

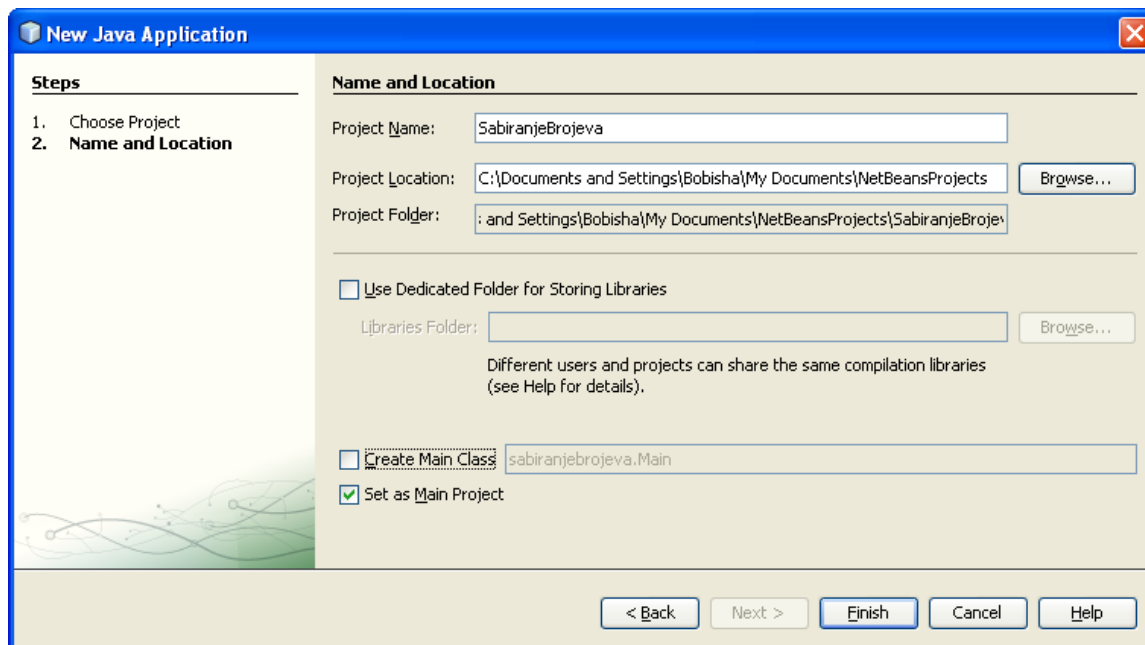


Primenjeno programiranje

Java i NetBeans IDE 7.0 – vežbe 8-9-10

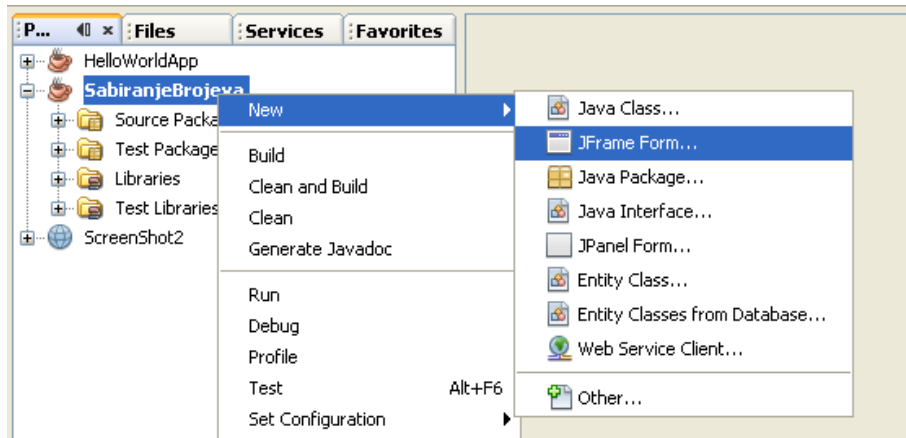
Kreiranje korisničkog interfejsa desktop aplikacije

1. Odabrati opciju *File > New Project*
2. Među kategorijama odabrati Java i podkategoriju *Java Application*
3. Dati naziv projektu, i podesiti opcije kao na slici:

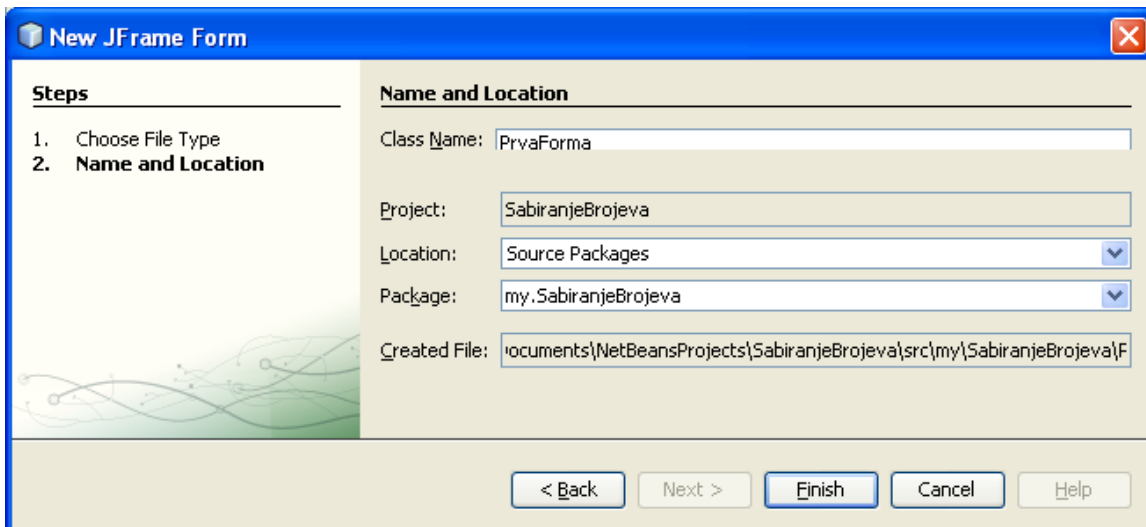


Kao početak izgradnje interfejsa potrebno je kreirati jedan kontejner koji će sadržati elemente koje nameravamo da dodamo na interfejs, kreiramo *JFrame* kontejner:

1. Odabrati opciju *New > JFrame Form*



2. Dati naziv formi i definisati paket:



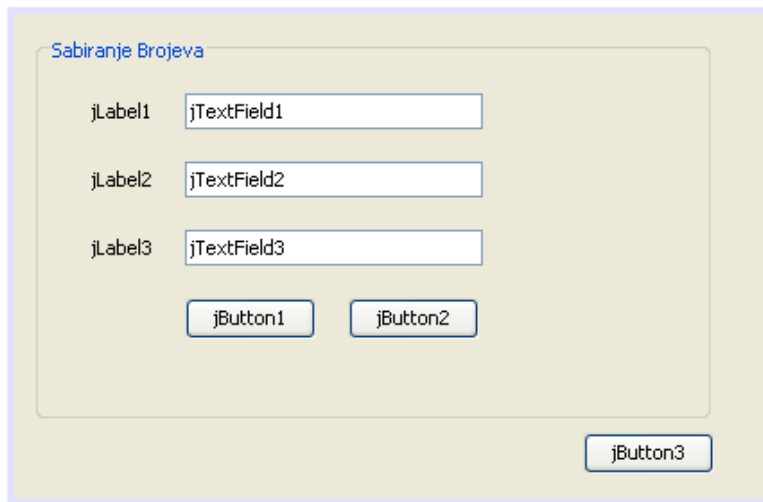
3. Kreira se klasa *PrvaForma*, unutar paketa *my.SabiranjeBrojeva*, otvara se prozor za dizajn forme i sa desne strane odgovarajuća paleta.

Zadatak 1:

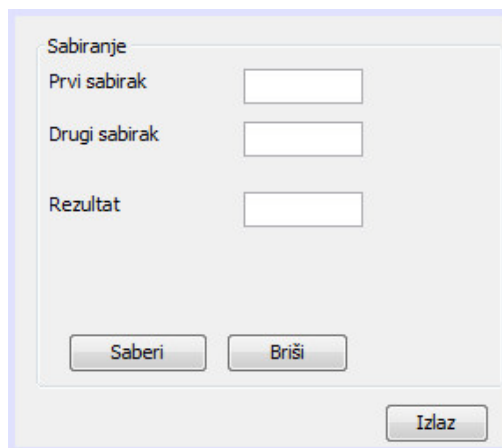
Na kreiranu formu dodati dugmad i polja sa palete, jedan *JPanel* a zatim dodati i tri labele (*JLabel*), tri polja za unos teksta (*JTextField*), i tri dugmeta (*JButton*) kao na slici:

Postavljanje naslova u *JPanel*-u:

1. Postaviti *JPanel* na formu
2. u *Properties* odabrati polje *Border* i zatim *Titled Border*, i uneti naslov panela u polje *Title*



4. Preimenovati komponente tako da izgledaju kao na slici:



Dodavanje funkcije dugmadima

1. Desnim klikom na dugme *Izadi* iz konteksnog menija odabrati *Events -> Action -> ActionPerformed*
2. U metodi koju *NetBeans* kreira zameniti red:

```
// TODO add your handling code here:
```

Sa:

```
System.exit(0);
```

3. istim postupkom na dugme *Briši* kod zameniti sa:

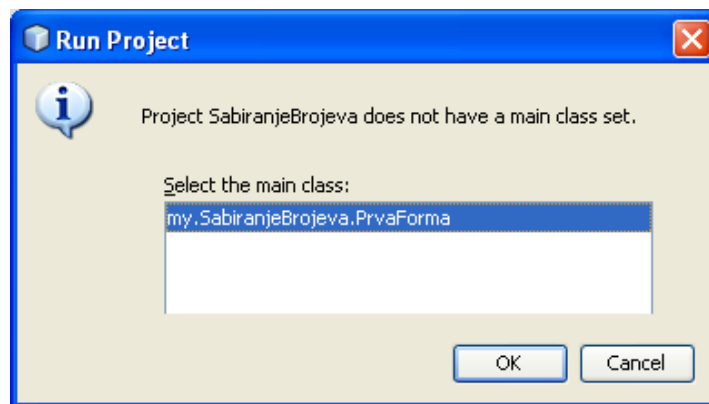
```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");
```

4. ovaj kod briše sadržaj tri polja.
5. Za dugme *Saberi* na isti način uneti

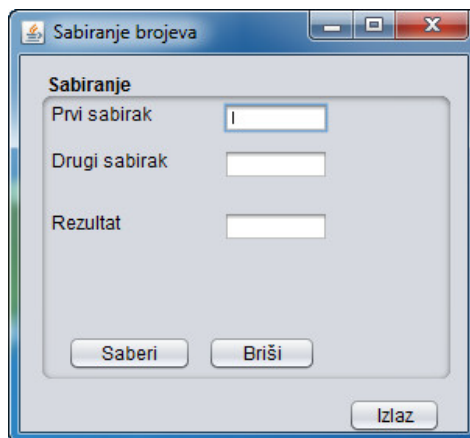
```
float num1, num2, result;  
num1 = Float.parseFloat(jTextField1.getText());  
num2 = Float.parseFloat(jTextField2.getText());  
result = num1+num2;  
jTextField3.setText(String.valueOf(result));
```

Pokretanje programa:

1. Odabрати *Run > Run Main Project*
2. Ako upozori da projekat nema glavnu klasu, odabрати je iz liste:



3. Rezultat izvođenja je:



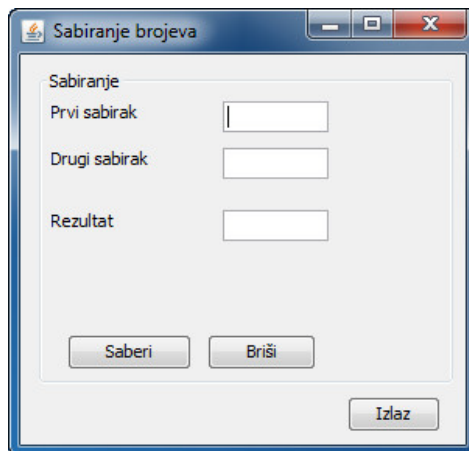
4. Ako želite *Windows Look and Feel* u main klasu uneti naredni kod:

```

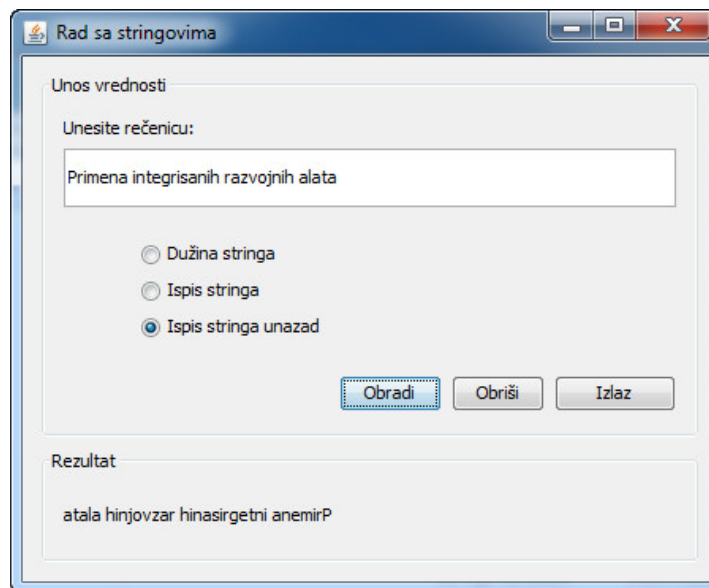
try {
    UIManager.setLookAndFeel(
        UIManager.getSystemLookAndFeelClassName());
} catch (Exception ex) {
    ex.printStackTrace();
}

```

5. Pa će prozor izgledati:



Zadatak 2. Programirati aplikaciju koja će imati sledeći korisnički interfejs:



- Pritiskom na dugme **Obradi** se obavljaju sledeće akcije u zavisnosti koje radi dugme je obeleženo:
 - Dužina stringa: ispisuje kao rezultat obrade poruku o dužini stringa
 - Ispis stringa: ispisuje kao rezultat obrade sam uneti string
 - Ispis stringa unazad: Ispisuje se uneti string unazad
- Pritiskom na dugme **Obriši** se briše sadržaj polja za unos i rezultat prethodne obrade

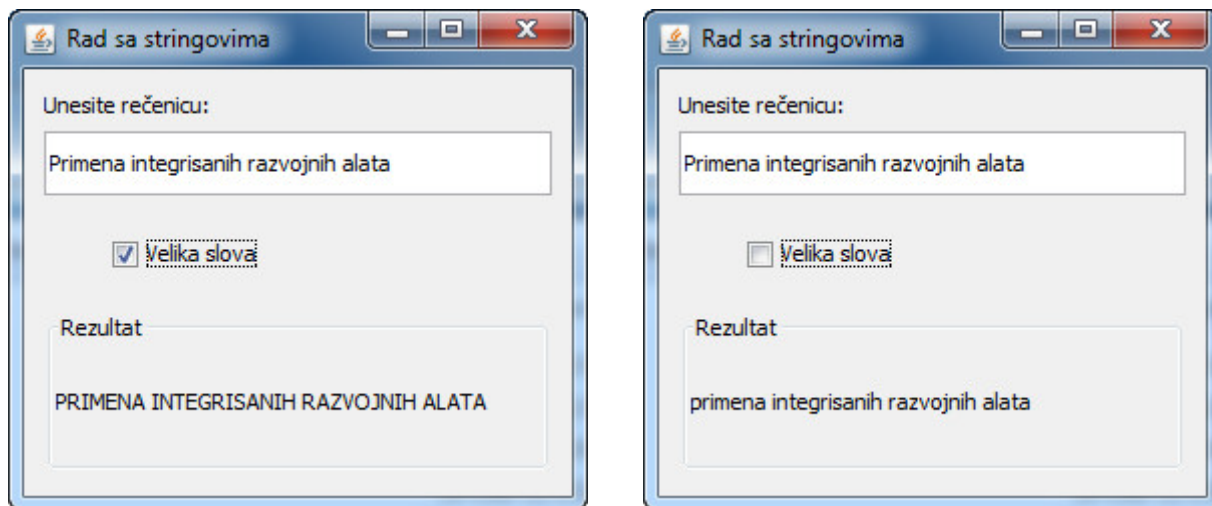
- Pritiskom na dugme **Izlaz** se zatvara aplikacija

Postavljanje radio dugmadi:

1. Na formu postaviti jednu *Button Group* komponentu (biće vidljiva samo u *Inspector-u*)
2. Zatim na formu postaviti potreban broj *Radio Button* komponenti
3. Za svaku od *Radio Button* komponenti odabrati u prozoru properties opciju *ButtonGroup* i u njoj selektovati *Button Group* kojem želimo da pripadaju

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String str = jTextField1.getText();
    int duzina = str.length();
    if (jRadioButton1.isSelected()) {
        jLabel2.setText("Duzina je:" + str.length());
    } else if (jRadioButton2.isSelected()) {
        jLabel2.setText(str);
    } else {
        jLabel2.setText("");
        for (int i = 0; i < duzina; i++) {
            jLabel2.setText(jLabel2.getText()+str.charAt(duzina-i-1));
        }
        // II nacin:
        //String s=jTextField1.getText().trim();
        //char [] niz = new char[len];
        //for (int i = len-1; i >=0; i--) {
            //niz[len-1-i]=s.charAt(i);
            //jLabel2.setText(String.valueOf(niz));
        //}
    }
}
```

Zadatak 3. Programirati aplikaciju koja će imati sledeći korisnički interfejs:



Aplikacija treba da u donjem delu ispisuje unetu rečenicu velikim ili malim slovima u zavisnosti da li je *JCheckBox* selectovan ili ne

```
private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    String str = jTextField1.getText();
    if (jCheckBox1.isSelected()) {
        jLabel1.setText(str.toUpperCase());
    } else {

```

```

        jLabel1.setText(str.toLowerCase());
    }
}

```

Zadatak 1. Programirati aplikaciju koja će sadržati formu kao na slici:

U Combo box-u definisati sledeće opcije:

Dodavanje opcija Combo Box-u (isto u JListi)

1. Postaviti *Combo box* na formu
2. U prozoru *Properties* u polju *model* uneti željene opcije

Dodavanje funkcija komponenti JList

1. Dodavanje elemenata u listu. Dodati naredni kod koji kreira model koji će omogućavati unos elemenata u listu (postavlja se pre implementacije metoda):

```
DefaultListModel listModel = new DefaultListModel();
```

U konstruktoru:

```
jList1.setModel(listModel);
```

2. Za dugme **Dodaj** odabrati opciju *Events>Action>actionPerformed* i u metodu za obradu izuzetka dodati sledeći kod:

```
String str = jTextField5.getText();
listModel.addElement(str);
//jList1.setModel(listModel);
jTextField5.setText("");
```

3. Brisanje elemenata iz liste. Za dugme izbaci dodati metodu za obradu izuzetka sa sledećim kodom:

```
if(jList1.getSelectedIndex()==-1){
    jLabel8.setText("Morate selektovati neku stavku iz
liste!"); //podesi veci font i boju u properties
}
else {
    listModel.removeElementAt(jList1.getSelectedIndex());
    jLabel8.setText("");
}
```

4. Postavljanje sadržaja selektovanog polja liste u *TextField*. Na *jList* kliknuti desnim i odabrati opciju *Events > Mouse > MouseClicked* i u metodu dodati naredni kod

```
jTextField5.setText(listModel.getElementAt(jList1.getSelectedIndex()).toString());
```

ili:

```
int broj = jList1.getSelectedIndex();
String str = model.getElementAt(broj).toString();
jTextField5.setText(str);
```

5. Izmena sadržaja elemenata iz liste se radi sa narednim kodom dodeljenom dugmetu *Izmeni*

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String str = jTextField5.getText();
    int broj = jList1.getSelectedIndex();
    listModel.removeElementAt(broj);
    listModel.add(broj, str);
}
```

6. Klikom na dugme *Ispis* u zavisnosti od selektovane stavke iz Combo Box-a ispisuju se podaci i prvi mail iz liste:

```
String s = jComboBox1.getSelectedItem().toString();
//jComboBox1.setPopupVisible(true);
if(s.equalsIgnoreCase("Ime + prezime")){
    jLabel9.setText("Licni podaci: "+jTextField1.getText()+"
"+jTextField3.getText());
```



```

        jLabel10.setText("Mail: "+listModel.getElementAt(0).toString());
    }else if(s.equalsIgnoreCase("Ime + prezime + nadimak")){
        jLabel9.setText(jTextField1.getText()+"
        "+jTextField3.getText()+" "+jTextField4.getText());
    }else if(s.equalsIgnoreCase("Ime + grad")){
        jLabel9.setText(jTextField1.getText()+" "+jTextField2.getText());
    }else if(s.equalsIgnoreCase("Prezime + grad")){
        jLabel9.setText(jTextField3.getText()+" "+jTextField2.getText());
    }else jLabel9.setText("Selektujte polje u ComboBoxu!");

```

Neke dodatne opcije prozora *Properties* za postavljene forme:

1. Željeni naslov forme se unosi u opciji *Title*
2. U polju *Bounds* se definišu koordinate početne pozicije prozora

ili u Main ->Run metodi:

```

MojaForma mf = new MojaForma();
mf.setSize(200, 200);
mf.setVisible(true);

```

ili ako je u pitanju dijalog:

```

NewJDialog nd = new NewJDialog(new javax.swing.JFrame(), true);
nd.setSize(200, 200);
nd.setVisible(true);

```

3. Opcijom *Resizable* se definiše da li je prozor promenjive ili fiksne veličine
4. U polje *toolTipText* se unosi tekst koji se pojavljuje kada se miš nadnese nad komponentu

Pokretanje programa van NetBeans-a

Odabrali opciju *Run > Clean and Build Main Project*

Pokrenuti *jar* fajl u *dist* direktorijumu projekta

Napomena: Između dijaloga i forme ne postoji velika razlika, prilikom otvaranja nekog dijaloga ne može se pristupiti ni jednom drugom otvorenom dijalogu ili formi i prilikom njegovog zatvaranja zatvaraju se svi pokrenuti dijalozi - forme. Prilikom otvaranja forme može se pristupiti ostalim formama a prilikom zatvaranja te forme zatvara se samo ona.

Podšavanje opcija zatvaranja formi – dijaloga:

Desni klik na formu->Properties-> defaultCloseOperation: DISPOSE (zatvara se samo ta forma), HIDE (sakriva se samo ta forma), EXIT_ON_CLOSE (zatvaraju se sve otvorene forme)