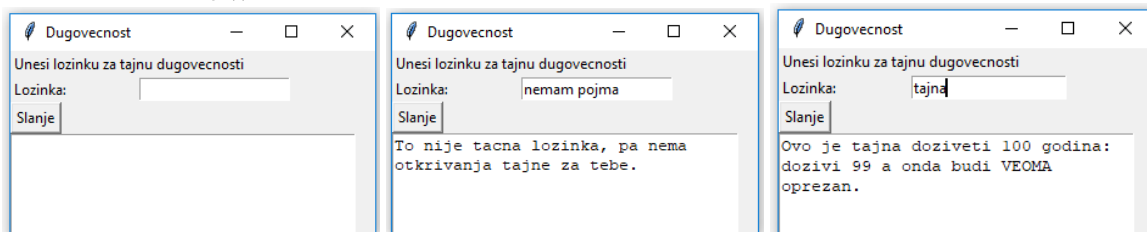
Metoda `insert()` može ubaciti string u tekst bazirani vidžet. Metoda uzima poziciju unosa i string. U pethodnoj liniji koda, dodato je 0.0 kao pozicija unosa, štp znači da metod treba da počne unos od reda 0 i kolone 0. Dodata je `poruka` kao druga vrednost, tako da odgovarajuća poruka se pojavljuje u tekst boks. I ovu metodu koriste i `Text` i `Entry` vidžeti.

Sa metodom `insert()` se ne može zameniti vidžet baziran na tekstu, već se samo unosi. Ako je potrebno zameniti postojeći tekst sa novim tekstom, prvo se poziva `delete()` metoda.

```

#main
osnovni = Tk()
osnovni.title("Dugovecnost")
osnovni.geometry("300x150")
okvir = Aplikacija(osnovni)
osnovni.mainloop()

```



Korišćenje dugmadi za potvrdu (check buttons)

Dugme za potvrdu omogućava korisniku da izabere veći broj mogućnosti iz neke predložene grupe. Iako ovo daje korisniku dosta fleksibilnosti, ovo obično daje programeru veću kontrolu limitiranjem posebne listu stvari koje korisnik može izabrati.

program Izbor_Filma.py

Ovaj program omogućava korisniku da izabere omiljene žanrove filmova iz liste: komedija, drama, romantika. Pošto program koristi dumad za potvrdu, korisnik može izabrati koliko želi od ponuđenih odgovora. Program prikazuje rezultate korisnikove selekcije u tekst boks.

```

#Izbor_Filma
#Prikazuje dugmad za potvrdu
from tkinter import *
class Aplikacija(Frame):
    """ GUI Aplikacija za omiljene filmske zanrove. """
    def __init__(self, master):
        super(Aplikacija, self).__init__(master)

```

```

self.grid()
self.kreira_vidzete()

def kreira_vidzete(self):
    """ Kreira vidzete za ponudjene tipove filmova. """
    #kreira lejbela za opis
    Label(self, text = "Izaberi omiljeni filmski zanr").grid(row = 0, column = 0,
        sticky = W)
    #kreira lejbela instrukcija
    Label(self, text = "Izaberi sve sto odgovara: ").grid(row = 1, column = 0,
        sticky = W)
    #kreira dugme za potvrdu komedije
    self.voli_komediju = BooleanVar()
    Checkbutton(self, text = "Komedija", variable = self.voli_komediju,
        command = self.updejt_teksta).grid(row = 2, column = 0, sticky = W)
    #kreira dugme za potvrdu drame
    self.voli_dramu = BooleanVar()
    Checkbutton(self, text = "Drama", variable = self.voli_dramu,
        command = self.updejt_teksta).grid(row = 3, column = 0, sticky = W)
    #kreira dugme za potvrdu romana
    self.voli_romansu = BooleanVar()
    Checkbutton(self, text = "Romansa", variable = self.voli_romansu,
        command = self.updejt_teksta).grid(row = 4, column = 0, sticky = W)
    #kreira polje teksta za prikaz rezultata
    self.rezultati = Text(self, width = 40, height = 5, wrap = WORD)
    self.rezultati.grid(row = 5, column = 0, columnspan = 3)

def updejt_teksta(self):
    """Updejtuje tekst vidzet i prikazuje korisnikove omiljene filmske zanrove. """
    voli = ""
    if self.voli_komediju.get():
        voli += "Volis filmove komedije.\n"
    if self.voli_dramu.get():
        voli += "Volis filmove drame.\n"
    if self.voli_romansu.get():
        voli += "Volis filmove romanse."
    self.rezultati.delete(0.0, END)
    self.rezultati.insert(0.0, voli)

# main
osnovni = Tk()
osnovni.title("Izbor Filma")
okvir = Aplikacija(osnovni)
osnovni.mainloop()

```

006 Postavljanje programa

Postavlja se program importovanjem **tkinter** i startovanjem definicije klase **Aplikacija**:

```

#Izbor_Filma
#Prikazuje dugmad za potvrdu
from tkinter import *
class Aplikacija(Frame):
    """ GUI Aplikacija za omiljene filmske zanrove. """
    def __init__(self, master):
        super(Aplikacija, self).__init__(master)
        self.grid()
        self.kreira_vidzete()

```

007 Omogućava vidžet masteru da bude njegova jedina referenca

Kreira se lejbela koji opisuje program:

```

def kreira_vidzete(self):
    """ Kreira vidzete za ponudjene tipove filmova. """
    #kreira lejbela za opis

```

```
Label(self, text = "Izaberi omiljeni filmski žanr").grid(row = 0, column = 0,
    sticky = W)
```

Postoji bitna razlika između ovog lejbel a prethodnih: nije dodeljen rezultujući **Label** objekat promenljivoj. Normalno, ovo bi bila velika greška, pošto ni na koji način ovakav objekat ne bi bio povezan sa programom. Ali preko modula **tkinter**, objekat **Label** jeste povezan sa programom, kao i svi GUI elementi, povezan sa svojim masterom. Ovo znači da pošto znam da mi ne treba direktan pristup vidžetu, onda nema potrebe za dodeljivanjem objekta promenljivoj. Glavna dobit ovog pristupa je kraći i čistiji kod.

U dosadašnjim programima, svaki novi vidžet je dodeljivan promenljivoj. U ovom slučaju, zna se da nema potrebe za pristupom ovom lejbelu, pa se ne dodeljuje **Label** objekat promenljivoj. Umesto toga, master će održavati jedinu referencu sa njime.

Zatim se kreira drugi lejbel:

```
#kreira lejbel instrukcija
```

```
Label(self, text = "Izaberi sve sto odgovara: ").grid(row = 1, column = 0, sticky = W)
```

Lejbel daje instrukcije, govori korisniku da može izabrati više mogućih odgovora.

008 Kreiranje dugmeta za potvrdu

Sledeće, kreira se dugme za potvrdu, jedno za svaki filmski žanr. Prvo se postavlja dugme za žanr komedije.

Svako dugme za potvrdu traži poseban objekat pridružen njime da bi automatski sadržao status dugmeta za potvrdu. Ovaj poseban objekat mora biti instanca **BooleanVar** klase iz **tkinter** modula. Tako da pre nego se kreira dugme za potvrdu žanra komedije, instancira se **BooleanVar** objekat i dodeljuje se atributu novog objekta, **voli_komediju**:

```
#kreira dugme za potvrdu komedije
```

```
self.voli_komediju = BooleanVar()
```

Bulova promenjiva je posebna vrsta promenjive koja može biti samo **true** ili **false**. Programeri često koriste ovakve promenjive i nazivaju ih samo **Boolean**. Izraz je uvek sa početnim slovom velikim pošto potiče od imena engleskog matematičara Džordža Bula.

Zatim se kreira i samo dugme za potvrdu:

```
Checkbutton(self, text = "Komedija", variable = self.voli_komediju,
    command = self.update_text).grid(row = 2, column = 0, sticky = W)
```

Kod kreiranja novog dugmeta za potvrdu sa tekstom **Komedija**. Dodavanjem **self.voli_komediju** parametru **variable**, pridružen je status dugmeta za potvrdu (izabran ili nepotvrđen) sa atributom **voli_komediju**. Dodavanjem **self.update_text()** parametru **command**, povezuje se aktivacija dugmeta za potvrdu sa metodom **update_text()**. To znači da kadgod korisnik selektuje ili izbriše unos u dugme za potvrdu, **update_text()** metod se poziva. Na kraju, smešta se dugme za potvrdu u sledeći red, skroz nalevo.

Primititi da se nije dodelio rezultujući **Checkbutton** objekat u promenljivu. Ovo je u redu, pošto ono što je zaista bitno je status dugmeta, kojim se može prići iz **voli_komedije** atributa.

Na isti način se kreiraju sledeća dva dugmeta za potvrdu:

```
#kreira dugme za potvrdu drame
```

```
self.voli_dramu = BooleanVar()
```

```
Checkbutton(self, text = "Drama", variable = self.voli_dramu,
    command = self.update_text).grid(row = 3, column = 0, sticky = W)
```

```
#kreira dugme za potvrdu romana
```

```
self.voli_romansu = BooleanVar()
```

```
Checkbutton(self, text = "Romansa", variable = self.voli_romansu,
    command = self.update_text).grid(row = 4, column = 0, sticky = W)
```

Zato, kadagod korisnik izabere ili izbriše izbor u Drama ili Romansa dugmetu za potvrdu, `update_text()` metoda se priziva. I čak iako nisu dodeljeni rezultujući `Checkbutton` objekti bilo kojoj promenljivoj, uvek se može videti status Drama dugmeta za potvrdu kroz `voli_dramu` atribut, a i može se uvek videti status Romansa dugmeta za potvrdu kroz `voli_romansu` atribut.

Na kraju, kreira se tekst boks koji se koristi za prikaz rezultata korisnikovog izbora:

```
#kreira polje teksta za prikaz rezultata
self.rezultati = Text(self, width = 40, height = 5, wrap = WORD)
self.rezultati.grid(row = 5, column = 0, columnspan = 3)
```

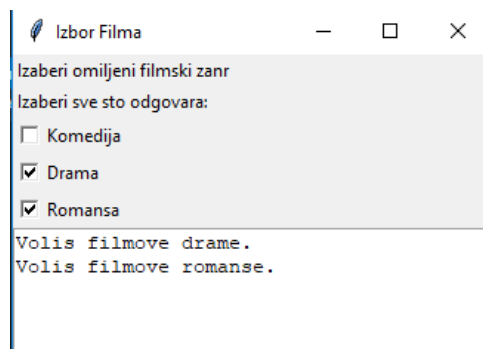
009 Dobijanje statusa dugmeta za potvrdu

Piše se `updejt_teksta()` metoda, koja updejtuje tekst boks čime se prikazuje stanje dugmadi za potvrdu koje je korisnik čekirao:

```
def updejt_teksta(self):
    """ Updejtuje tekst vidzet i prikazuje korisnikove omiljene filmske zanrove. """
    voli = ""
    if self.voli_komediju.get():
        voli += "Volis filmove komedije.\n"
    if self.voli_dramu.get():
        voli += "Volis filmove drame.\n"
    if self.voli_romansu.get():
        voli += "Volis filmove romanse."
    self.rezultati.delete(0.0, END)
    self.rezultati.insert(0.0, voli)
```

Vrednosti `BooleanVar` objekta se ne može prići direktno. Umesto toga, mora se prizvati objektova `get()` metoda. U prethodnom kodu, koristila se metoda `get()` iz `BooleanVar` objekta na koju je ukazivao `voli_komediju` da bi dobio vrednost objekta. Ako je vrednost evaluirana na `true`, to znači da dugme za potvrdu Komedije je selektovano, pa se dodaje string `"Volis filmove komedije.\n"` na string koji se gradi za prikaz u tekst boks. Slične operacije se izvode u zaisnosti od stanja druga dva dugmeta za potvrdu. Na kraju, briše se sav tekst iz tekst boksa pa se unosi sledeći string, `voli`.

```
# main
osnovni = Tk()
osnovni.title("Izbor Filma")
okvir = Aplikacija(osnovni)
osnovni.mainloop()
```



Korišćenje okruglih dugmadi (radio buttons)

Okrugla dugmad liče na dugmad za potvrdu, osim što oni omogućavaju samo da se izabere jedno dugme iz grupe ponuđenih odgovora. Ovo je dobro ako se traže jedna odluka od korisnika iz grupe mogućnosti.

program Izbor_Filma_2.py

Ovaj program je sličan sa prethodnom verzijom i omogućava korisniku da izabere jedan od omiljenih filmskih žanrova iz ponuđene grupe.

```
#Izbor_Filma_2
#prikazuje okruglu dumad
from tkinter import *
class Aplikacija(Frame):
    """ GUI Aplikacija za omiljeni filmski zanr. """
    def __init__(self, master):
        """ Inicijalizuje Frame. """
        super(Aplikacija, self).__init__(master)
        self.grid()
        self.kreira_vidzete()

    def kreira_vidzete(self):
        """ Kreira vidzete za izabrani filmski zanr. """
        #kreira lejbela opisa
        Label(self, text = "Izaberi omiljeni filmski zanr").grid(row = 0, column = 0,
            sticky = W)
        #kreira lejbela instrukcija
        Label(self, text = "Izaberi jedan:").grid(row = 1, column = 0, sticky = W)
        #kreira promenljivu za jedan, omiljeni filmski zanr
        self.omiljen = StringVar()
        self.omiljen.set(None)
        #kreira Komedija okruglo dugme
        Radiobutton(self, text = "Komedija", variable = self.omiljen,
            value = "komedija.", command = self.updejt_teksta).grid(row = 2,
            column = 0, sticky = W)
        #kreira Drama okruglo dugme
        Radiobutton(self, text = "Drama", variable = self.omiljen, value = "drama.",
            command = self.updejt_teksta).grid(row = 3, column = 0, sticky = W)
        #kreira Romansa okruglo dugme
        Radiobutton(self, text = "Romansa", variable = self.omiljen,
            value = "romansa.", command = self.updejt_teksta).grid(row = 4,
            column = 0, sticky = W)
        #kreira polje teksta za prikaz rezultata
        self.rezultati = Text(self, width = 40, height = 5, wrap = WORD)
        self.rezultati.grid(row = 5, column = 0, columnspan = 3)

    def updejt_teksta(self):
        """ Updejtuj tekst oblast i prikazi korisnikov omiljeni filmski zanr. """
        poruka = "Tvoj omiljeni filmski zanr je "
        poruka += self.omiljen.get()
        self.rezultati.delete(0.0, END)
        self.rezultati.insert(0.0, poruka)

#main
osnovni = Tk()
osnovni.title("Izbor Filma 2")
okvir = Aplikacija(osnovni)
osnovni.mainloop()
```

010 Kreiranje okruglih dugmeta

Pošto samo jedno okruglo dugme može biti selektovano u jednom momentu, nema potrebe da svako okruglo dugme ima posebnu promenljivu za status, kao što se traži kod dugmadi za potvrdu. Umesto toga, grupa okruglih dugmadi deli jedan, poseban objekat koji prikazuje koje je okrugle dugme izabrano. Ovaj objekat može biti instanca **StringVar** klase iz tkinter modula, koji omogućava da se string sačuva i pozove. Tako, pre nego se kreiraju sama okrugla dugmad, kreira se **StringVar** objekat koji će sva okrugla dugmad da dele, dodeljujući ga atributu

omiljen, i postavljajući njegove inicijalne vrednosti na **None** korišćenjem objektove metode **set()**:

```
#kreira promenjivu za jedan, omiljeni filmski zanr
self.omiljen = StringVar()
self.omiljen.set(None)
```

Zatim se kreira okruglo dugme Komedija:

```
#kreira Komedija okruglo dugme
Radiobutton(self, text = "Komedija", variable = self.omiljen, value = "komedija.",
            command = self.updejt_teksta).grid(row = 2, column = 0, sticky = W)
```

Opcija **variable** okruglog dugmeta definiše specijalnu promenjivu dodeljenu sa okruglim dugmetom, dok opcija **value** okruglog dugmeta definiše da se vrednost smesti u posebnu promenjivu kada se izabere okruglo dugme. Zato, postavljanjem opcije **variable** okruglog dugmeta na **self.omiljeno** a njene opcije **value** na „komedija.“, to znači da kada se izabere Komedija okruglo dugme, **StringVar** postaje referentan na **self.omiljeno** i čuva string „komedija“.

Isto važi i za druga dva okrugla dugmeta.

011 Dobijanje vrednosti iz grupe okruglih dugmadi

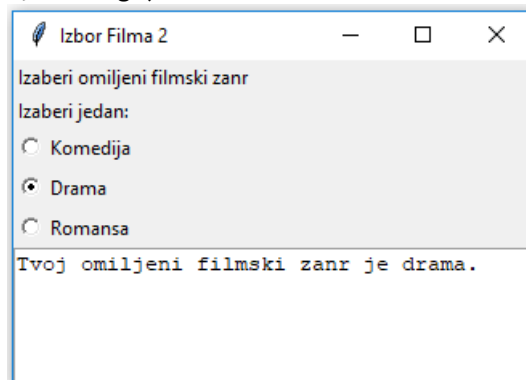
Dobijanje vrednosti iz grupe okruglih dugmadi je pozivanjem **get()** metode **StringVar** objekta kojeg svi dele:

```
def update_text(self):
    """ Updejtuj tekst oblast i prikazi korisnikov omiljeni filmski zanr. """
    poruka = "Tvoj omiljeni filmski zanr je "
    poruka += self.omiljen.get()
```

Kada se izabere Komedija okruglo dugme, **self.omiljen.get()** vraća „komedija.“.

Briše se bilo koji tekst koji je ostao u tekst boks i unosi se upravo kreirani string, koji deklariše koji je omiljeni filmski žanr korisnika:

```
self.rezultati.delete(0.0, END)
self.rezultati.insert(0.0, message)
```



program Luda_Prica.py

```
#Luda_Prica
#Kreira pricu baziranu na korisnikovom unosu
from tkinter import *
class Aplikacija(Frame):
    """ GUI aplikacija koja kreira pricu zasnovanu na korisnickom unosu. """
    def __init__(self, master):
        """ Inicijalni Okvir. """
        super(Aplikacija, self).__init__(master)
        self.grid()
        self.kreira_vidzete()

    def kreira_vidzete(self):
        """ Kreira vidzete za dobijanje informacija za pricu i za prikaz price. """
```

```

#kreira lejbhel instrukcija
Label(self, text = "Unesi informaciju za novu pricu").grid(row = 0, column =
0, columnspan = 2, sticky = W)
#kreira lejbhel i tekst unos za ime osobe
Label(self, text = "Osoba: ").grid(row = 1, column = 0, sticky = W)
self.osoba = Entry(self)
self.osoba.grid(row = 1, column = 1, sticky = W)
#kreira lejbhel i tekst unos za imenicu u mnozini
Label(self, text = "Imenica u mnozini:").grid(row = 2, column = 0, sticky = W)
self.imenica = Entry(self)
self.imenica.grid(row = 2, column = 1, sticky = W)
#kreira lejbhel i tekst unos za glagol
Label(self, text = "Glagol:").grid(row = 3, column = 0, sticky = W)
self.glagol = Entry(self)
self.glagol.grid(row = 3, column = 1, sticky = W)
#kreira lejbhel za dugmad za potvrdu prideva
Label(self, text = "Pridev-i:").grid(row = 4, column = 0, sticky = W)
#kreira dugme za potvrdu svraba
self.svrbi = BooleanVar()
Checkbutton(self, text = "svrbece", variable = self.svrbi).grid(row = 4,
column = 1, sticky = W)
#kreira dugme za proveru radosti
self.radostan = BooleanVar()
Checkbutton(self, text = "radosno", variable = self.radostan).grid(row = 4,
column = 2, sticky = W)
#kreira dugme za proveru naelektrisan
self.naelektrisan = BooleanVar()
Checkbutton(self, text = "naelektrisano", variable =
self.naelektrisan).grid(row = 4, column = 3, sticky = W)
#kreira lejbhel za okruglo dugme delova tela
Label(self, text = "Delovi Tela:").grid(row = 5, column = 0, sticky = W)
#kreira promenjivu za jedan deo tela
self.deo_tela = StringVar()
self.deo_tela.set(None)
#kreira okrugla dugmad delovi tela
delovi_tela = ["pupak", "veliko stopalo", "kvrgavo celo"]
column = 1
for deo in delovi_tela:
    Radiobutton(self, text = deo, variable = self.deo_tela, value =
deo).grid(row = 5, column = column, sticky = W)
    column += 1
#kreira dugme za zavrsetak rada
Button(self, text = "Klik za kreiranje price", command =
self.pricaj_pricu).grid(row = 6, column = 0, sticky = W)
self.prica_tekst = Text(self, width = 75, height = 10, wrap = WORD)
self.prica_tekst.grid(row = 7, column = 0, columnspan = 4)

def pricaj_pricu(self):
    """ Popunjavanje teksta boks sa novom pricom zasnovanoj na unosu korisnika. """
    #dobija vrednosti iz GUI
    persona = self.osoba.get()
    imen = self.imenica.get()
    glag = self.glagol.get()
    prid = ""
    if self.svrbi.get():
        prid += "svrbi, "
    if self.radostan.get():
        prid += "radostan, "
    if self.naelektrisan.get():
        prid += "naelektrisan, "
    deo_tela = self.deo_tela.get()

```

```

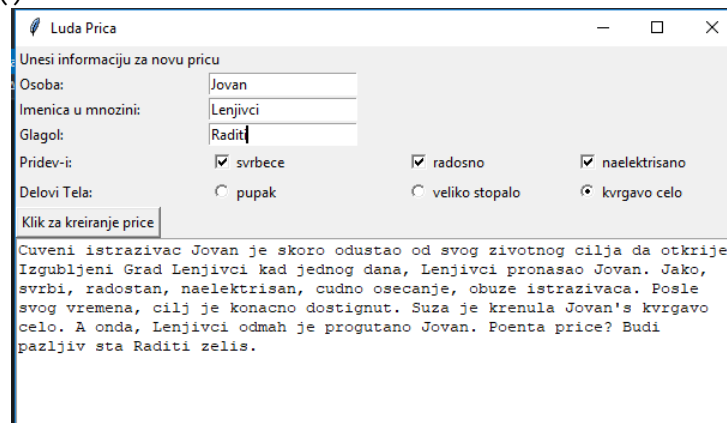
#kreira pricu
story = "Cuveni istrazivac "
story += persona
story += " je skoro odustao od svog zivotnog cilja da otkrije Izgubljeni Grad
"

story += imen.title()
story += " kad jednog dana, "
story += imen
story += " pronasao "
story += persona + ". "
story += "Jako, "
story += prid
story += "cudno osecanje, obuze istrazivaca. "
story += "Posle svog vremena, cilj je konacno dostignut. Suza je krenula "
story += persona + "'s "
story += deo_tela + ". "
story += "A onda, "
story += imen
story += " odmah je progutano "
story += persona + ". "
story += "Poenta price? Budi pazljiv sta "
story += glag
story += " zelis."

#prikaz price
self.prica_tekst.delete(0.0, END)
self.prica_tekst.insert(0.0, story)

# main
osnovni = Tk()
osnovni.title("Luda Prica")
okvir = Aplikacija(osnovni)
osnovni.mainloop()

```



Luda Prica

Unesi informaciju za novu pricu

Osoba: Jovan

Imenica u množini: Lenjivci

Glagol: Raditi

Pridev-i: ☒ svrbece ☒ radosno ☒ naelektrisano

Delovi Tela: ☐ pupak ☐ veliko stopalo ☒ kvrgavo celo

Klik za kreiranje price

Cuveni istrazivac Jovan je skoro odustao od svog zivotnog cilja da otkrije Izgubljeni Grad Lenjivci kad jednog dana, Lenjivci pronasao Jovan. Jako, svrbi, radostan, naelektrisan, cudno osecanje, obuze istrazivaca. Posle svog vremena, cilj je konacno dostignut. Suza je krenula Jovan's kvrgavo celo. A onda, Lenjivci odmah je progutano Jovan. Poenta price? Budi pazljiv sta Raditi zelis.