

Grafički interfejsi (Graphical User Interface – GUI)

Zadatak 1

Napraviti klasu PozdravGUI koja predstavlja grafički interfejs kao na slici (Slika 1). Potrebno je da se, kada se klikne na dugme “Pozdrav1”, na labeli (komponenta na kojoj piše JLabel) prikaže tekst “Dobar dan”. Isto tako, kada se klikne na dugme “Pozdrav 2”, na labeli bi trebalo da se prikaže tekst “Laku noc”.

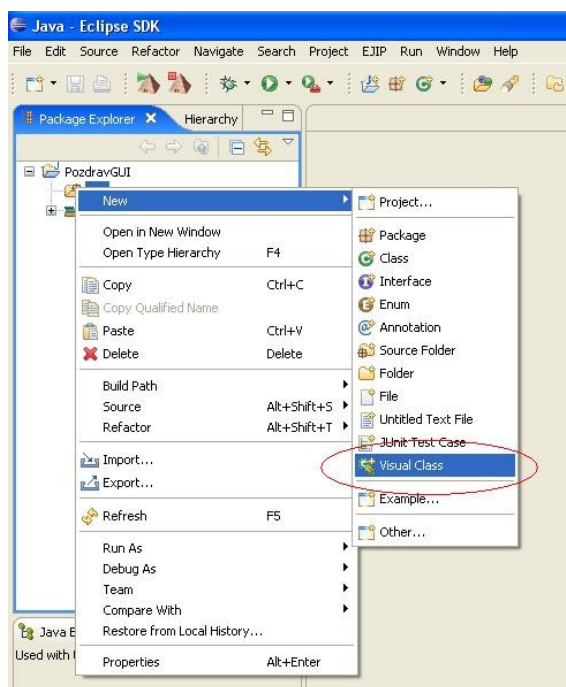


Slika 1: Grafički interfejs "PozdravGUI"

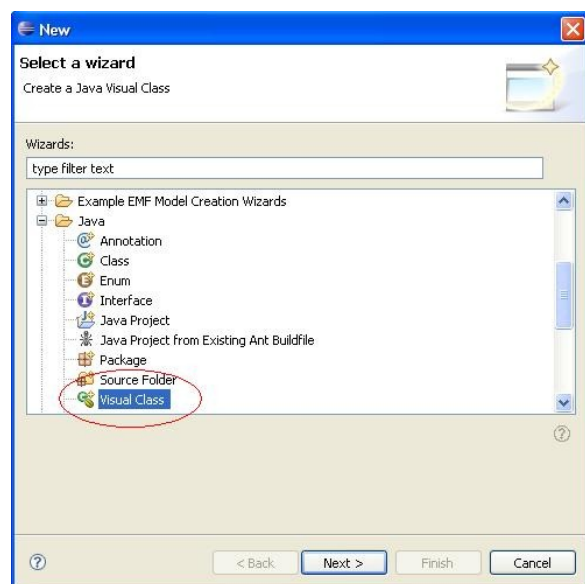
Posle startovanja aplikacije, pokušati sa uvećavanjem i smanjivanjem dimenzija prozora (minimize , maximize) i obratiti pažnju na to šta se dešava sa rasporedom komponenti na prozoru. Nakon toga, podesiti grafički interfejs tako da se ne mogu povećavati niti smanjivati dimenzije prozora.

Rešenje

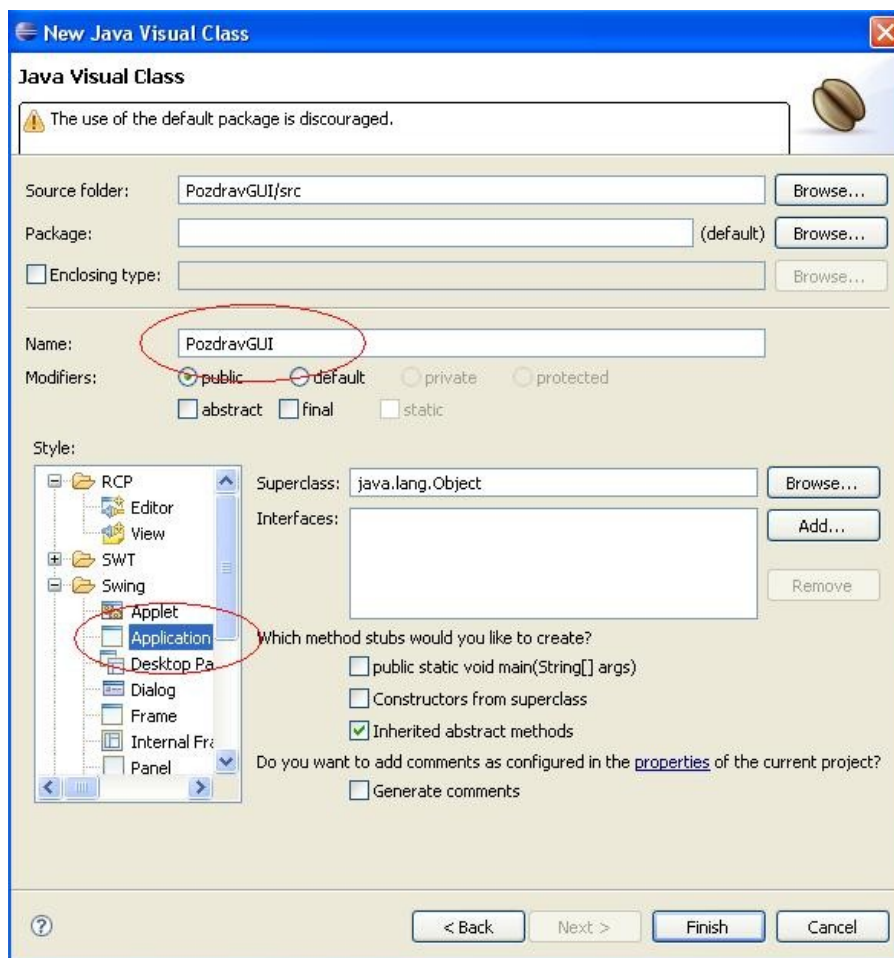
1. Prvo, potrebno je napraviti novi projekat u Eclipse okruženju sa imenom PozdravGUI.
2. Sledeći korak je kreiranje novog grafičkog interfejsa (vizuelne klase). Kliknuti desnim tasterom miša na naziv projekta ili na “src” folder i iz menija izabrati New->VisualClass (Slika 2). Ako te stavke nema na meniju onda izabrati New->Other pa iz prozora koji se pojavi (Slika 3) izabrati Java->VisualClass i ići na Next. U svakom slučaju, na ekranu se pojavljuje sledeći prozor (Slika 4). U polje Name upisati ime grafičkog interfejsa - “PozdravGUI”, a iz menija Style izabrati Swing->Application i kliknuti na “Finish”. Novi grafički interfejs je kreiran (Slika 5). Opisani postupak za kreiranje grafičkog interfejsa je uvek isti.



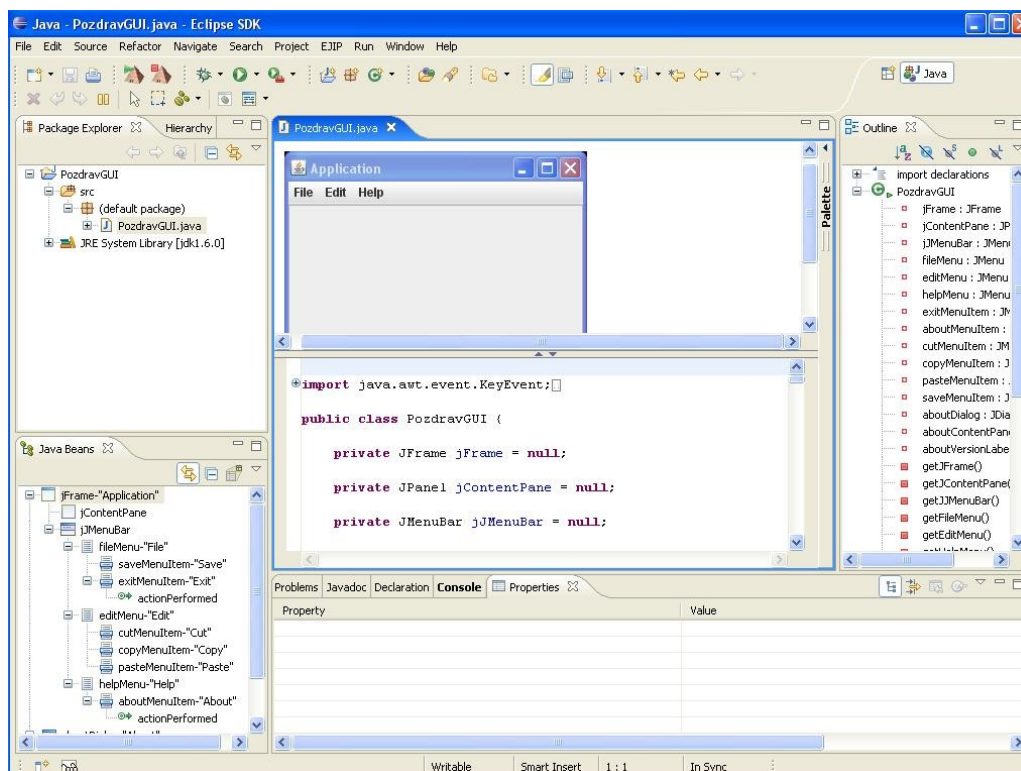
Slika 2: Kreiranje novog grafičkog interfejsa - 1. korak



Slika 3: Kreiranje novog grafičkog interfejsa - 1. korak (drugi način)



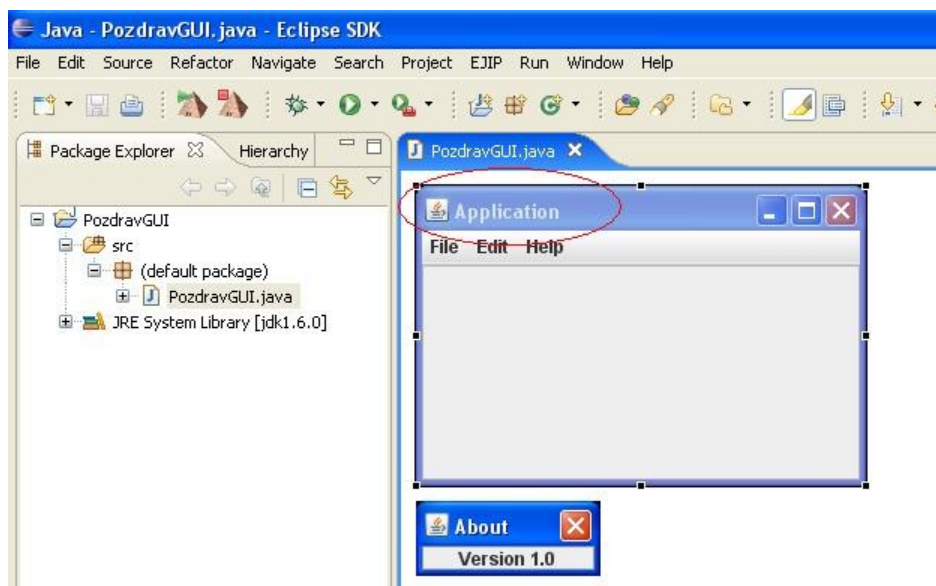
Slika 4: Kreiranje novog grafičkog interfejsa - 2. korak



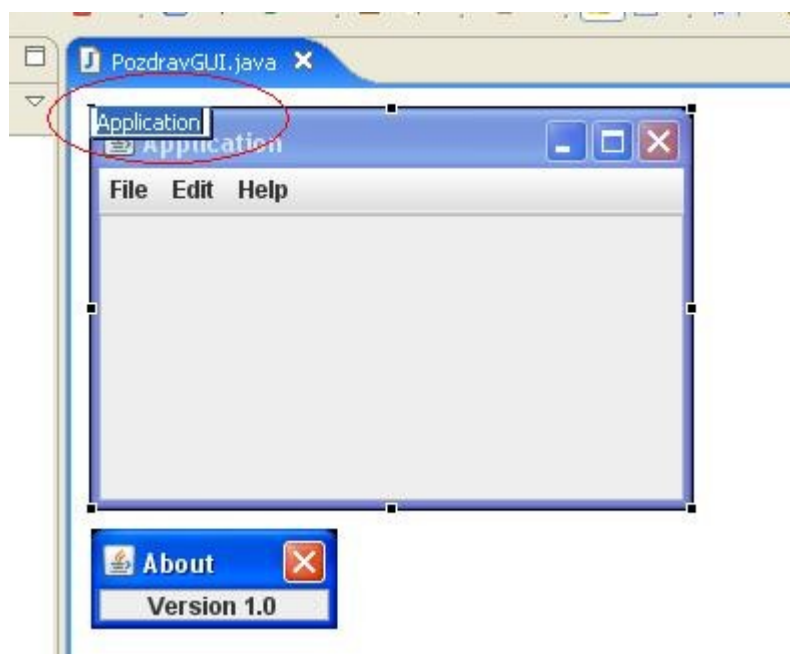
Slika 5: Kreiranje novog grafičkog interfejsa - 3. korak

- Na prethodnoj slici se vidi kako izgleda tek kreirani grafički interfejs. Taj interfejs još uvek nema nikakve

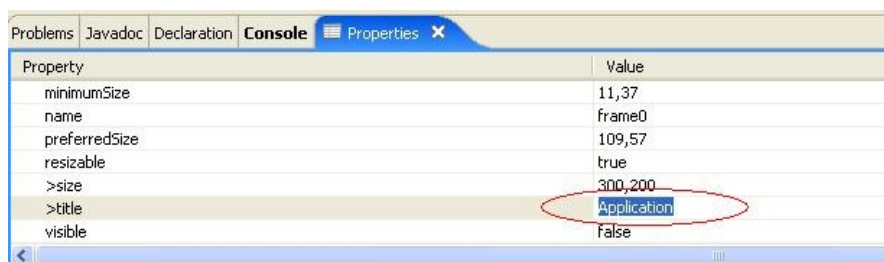
komponente na sebi (u ovom slučaju, dugmiće i labele), a čak ni naslov na formi nije odgovarajući (stoji “Application” a trebalo bi da stoji “Pozdrav” - Slika 1). Naslov na formi se može promeniti na dva načina. Prvi način je tako što se klikne na sam naslov i time se otvori mali editor u koji se upiše novi naslov (Slika 6, Slika 7). Drugi način je tako što se klikne na formu i iz Properties taba promeni vrednost atributa “Title” na “Pozdrav” (Slika 8).



Slika 6: Menjanje naslova - 1. korak



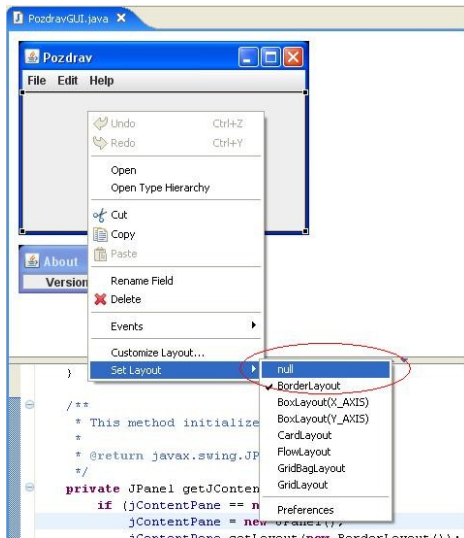
Slika 7: Menjanje naslova - 2. korak



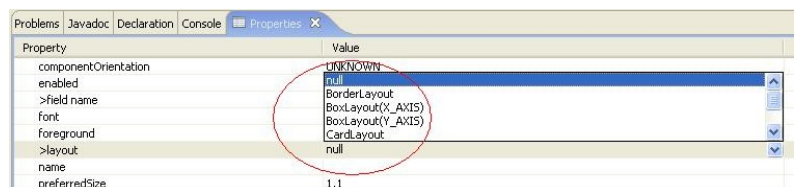
Slika 8: Menjanje naslova - drugi način

- Pošto je naslov sada podešen, potrebno je dodati neke grafičke komponente na formu. Ako se pogleda izgled

traženog interfejsa (Slika 1), može se videti da je potrebno dodati tri grafičke komponente: dva dugmeta i jednu labelu. Na prvom dugmetu bi trebalo da piše “Pozdrav 1” a na drugom “Pozdrav 2”, dok bi na labeli trebalo da piše “JLabel”. Pre nego što se dodaju komponente na formu, potrebno je izabrati neki od predefinisanih rasporeda komponenti na formi - takozvani Layout-i. U ovom primeru se koristi null layout. Podešavanje layout-a se može uraditi na dva načina. Jedan je da se desnim tasterom miša klikne na praznu površinu forme i iz menija izabere Set Layout->>null (Slika 9), a drugi je da se levim tasterom miša klikne na istu tu površinu pa da se iz Properties taba podesi atribut layout na null (Slika 10).

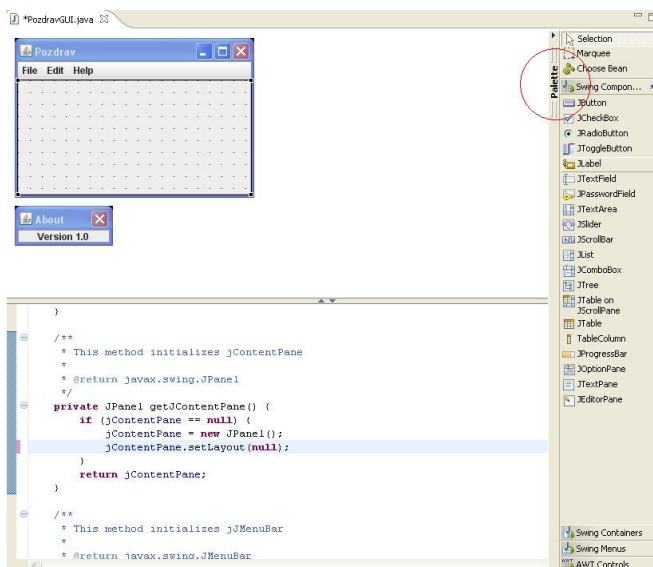


Slika 9: Podešavanje layout-a

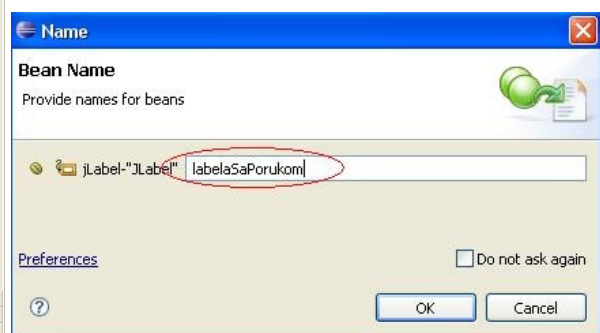


Slika 10: Podešavanje layout-a (drugi način)

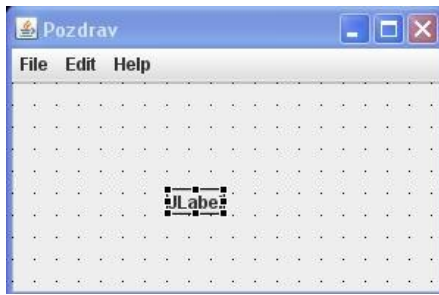
- Sada se može pristupiti dodavanju komponenti. Neka se prvo dodaje labela. Namestiti miš tačno iznad dela ekrana na kome piše Pallette (gore desno). Automatski se otvara paleta svih grafičkih komponenti (Slika 11). Potrebno je kliknuti na JLabel pa onda kliknuti na formu. U tom trenutku se otvara novi prozor u kome se traži unos imena komponente. Uneti ime “labelaSaPorukom” (Slika 12). Ovo je ime objekta labele i koristi se kasnije da bi se pozivale metode za menjanje teksta labele. Kada se klikne na dugme OK, labela je kreirana (Slika 13). Njena veličina i pozicija može da se ručno menja korišćenjem miša - pomeranje i razvlačenje (Slika 14). Tekst na samoj labeli može da se promeni slično kao što se manja naslov forme - klikom na samu labelu ili promenom atributa text iz properties taba, ali se to u zadatku nije tražilo, pa na labeli ostaje izvorni tekst “JLabel”.



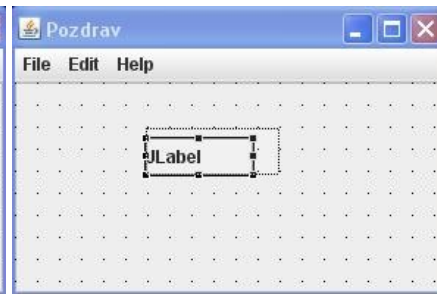
Slika 11: Postavljanje komponenti na formu - 1. korak



Slika 12: Postavljanje komponenti na formu - 2. korak



Slika 13: Labela je kreirana

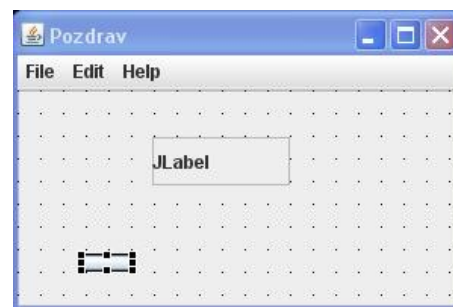


Slika 14: Promena veličine i poz. labela

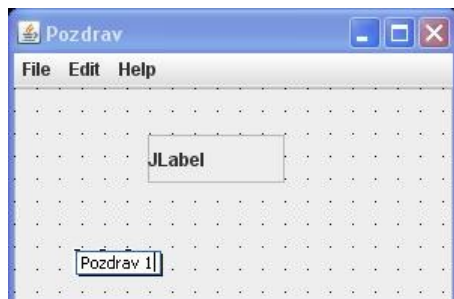
6. Sledeće, potrebno je dodati jedno dugme (button). Izabrati iz palete komponenti JButton i kliknuti na formu. Kao ime komponente upisati "Pozdrav1Dugme" (Slika 15). Kada se dugme kreira (Slika 16), potrebno je na njega postaviti pravi tekst tj. naslov - "Pozdrav 1". Postavljanje teksta se vrši slično kao i naslov za formu - klikom miša direktno na dugme pri čemu se otvara polje za upis (Slika 17) ili menjanjem atributa text (Slika 18). Posle promene naslova dugme se i dalje ne vidi lepo pa je potrebno podesiti veličinu i poziciju dugmeta na formi korišćenjem miša (Slika 19).



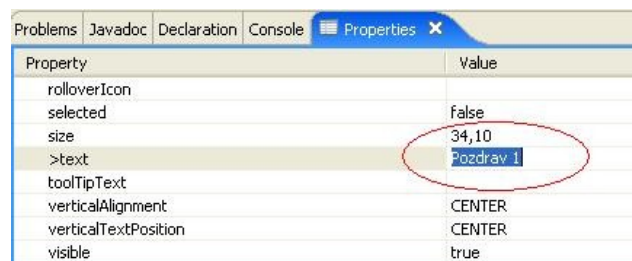
Slika 15: Kreiranje dugmeta 1. korak



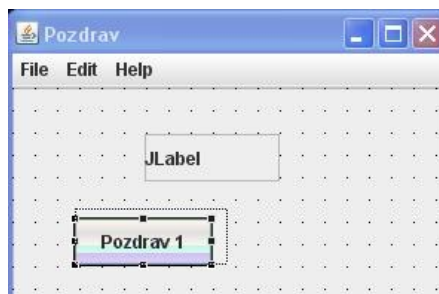
Slika 16: Kreiranje dugmeta 2. korak



Slika 17: Kreiranje dugmeta 3. korak



Slika 18: Kreiranje dugmeta 3. korak (drugi način)



Slika 19: Kreiranje dugmeta 4. korak

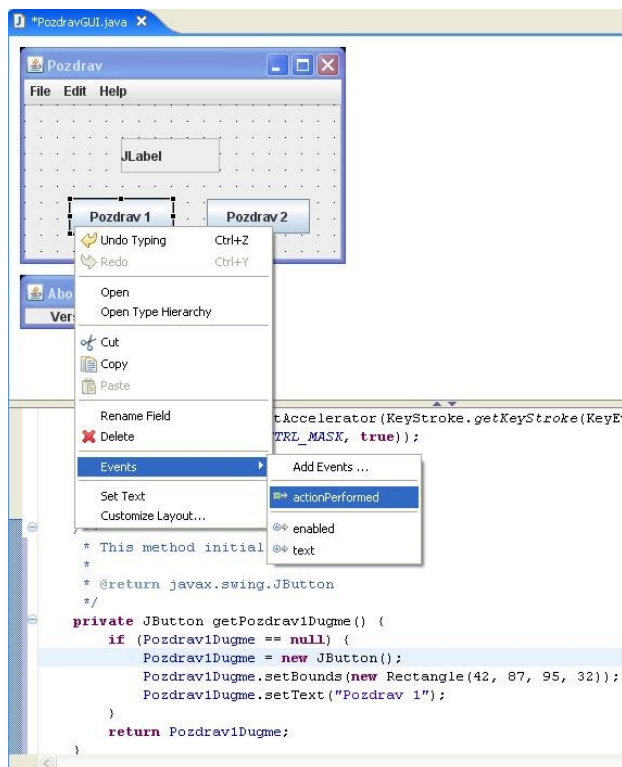


Slika 20: Gotov GUI

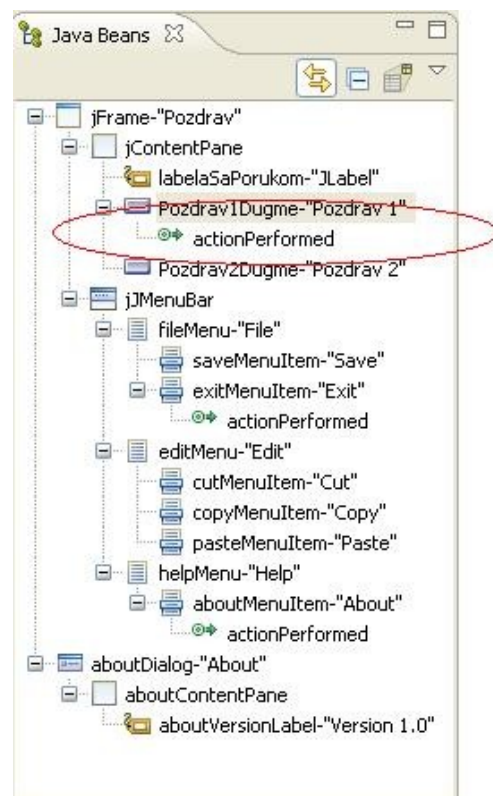
7. Korišćenjem istog postupka kao za prvo dugme, kreirati i drugo dugme sa imenom "Pozdrav2Dugme" i naslovom "Pozdrav 2". Forma bi trebalo da izgleda kao na slici (Slika 20). Ako se malo bolje pogleda, to je upravo ono što se razililo u zadatku.
8. Ovako formiran grafički interfejs izgleda upravo onako kako se traži u zadatku. Međutim, on nema nikakvu funkcionalnost, tj. ne radi ništa. Ako bi se program startovao i ako bi se kliknulo na bilo koje dugme, ništa se ne bi desilo. Drugim rečima, potrebno je da se programu doda funkcionalnost. Grafički interfejsi reaguju na događaje (event) - kada se klikne mišem na nešto, kada se neka vrednost promeni itd. Svaki događaj može biti

u vezi sa nekom komponentom. U zadatku se traži da na klik mišem na dugme “Pozdrav 1”, labela dobije tekst “Dobar dan”. To znači da se događaj klikanja mišem treba povezati sa dugmetom Pozdrav1Dugme. Metode koje se automatski pozivaju svaki put kada se neki događaj desi se zovu event handler-i. U ovom slučaju, potrebno je dodati event handler za dugme Pozdrav1Dugme. To se radi tako što se desnim tasterom miša klikne na Pozdrav1Dugme i iz menija izabere Events->actionPerformed (Slika 21). Kada se to uradi, dešavaju se dve stvari: u JavaBeans tab-u se pojavljuje novi event handler za dugme Pozdrav1Dugme (Slika 22) a u Java editoru se automatski kreira nova metoda za event handler - metoda actionPerformed (Slika 23). Metoda actionPerformed će automatski da se pozove i izvrši svaki put kada neko klikne na dugme Pozdrav1Dugme. U ovom trenutku, ova metoda ne radi ništa tj. samo na ekranu ispisuje poruku “actionPerformed()”. U zadatku se traži da se tekst “Dobar dan” ispiše u okviru labele. Da bi to moglo da se desi, potrebno je “actionPerformed” metodu popraviti tako da sadrži sledeći kod.

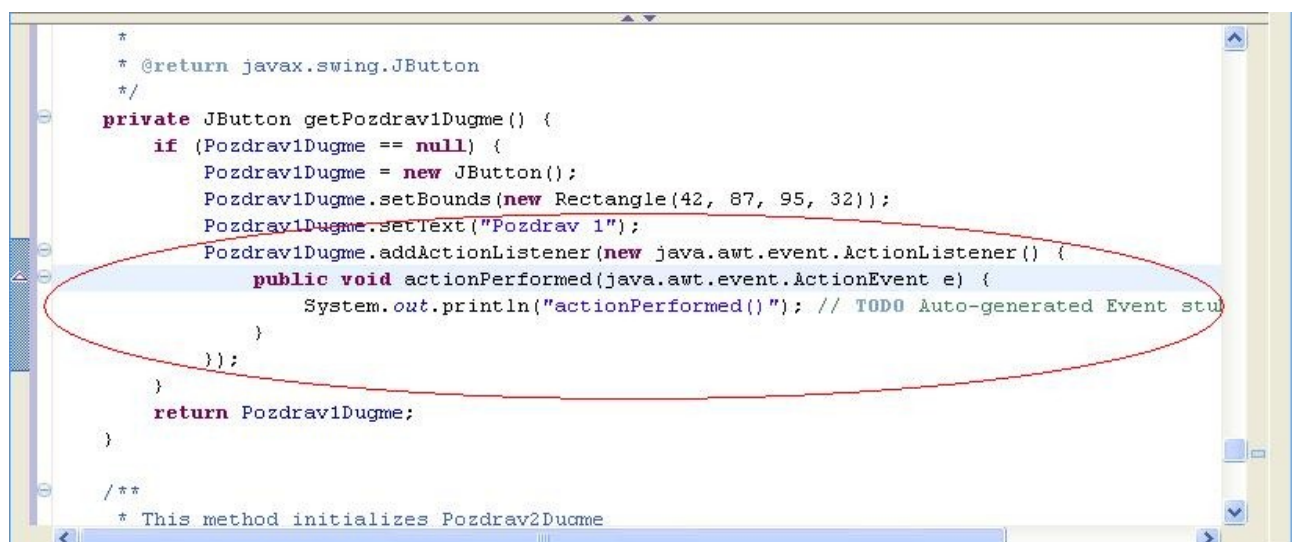
```
public void actionPerformed(java.awt.event.ActionEvent e) {
    labelaSaPorukom.setText("Dobar dan");
}
```



Slika 21: Dodavanje događaja 1. korak



Slika 22: Dodavanje događaja 2. korak

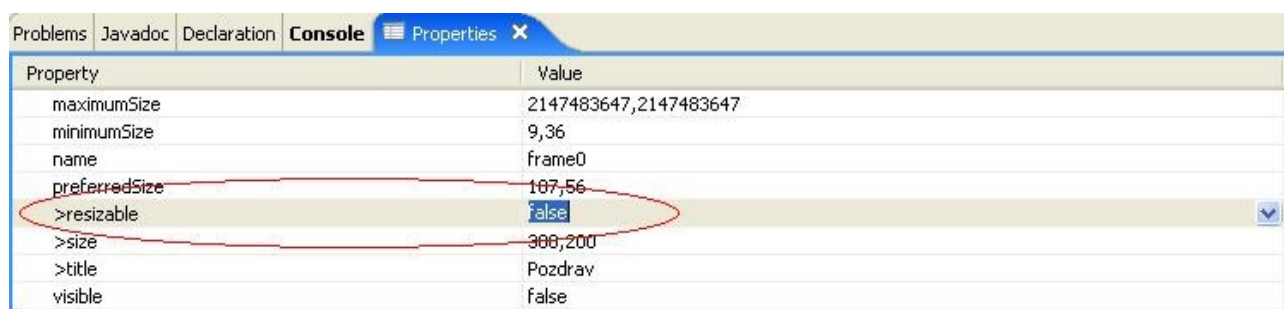


Slika 23: Dodavanje događaja 2. korak drugi deo

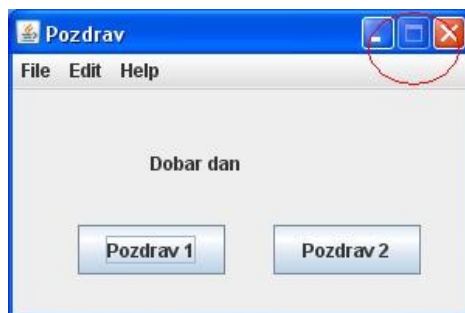
9. Potrebno je na isti način dodati event handler za Pozdrav2Dugme. Kod njegove actionPerformed metode treba da se prepravi tako da bude:

```
public void actionPerformed(java.awt.event.ActionEvent e) {  
    labelaSaPorukom.setText("Laku noc");  
}
```

10. Potrebno je startovati program. Iz package explorera kliknuti desnim tasterom miša na “PozdravGUI.java” i iz menija izabrati Run As->Java Application. Na monitoru će ubrzo da se pojavi grafički interfejs PozdravGUI.
11. Ako se, dok aplikacija radi, pokuša sa povećavanjem ili smanjivanjem forme, može se primetiti da sve komponente (dugmići i labela) ostaju na istom mestu a da se površina oko njih povećava ili smanjuje. Ovo je karakteristika null layout-a - komponente se mogu postavljati po izboru (lokacija i veličina), ali se prilikom maksimizacije ili minimizacije cele forme komponente ne pomeraju već ostaju iste veličine i oblika. Aplikacija se zatvara kao i bilo koja druga Windows aplikacija - klikom na krstić u gornjem desnom uglu forme.
12. Da bi se sprečila promena dimenzija forme, potrebno je promeniti formin atribut “resizable” na false u okviru Properties tab-a (Slika 24). Naravno, to se može uraditi tek pošto se zatvori aplikacija i pošto se na tab-u JavaBeans izabere forma (JFrame). Kada se ponovo startuje aplikacija, veličina forme neće moći da se menja (Slika 25). Ovim je aplikacija završena.



Slika 24: Podešavanje forme tako da ne mogu da joj se menjaju dimenzije u toku rada aplikacije



Slika 25: Gotova aplikacija

Zadatak 2

Napraviti grafički interfejs KopiranjeTekstaGUI koji izgleda kao na slici (Slika 26). Pri pritisku na dugme Prekopiraj, potrebno je da se tekst koji korisnik upiše u levo polje za unos ispiše u desnom polju za unos i u desnoj labeli na kojoj piše Tekst. Pri pritisku na dugme Izlaz, potrebno je da se aplikacija prekine tj. da prestane sa izvršavanjem. Podesiti grafički interfejs tako da se, u toku rada aplikacije, ne mogu menjati dimenzije forme.



Slika 26: KopiranjeTekstaGUI

Rešenje

1. Prvo, potrebno je napraviti novi projekat u Eclipse okruženju.
2. Sledeći korak je kreiranje novog grafičkog interfejsa (vizuelne klase) sa imenom KopiranjeTekstaGUI.
3. Promeniti naslov na formi tako da bude "Kopiranje teksta".
4. Potrebno je podesiti formu tako da ima null layout.
5. Dodati dugme na formu sa imenom PrekopirajDugme i naslovom "Prekopiraj".
6. Dodati dugme na formu sa imenom IzlazDugme i naslovom "Izlaz".
7. Dodati labelu na formu sa imenom LabelaUnos i naslovom "Unesite tekst". Tekst na labeli se menja na sličan način kao i tekst na dugmetu. Forma bi u ovom trenutku trebalo da izgleda kao na slici (Slika 27). Bitno je primetiti da će, za razliku od prethodnog primera, upravo dodata labela uvek sadržati isti tekst.



Slika 27: Delimično gotov GUI

8. Dodati na formu polje za unos sa imenom PoljeZaUnos. Da bi se to uradilo, potrebno je iz palete izabrati JTextField (Slika 28) i dodati ga na formu. Kada se doda, potrebno je da se podesi njegova veličina i pozicija (Slika 29). Polje za unos (JTextField komponenta) služi za to da korisnik unese neki tekst ili broj preko grafičkog interfejsa.

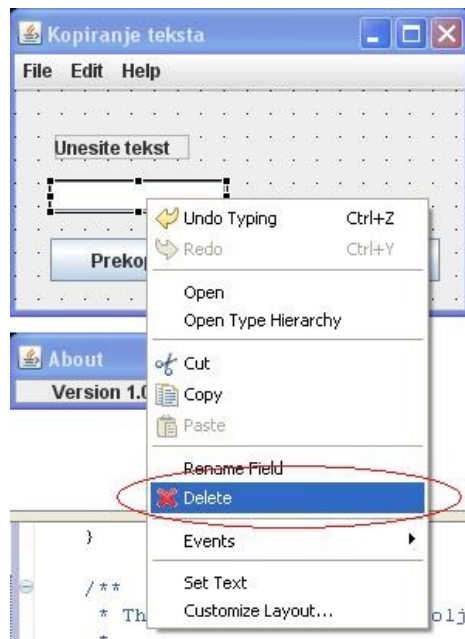


Slika 28: Dodavanje polja za unos 1. korak



Slika 29: Dodavanje polja za unos 2. korak

9. Dodati na formu drugo polje za unos sa imenom PoljeZaIspis. Ako u bilo kom trenutku pogrešite i dodate neku komponentu koji niste hteli, uvek možete da je obrišete tako što ćete kliknuti desnim tasterom miša na nju i iz menija izabrati Delete (Slika 30).
10. Dodati na formu labelu sa imenom LabelaZaIspis i tekstom "Tekst".
11. Podesiti atribut forme "resizeable" na false uz pomoć Properties tab-a. Grafički interfejs je gotov (Slika 31).



Slika 30: Brisanje komponenti



Slika 31: Gotov GUI

12. Dodate event handler za dugme PrekopirajDugme koji sadrži sledeći kod:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    String tekst = PoljeZaUnos.getText();

    PoljeZaIspis.setText(tekst);
    LabelaZaIspis.setText(tekst);
}
```

13. Dodate event handler za dugme IzlazDugme. Potrebno je da se, kada korisnik klikne na ovo dugme, prekine rad programa tj. da bude isti efekat kao da je kliknuo na krstić u gornjem desnom uglu forme. Prema tome, metoda actionPerformed (event handler) bi trebalo da sadrži sledeći kod:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    //Sledeca komanda služi za prekidanje rada programa
    System.exit(0);
}
```

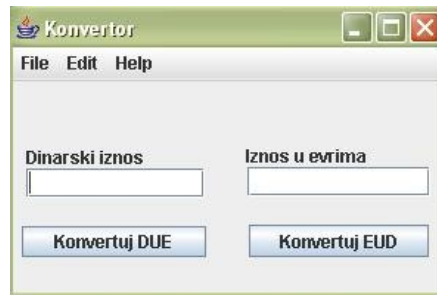
14. Startovati program i testirati ga.

Zadatak 3

Napisati klasu Konvertor koja ima:

- Javnu metodu koja kao ulazni argument dobija iznos u dinarima (realan broj). Ova metoda konvertuje ovaj iznos u evre i vraća ga (1EUR = 80DIN).
- Javnu metodu koja kao ulazni argument dobija iznos u evrima (realan broj). Ova metoda konvertuje ovaj iznos u dinare i vraća ga (1EUR = 80DIN).

Napraviti klasu KonvertorGUI koja izgleda kao na slici (Slika 32) i povezati je sa konvertorom tako da se na pritisak dugmeta "Konvertuj DUE" dinarski iznos koji je upisan u levo polje za unos konvertuje u evre i prikaže u desnom polju za unos. Takođe, potrebno je da se na pritisak dugmeta Konvertuj EUD iznos u evrima koji je upisan u desno polje za unos konvertuje u dinare i prikaže u levom polju za unos. Podesiti grafički interfejs tako da se, u toku rada aplikacije, ne mogu menjati dimenzije forme.



Slika 32: KonvertorGUI

Rešenje

```
public class Konvertor {

    public double konvertujDUE(double dinari) {
        return dinari/80;
    }

    public double konvertujEUD(double evri) {
        return evri*80;
    }

}
```

1. Kreirati nov grafički interfejs (vizuelnu klasu) sa imenom KonvertorGUI.
2. Promeniti naslov na formi tako da bude "Konvertor".
3. Potrebno je podesiti formu tako da ima null layout.
4. Dodati dugme na formu sa imenom KonvertujDUEDugme i naslovom "KonvertujDUE".
5. Dodati dugme na formu sa imenom KonvertujEUDDugme i naslovom "KonvertujEUD".
6. Dodati labelu na formu sa bilo kojim imenom i naslovom "Dinarski iznos".
7. Dodati labelu na formu sa bilo kojim imenom i naslovom "Iznos u evrima".
8. Ispod labele na kojoj piše "Dinarski iznos" dodati novo polje za unos sa imenom PoljeZaUnosDinara.
9. Ispod labele na kojoj piše "Iznos u evrima" dodati novo polje za unos sa imenom PoljeZaUnosEvra.
10. Podesiti atribut forme "resizeable" na false uz pomoć Properties tab-a. Grafički interfejs je gotov.
11. Dodati event handler za dugme KonvertujDUEDugme koji sadrži sledeći kod:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    Konvertor k = new Konvertor();

    double dinari = Double.parseDouble(PoljeZaUnosDinara.getText());
    double evri = k.konvertujDUE(dinari);
    PoljeZaUnosEvra.setText(""+evri);
}
```

12. Dodati event handler za dugme KonvertujEUDDugme koji sadrži sledeći kod:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    Konvertor k = new Konvertor();

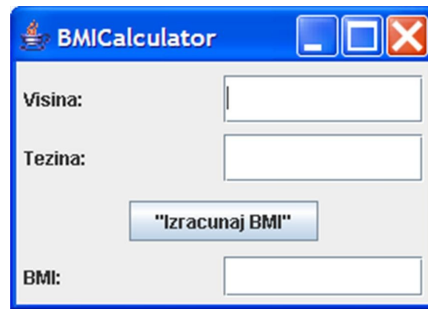
    double evri = Double.parseDouble(PoljeZaUnosEvra.getText());
    double dinari = k.konvertujEUD(evri);
    PoljeZaUnosDinara.setText(""+dinari);
}
```

Zadatak 4

Napisati javnu klasu BMIKalkulator koja ima:

- Javnu metodu izracunajBMI koja kao ulazni argument dobija visinu i težinu čoveka (realni brojevi). Visina je data u metrima a težina u kilogramima. Metoda izračunava i vraća BMI tj. Body Mass Index po formuli $BMI = \text{težina (kg)} / [\text{visina(m)}]^2$

Napisati klasu BMIKalkulatorGUI kojom se kreira korisnički interfejs prikazan na sledećoj slici (Slika 33). Potrebno je obezbediti da nakon što korisnik unese podatke o visini i težini u odgovarajuća polja i klikne na dugme 'Izracunaj BMI' u polju BMI se ispisuje izračunata vrednost BMI (Body Mass Index)-a. Podesiti grafički interfejs tako da se, u toku rada aplikacije, ne mogu menjati dimenzije forme. Takođe, potrebno je podesiti da se tekst polja za unos u kome se ispisuje vrednost BMI ne može ručno menjati.



Slika 33: BMIKalkulatorGUI

Rešenje

```
public class BMIKalkulator {

    public double izracunajBMI(double visina, double tezina){
        return tezina/(visina*visina);
    }

}
```

1. Kreirati nov grafički interfejs (vizuelnu klasu) sa imenom BMIKalkulatorGUI.
2. Promeniti naslov na formi tako da bude "BMICalculator".
3. Potrebno je podesiti formu tako da ima null layout.
4. Dodati dugme na formu sa imenom IzracunajBMIDugme i naslovom "Izracunaj BMI".
5. Dodati labelu na formu sa bilo kojim imenom i naslovom "Visina:".
6. Dodati labelu na formu sa bilo kojim imenom i naslovom "Tezina:".
7. Dodati labelu na formu sa bilo kojim imenom i naslovom "BMI:".
8. Pored labele na kojoj piše "Visina:" dodati novo polje za unos sa imenom PoljeZaUnosVisine.
9. Pored labele na kojoj piše "Tezina:" dodati novo polje za unos sa imenom PoljeZaUnosTezine.
10. Pored labele na kojoj piše "BMI:" dodati novo polje za unos sa imenom PoljeZaIspisBMI. Da bi se sprečilo ručno unošenje vrednosti u ovo polje, potrebno je da se vrednost njegovog atributa "editable" podesi na false uz pomoć Properties tab-a.
11. Podesiti atribut forme "resizable" na false uz pomoć Properties tab-a. Grafički interfejs je gotov.
12. Dodati event handler za dugme IzracunajBMIDugme koji sadrži sledeći kod:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    BMIKalkulator b = new BMIKalkulator();

    double visina = Double.parseDouble(PoljeZaUnosVisine.getText());
    double tezina = Double.parseDouble(PoljeZaUnosTezine.getText());
    double bmi = b.izracunajBMI(visina, tezina);
    PoljeZaIspisBMI.setText(""+bmi);
}
```

14. Startovati aplikaciju. Primititi da se ne može ručno upisati vrednost u polje PoljeZaIspisBMI.

Zadatak 5

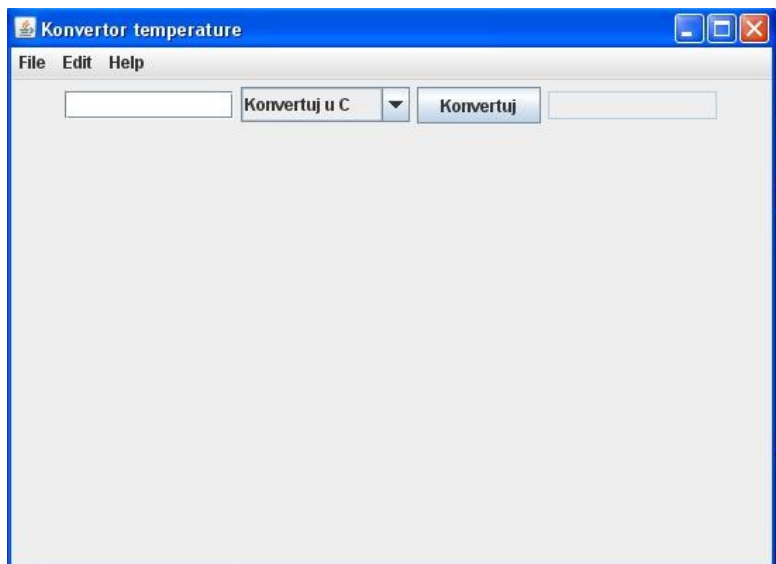
Napraviti javnu klasu **KonvertorTemperature**. Ova klasa bi trebalo da ima:

- Javnu metodu **konvertujCUF** koja prima kao ulazni parametar temperaturu izraženu u Celzijusima (realan broj npr. 27.9), pretvara je u temperaturu u Farenhajtima i vraća kao rezultat. Konverzija se vrši po formuli $T(F) = (T(C)*9)/5 + 32$.
- Javnu metodu **konvertujFUC** koja prima kao ulazni parametar temperaturu izraženu u Farenhajtima (realan broj npr. 100.4), pretvara je u temperaturu u Celzijusima i vraća kao rezultat. Konverzija se vrši po formuli $T(C) = (T(F)-32)*(5/9)$.

Napisati klasu KonvertorTemperatureGUI kojom se kreira korisnički interfejs prikazan na sledećoj slici (Slika 34). Potrebno je obezbediti da nakon što korisnik unese podatke o temperaturi u odgovarajuće polje i iz padajuće liste (combo-box) izabere koju vrstu konverzije želi ("konvertuj u C" ili "konvertuj u F"), na klik dugmeta "Konvertuj" u polje za ispis sa desne strane padajuće liste se prikazuje konvertovana temperatura. Konverzija temperature se vrši pozivanjem odgovarajuće metode klase KonvertorTemperature. Podesiti grafički interfejs tako da se, ako se promene dimenzije forme, komponente na formi raspoređuju kao na slici (Slika 35). Takođe, potrebno je podesiti da se tekst polja za ispis konvertovane temperature ne može ručno menjati.



Slika 34: KonvertorTemperatureGUI



Slika 35: KonvertorTemperatureGUI - raspored komponenti posle promena dimenzije forme

Rešenje

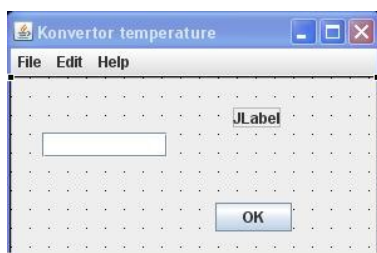
```
public class KonvertorTemperature {

    public double konvertujCUF(double tc) {
        // Ovde mora da stoji 5.0 i 9.0 a ne 5 i 9 da se ne bi pozvalo
        // celobrojno DIV deljenje vec deljenje realnih brojeva
        return ((tc * 9.0) / 5.0) + 32;
    }

    public double konvertujFUC(double tf) {
        // Ovde mora da stoji 5.0/9.0 a ne 5/9 da se ne bi pozvalo
        // celobrojno DIV deljenje vec deljenje realnih brojeva
        return (tf - 32) * (5.0 / 9.0);
    }

}
```

1. Kreirati nov grafički interfejs (vizuelnu klasu) sa imenom KonvertorTemperatureGUI.
2. Promeniti naslov na formi tako da bude "Konvertor temperature".
3. Potrebno je podesiti formu tako da ima **FlowLayout**. Za razliku od null layout-a koji raspoređuje komponente po formi u raspored koji korisnik želi (komponente ostaju na onom mestu gde se postave - Slika 36), FlowLayout ne dozvoljava to i raspoređuje sve komponente u jedan predefinisani raspored. Tačnije rečeno, kako se komponente dodaju na formu, FlowLayout ih raspoređuje u jedan red sve dok mogu da stanu u taj red (Slika 37). Kada neka komponenta ne može da stane u zadati red, FlowLayout je raspoređuje u sledeći red (Slika 38) itd. Međutim, glavna razlika između ova dva layout-a je upravo u tome šta se dešava sa rasporedom komponenti na formi ako se forma poveća ili smanji. Kada je null layout u pitanju, komponente ostaju uvek isto raspoređene bilo da se forma poveća (Slika 39) ili smanji (Slika 40) - samo se prostor oko njih povećava ili smanjuje. Kod flow layout-a je drugačije. Ako se promene dimenzije prozora, komponente se preraspoređuju ravnomerno i to po principu - staviti što više komponenti u jedan red, a one koje ne mogu da stanu staviti u sledeći itd. (Slika 41 i Slika 42).



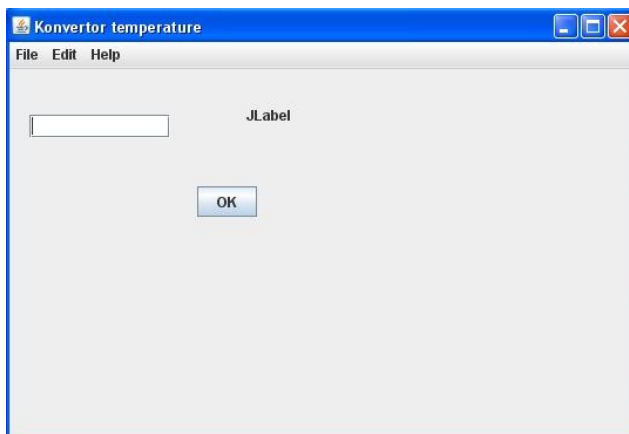
Slika 36: Primer komponenti raspoređenih prema null layout-u



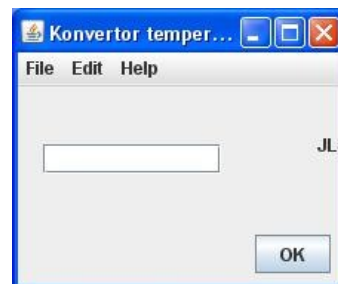
Slika 37: Iste komponente samo raspoređene prema flow layout-u



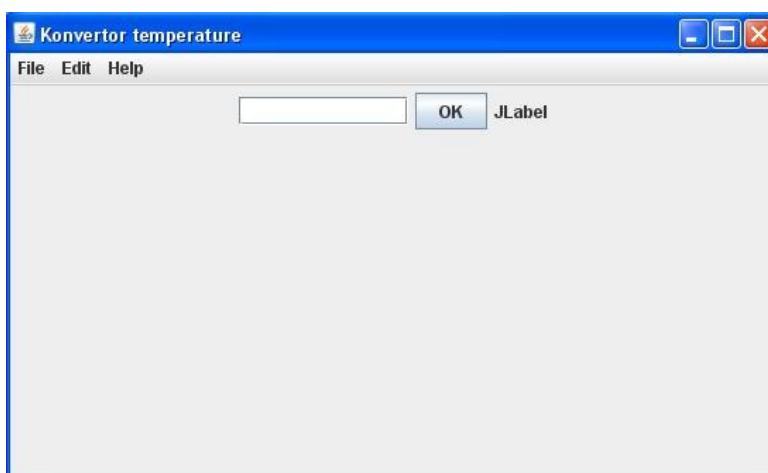
Slika 38: Komponenta koja ne može da stane u tekući red se automatski raspoređuje u sledeći



Slika 39: Null layout - komponente ostaju isto raspoređene čak iako se forma poveća



Slika 40: Null layout - komponente ostaju isto raspoređene čak iako se forma smanji



Slika 41: Flow layout - komponente se raspoređuju ravnomerno ako se forma poveća

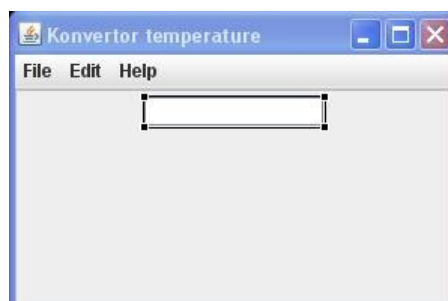


Slika 42: Flow layout - komponente se raspoređuju ravnomerno ako se forma smanji

4. Dodati novo polje za unos sa imenom PoljeZaUnosTemperature. Primetiti da se polje za unos pojavilo na sredini “prvog reda” forme i da je veoma malih dimenzija (Slika 43). Dimenzije ove komponente se, nažalost, ne mogu promeniti “razvlačenjem” uz pomoć miša. Jedini način da se ovo polje za unos dovede do prave veličine je da se izmeni njegov atribut “preferred size” u okviru Properties tab-a. Ovaj atribut trenutno ima vrednost “4,20”. To znači da je polje za unos “dugačko” 4 piksela (X osa) i “visoko” 20 piksela (Y osa). Umesto ove vrednosti, potrebno je uneti “120,20” da bi polje imalo odgovarajuću dužinu. Posle promene, forma bi trebalo da izgleda kao na slici (Slika 44).



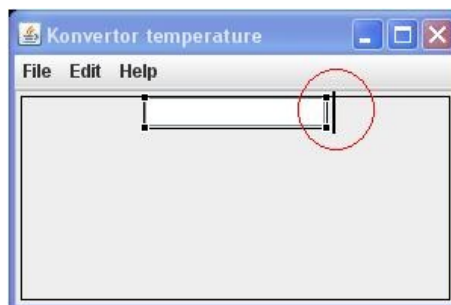
Slika 43: Konvertor temperature



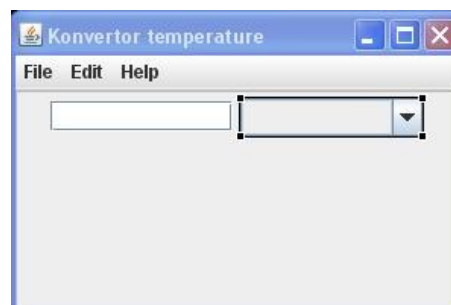
Slika 44: Konvertor temperature posle promene dimenzija komponente

5. Dodati padajuću listu (JComboBox komponenta) na formu sa imenom KonvertorPadajucaLista. Primetiti da se svaka komponenta može dodati u ispred ili iza postojećih komponenti - u odnosu na red u kome se nalaze (Slika 45). Potrebno je da se lista doda tako da bude iza polja za unos. I dimenzije ove komponente se menjaju promenom atributa “preferred size” u okviru Properties tab-a. Promeniti vrednost ovog atributa na 120,25.

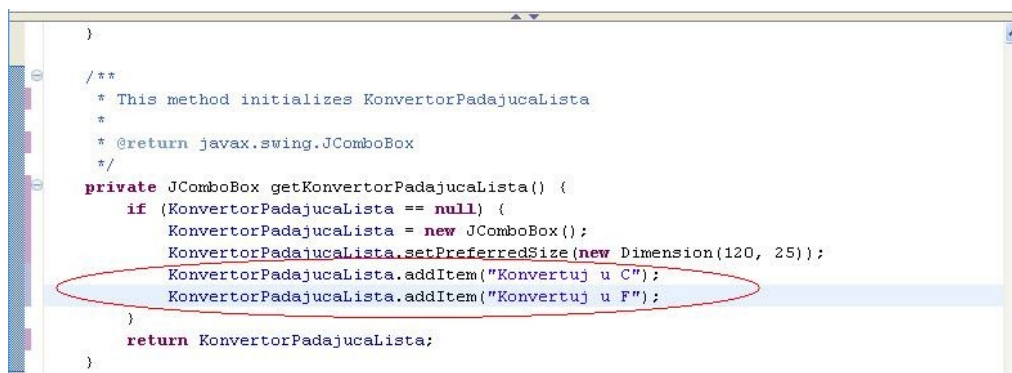
Forma bi trebalo da izgleda kao na slici (Slika 46). Padajuća lista koja je upravo dodata ne sadrži nikakve stavke, a trebalo bi da sadrži dve: “Konvertuj u C” i “Konvertuj u F”. Dodavanje stavki se vrši upisivanjem odgovarajućeg koda - Visual Editor za Eclipse nema meni koji bi se pobrinuo za ovo. Dakle, da bi se dodale stavke u padajuću listu, potrebno je ukucati sledeći kod u okviru editora (Slika 47). Iako se stavke liste ne vide u ovom trenutku, videće se kada se bude startovala aplikacija.



Slika 45: Konvertor temperature - dodavanje komponenti

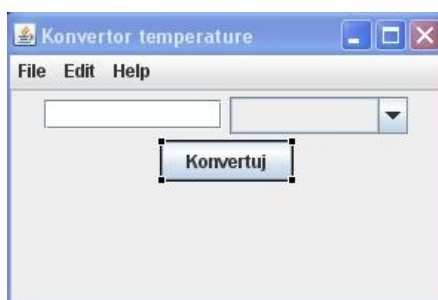


Slika 46: Konvertor temperature - dodata padajuća lista

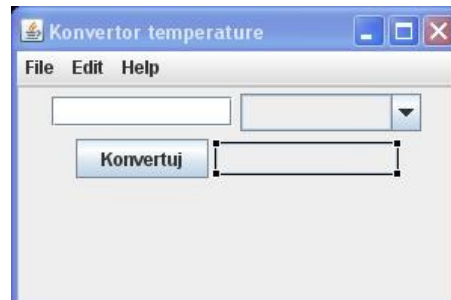


Slika 47: Dodavanje stavki u padajuću listu

6. Dodati dugme sa imenom KonvertujDugme i naslovom “Konvertuj”. Veličina dugmeta će automatski da se prilagodi naslovu, a ako je i pored toga potrebno da dugme bude veće ili manje, to se može namestiti izmenom vrednosti atributa preferred size. Posle dodavanja dugmeta, forma bi trebalo da izgleda kao na slici (Slika 48).



Slika 48: Dugme je dodato na formu



Slika 49: Gotov GUI

7. Dodati novo polje za unos sa imenom PoljeZaIspisTemperature, i podesiti njegovu veličinu na 120,20 menjanjem vrednosti atributa preferred size. Podesiti atribut editable na false da bi se sprečilo ručno menjanje vrednosti ovog polja. Grafički interfejs je sada gotov (Slika 49).
8. Dodati event handler za dugme KonvertujDugme koji sadrži sledeći kod:

```

public void actionPerformed(java.awt.event.ActionEvent e) {
    double temperatura = Double.parseDouble(PoljeZaUnosTemperature.getText());
    double konvertovanaTemperatura = 0;
    KonvertorTemperature kt = new KonvertorTemperature();

    if (KonvertorPadajucalista.getSelectedItemAt() != null &&
        KonvertorPadajucalista.getSelectedItemAt().equals("Konvertuj u C"))
        konvertovanaTemperatura = kt.konvertujFUC(temperatura);

    if (KonvertorPadajucalista.getSelectedItemAt() != null &&

```

```

        KonvertorPadajucaLista.getSelectedItem().equals("Konvertuj u F"))
            konvertovanaTemperatura = kt.konvertujCUF(temperatura);

        PoljeZaIspisTemperature.setText(""+konvertovanaTemperatura);
    }
}

```

Zadatak 6

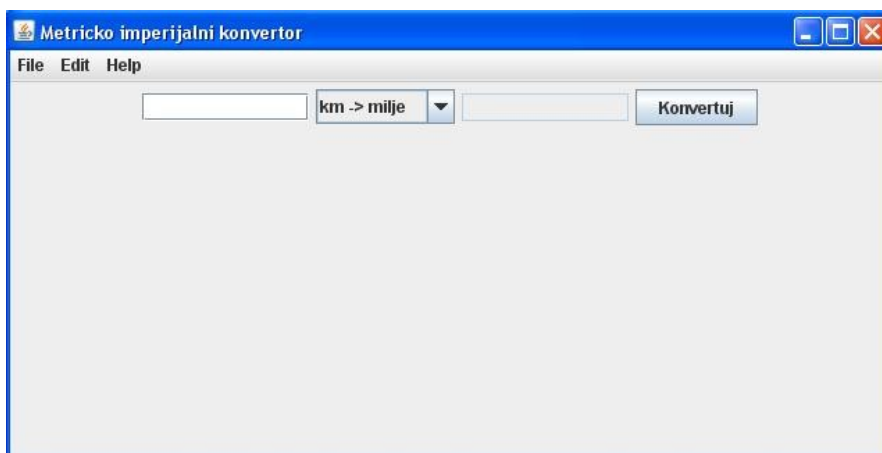
Napisati javnu klasu **MetrickoImperijalniKonvertor** koja ima:

- Javnu metodu `konvertujKmUMi` koja prima kao ulazni parametar razdaljinu izraženu u kilometrima (realan broj), pretvara je u razdaljinu izraženu u miljama i vraća kao rezultat (1 milja = 1.67 km).
- Javnu metodu `konvertujMiUKm` koja prima kao ulazni parametar razdaljinu izraženu u miljama (realan broj), pretvara je u razdaljinu izraženu u kilometrima i vraća kao rezultat (1 milja = 1.67 km).
- Javnu metodu `konvertujMUST` koja prima kao ulazni parametar razdaljinu izraženu u metrima (realan broj), pretvara je u razdaljinu izraženu u stopama i vraća kao rezultat (1 stopa = 0.32 m).
- Javnu metodu `konvertujStUM` koja prima kao ulazni parametar razdaljinu izraženu u stopama (realan broj), pretvara je u razdaljinu izraženu u metrima i vraća kao rezultat (1 stopa = 0.32 m).
- Javnu metodu `konvertujKgULb` koja prima kao ulazni parametar težinu izraženu u kilogramima (realan broj), pretvara je u težinu izraženu u librama (lb) i vraća kao rezultat (1 lb = 0.45 kg).
- Javnu metodu `konvertujLbUKg` koja prima kao ulazni parametar težinu izraženu u librama (realan broj), pretvara je u težinu izraženu u kilogramima i vraća kao rezultat (1 lb = 0.45 kg).

Napisati klasu **MetrickoImperijalniKonvertorGUI** kojom se kreira korisnički interfejs prikazan na sledećoj slici (Slika 50). Podesiti grafički interfejs tako da se, ako se promene dimenzije forme, komponente na formi raspoređuju kao na slici (Slika 51). Potrebno je da se, kada korisnik unese neku vrednost u prvo polje za unos, izabere iz padajuće liste koju vrstu konverzije želi i klikne na dugme konvertuje, u polju za ispis ispiše konvertovana vrednost zajedno sa jedinicom (45 kg, 30lb, 20 milja itd). Konverzija se vrši pozivanjem odgovarajuće metode klase **MetrickoImperijalniKonvertor**. Padajuća lista bi trebalo da ima sledeće stavke: "km -> milje", "milje -> km", "m -> stope", "stope -> m", "kg -> lb", "lb -> kg".



Slika 50: MetrickoImperijalniKonvertorGUI



Slika 51: MetrickoImperijalniKonvertorGUI - povećan

Rešenje

```

public class MetrickoImperijalniKonvertor {

```

```

public double konvertujKmUMi(double kilometri){
    return kilometri/1.67;
}

public double konvertujMiUKm(double milje){
    return milje*1.67;
}

public double konvertujMUST(double metri){
    return metri/0.32;
}

public double konvertujStUM(double stope){
    return stope*0.32;
}

public double konvertujKgULb(double kilogrami){
    return kilogrami/0.45;
}

public double konvertujLbUKg(double libre){
    return libre*0.45;
}
}

```

1. Kreirati nov grafički interfejs (vizuelnu klasu) sa imenom `MetrickoImperijalniKonvertorGUI`.
2. Promeniti naslov na formi tako da bude “Metricko imperijalni konvertor”.
3. Potrebno je podesiti formu tako da ima `FlowLayout`.
4. Dodati novo polje za unos sa imenom `PoljeZaUnos`. Izmeniti njegov atribut “preferred size” u okviru `Properties` tab-a na “120,20” da bi polje imalo odgovarajuću dužinu.
5. Dodati padajuću listu (`JComboBox` komponenta) na formu sa imenom `MIKonvertorPadajucaLista`. Promeniti vrednost atributa `preferred size` na 100,25. Padajuća lista koja je upravo dodata ne sadrži nikakve stavke, a trebalo bi da sadrži šest: “km -> milje”, “milje -> km”, “m -> stope”, “stope -> m”, “kg -> lb”, “lb -> kg”. Dodavanje stavki izvršiti upisivanjem odgovarajućeg koda u okviru editora tako da metoda `getMIKonvertorPadajucaLista` izgleda ovako:

```

private JComboBox getMIKonvertorPadajucaLista() {
    if (MIKonvertorPadajucaLista == null) {
        MIKonvertorPadajucaLista = new JComboBox();
        MIKonvertorPadajucaLista.setPreferredSize(new Dimension(100, 25));
        MIKonvertorPadajucaLista.addItem("km -> milje");
        MIKonvertorPadajucaLista.addItem("milje -> km");
        MIKonvertorPadajucaLista.addItem("m -> stope");
        MIKonvertorPadajucaLista.addItem("stope -> m");
        MIKonvertorPadajucaLista.addItem("kg -> lb");
        MIKonvertorPadajucaLista.addItem("lb -> kg");
    }
    return MIKonvertorPadajucaLista;
}

```

6. Dodati novo polje za unos sa imenom `PoljeZaIspisVrednosti`, i podesiti njegovu veličinu na 120,20 menjanjem vrednosti atributa `preferred size`. Podesiti atribut `editable` na false da bi se sprečilo ručno menjanje vrednosti ovog polja. Ako ovo polje za unos ne može da stane u prvi red, “razvući” celu formu na odgovarajuću veličinu.
7. Dodati dugme sa imenom `KonvertujDugme` i naslovom “Konvertuj”.
8. Dodati event handler za `KonvertujDugme` sa sledećim kodom:

```

public void actionPerformed(java.awt.event.ActionEvent e) {
    MetrickoImperijalniKonvertor kon =
        new MetrickoImperijalniKonvertor();
    double vrednost = Double.parseDouble(PoljeZaUnos.getText());
    double konvertovanaVrednost = 0;

    if (MIKonvertorPadajucaLista.getSelectedItem().equals("km -> milje")){
        konvertovanaVrednost = kon.konvertujKmUMi(vrednost);
        PoljeZaIspisVrednosti.setText(konvertovanaVrednost+" milja");
    }

    if (MIKonvertorPadajucaLista.getSelectedItem().equals("milje -> km")){
        konvertovanaVrednost = kon.konvertujMiUKm(vrednost);
        PoljeZaIspisVrednosti.setText(konvertovanaVrednost+" km");
    }

    if (MIKonvertorPadajucaLista.getSelectedItem().equals("m -> stope")){
        konvertovanaVrednost = kon.konvertujMUST(vrednost);
    }
}

```



```

        PoljeZaIspisVrednosti.setText(konvertovanaVrednost+" stopa");
    }

    if (MIKonvertorPadajucaLista.getSelectedItem().equals("stope -> m")) {
        konvertovanaVrednost = kon.konvertujStUM(vrednost);
        PoljeZaIspisVrednosti.setText(konvertovanaVrednost+" m");
    }

    if (MIKonvertorPadajucaLista.getSelectedItem().equals("kg -> lb")) {
        konvertovanaVrednost = kon.konvertujKgULb(vrednost);
        PoljeZaIspisVrednosti.setText(konvertovanaVrednost+" lb");
    }

    if (MIKonvertorPadajucaLista.getSelectedItem().equals("lb -> kg")) {
        konvertovanaVrednost = kon.konvertujLbUKg(vrednost);
        PoljeZaIspisVrednosti.setText(konvertovanaVrednost+" kg");
    }
}

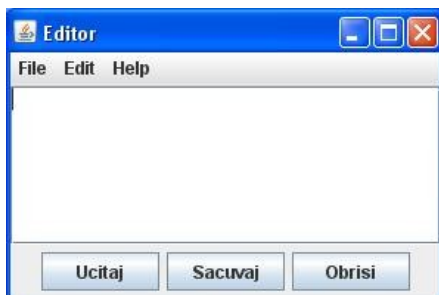
```

Zadatak 7

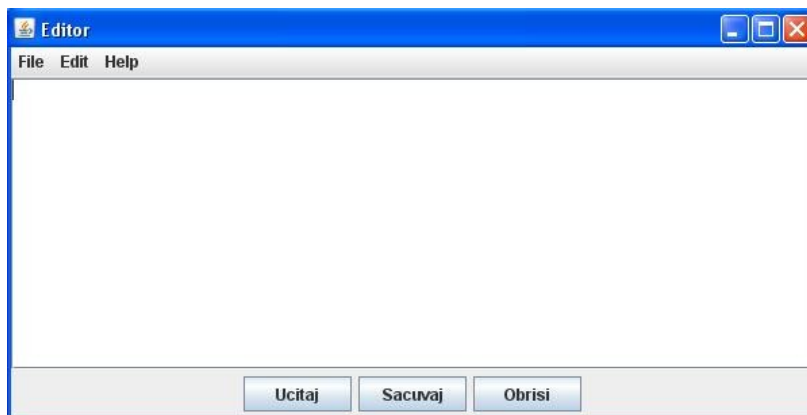
Napraviti javnu klasu TekstDemo koja ima:

- Javnu metodu koja iz tekstualnog fajla "tekst.txt" učitava sav tekst i vraća ga u vidu jednog String-a. Ako se u toku izvršavanja desi neka greška ili fajl "tekst.txt" ne postoji, potrebno je da ova metoda vrati null. Potrebno je obratiti pažnju na znak za kraj reda tj. povratni String treba da sadrži i znakove za kraj reda na kraju svakog reda.
- Javnu metodu koja kao ulazni argument prima String i upisuje ga u tekstualni fajl "tekst.txt".

Napraviti vizuelnu klasu TekstEditorGUI koja izgleda kao na slici (Slika 52). Centralni deo ekrana bi trebalo da bude editor za tekst. Kada se promeni dimenzija forme, trebalo bi da se ovaj centralni deo poveća/smanji a da dugmići ostanu na istom mestu (Slika 53). Kada se klikne na dugme "Obriši", briše se tekst koji se nalazi u editoru. Kada se klikne na dugme "Učitaj", iz fajla "tekst.txt" se učitava tekst i prikazuje u okviru editora. Kada se klikne na dugme "Sačuvaj", sav tekst koji se trenutno nalazi u editoru se upisuje u fajl "tekst.txt". Učitavanje i upisivanje je potrebno vršiti pozivanjem metoda klase TekstDemo.



Slika 52: EditorGUI



Slika 53: EditorGUI - povećan

Rešenje

```

import java.io.*;

public class TekstDemo {

    public String ucitaj(){
        try{
            BufferedReader in = new BufferedReader (new FileReader("tekst.txt"));

            boolean kraj = false;
            String tekst = "";

            while (!kraj){
                //Komanda readLine učitava red po red teksta ali ne učitava
                //i znak za kraj reda vec ga izostavlja.
                String pom = in.readLine();
                //Da bi String sadrzao i znak za kraj reda na odgovarajucem
            }
        }
    }
}

```

```

        //mestu potrebno ga je rucno ubaciti na kraj svakog reda ("\\n").
        if (pom == null) kraj = true;
        else tekst=tekst+pom+"\\n";
    }

    in.close();
    return tekst;
} catch (Exception e) {
    return null;
}

}

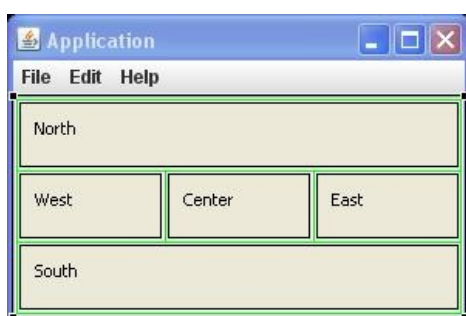
public void upisi(String tekst) {
    try {
        PrintWriter out = new PrintWriter(new FileWriter("tekst.txt"));

        out.write(tekst);

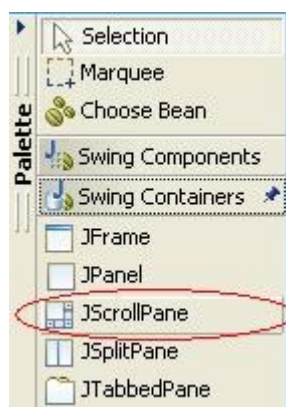
        out.close();
    } catch (Exception e) {
        System.out.println("GRESKA: "+e.getMessage());
    }
}
}

```

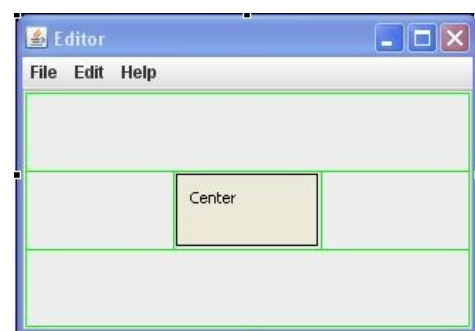
1. Kreirati nov grafički interfejs (vizuelnu klasu) sa imenom `TekstEditorGUI`.
2. Promeniti naslov na formi tako da bude "Editor".
3. Potrebno je podesiti formu tako da ima **BorderLayout**. Ovo je treća vrsta layout-a i karakteriše se time što formu deli na pet nezavisnih delova u koje se smeštaju komponente (Slika 54): severni (North), južni (South), istočni (East), zapadni (West) i centralni (Center). Princip funkcionisanja je sledeći: ako se neka komponenta postavi u severni deo, ona će uvek da ostane iste veličine i "zalepljena" za gornju ivicu forme bez obzira na to kako i koliko se promene dimenzije forme. Svaka komponenta koja se postavi u istočni deo će uvek da ostane iste veličine i "zalepljena" za desnu ivicu forme bez obzira na to kako i koliko se promene dimenzije forme. Isto važi i za južni i zapadni deo. Ali, ako se komponenta postavi u centralni deo forme, ona će da menja veličinu u zavisnosti od promene veličine forme. Ako se pogleda izgled traženog interfejsa kada je normalne veličine (Slika 52) i kada je maksimizovan (Slika 53), vidi se da je potrebno da tri dugmeta ostanu uz donju ivicu prozora čak iako se prozor poveća, a da se centralni deo u koji će se unositi tekst poveća zajedno sa prozorom. To znači da će sva tri dugmeta morati biti postavljena u južnom delu border layout-a (da bi stalno ostali pri dnu ekrana), a da će komponenta za unos teksta da bude u centralnom delu (da bi se automatski povećavala i smanjivala sa promenom veličine forme).



Slika 54: BorderLayout



Slika 55: JScrollPane



Slika 56: Dodavanje JScrollPane-a

4. Kliknuti na Swing containers u okviru palete pa izabrati `JScrollPane` komponentu (Slika 55). Dodati ovu komponentu u centralni deo border layout-a (Slika 56). Dodeliti joj bilo koje ime.
5. Kliknuti nazad na Swing components i izabrati `JTextArea`. Ovo je komponenta koja omogućava unos, izmenu i brisanje teksta na više linija. Za razliku od nje, `JTextField` omogućava unos teksta u jednoj liniji. Dodati ovu komponentu na formu i dati joj ime `Editor`. Primetićete da će automatski da se razvuče do svih ivica ekrana (Slika 57). Ono što se zaista desilo je sledeće: `JScrollPane` komponenta omogućava automatsko dodavanje strelica za skrolovanje u slučaju da tekst prevaziđe granice editora, dok `JTextArea` predstavlja sam editor. Da bi editor imao ove strelice za skrolovanje, potrebno je da se na formu (tj. njen `ContentPane`) prvo stavi `JScrollPane`, pa tek onda na njega `JTextArea`. Drugim rečima, `JScrollPane` je direktno na formi, a `JTextArea` je na `JScrollPane`. Ovaj odnos se može videti i na Java Beans tab-u (Slika 58).



Slika 57: JTextArea je dodat

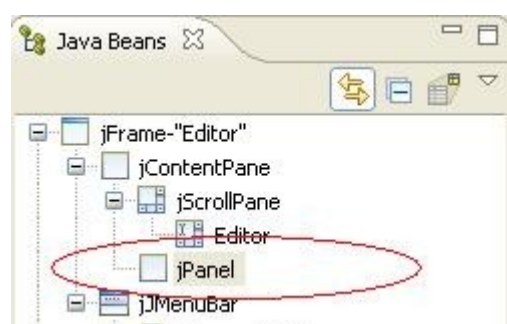


Slika 58: JScrollPane je na formi (Content pane), a JTextArea (Editor) je na JScrollPane

6. Kliknuti na Swing containers u okviru palete pa izabrati JPanel komponentu. Ova komponenta predstavlja jednu običnu površinu (panel) na koju se mogu smestati komponente. Dodavanje se može izvršiti direktno u okviru Java Beans taba tako što se klikne na jContentPane (Slika 59). Panelu dati bilo koje ime. Kada je panel dodat, on se pojavljuje kao element jContentPane-a (Slika 60). Na ovaj panel će kasnije da se dodaju tri tražena dugmeta. Panel je trenutno smešten na severni deo border layout-a, pa je to potrebno izmeniti. Promeniti panelov atribut constraint tako da ima vrednost "South" a ne "North".

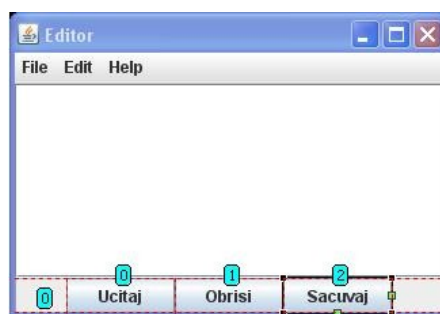


Slika 59: Dodavanje komponente u Java Beans tab-u

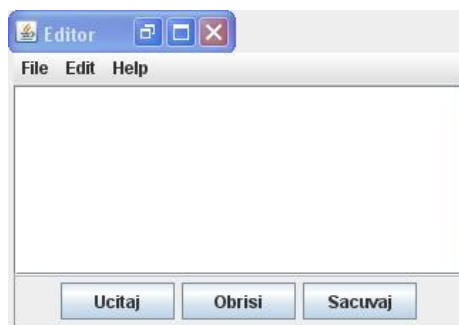


Slika 60: Panel je dodat

7. Dodati dugme sa imenom UcitajDugme i naslovom "Ucitaj". Dugme dodati klikom na ikonu JPanel u okviru Java Beans tab-a.
8. Po istom principu dodati još dva dugmeta: ObrisiDugme (naslov "Obrisi") i SacuvajDugme (naslov "Sacuvaj"). Podesiti velicinu sva tri dugmeta menjanjem atributa preferred size na "80,26". Forma bi trebalo da izgleda kao na slici (Slika 61). JPanel komponenta (na njoj se nalaze dugmići) je takva da ima svoj layout. Drugim rečima, komponente na panelu mogu biti raspoređene prema null, flow, border ili drugom layout-u. Potrebno je podesiti da panel ima flow layout. To se radi tako što se klikne desnim tasterom miša na sam panel i izabere se Set layout -> FlowLayout. Posle ovih izmena, forma je gotova i izgleda kao na slici (Slika 62).



Slika 61: Dugmići su dodati



Slika 62: TekstEditorGUI gotov

9. Dodati event handler za UcitajDugme:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    TekstDemo td = new TekstDemo();

    String tekst = td.ucitaj();

    Editor.setText(tekst);
}
```

10. Dodati event handler za ObrisiDugme:

```
public void actionPerformed(java.awt.event.ActionEvent e) {  
    Editor.setText(null);  
}
```

11. Dodati event handler za SacuvajDugme:

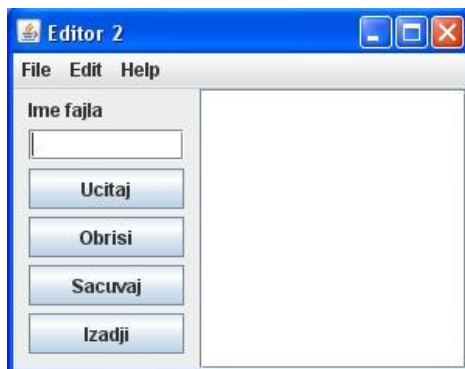
```
public void actionPerformed(java.awt.event.ActionEvent e) {  
    TekstDemo td = new TekstDemo();  
  
    String tekst = Editor.getText();  
  
    td.upisi(tekst);  
}
```

Zadatak 8

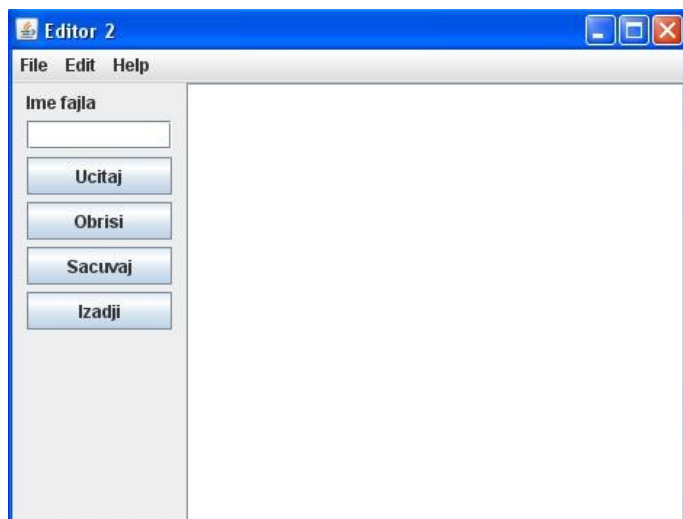
Napraviti javnu klasu TekstDemo2 koja ima:

- Javnu metodu koja kao ulazni argument prima ime tekstualnog fajla. Ova metoda učitava sav tekst iz ovog fajla i vraća ga u vidu jednog String-a. Ako se u toku izvršavanja desi neka greška ili fajl ne postoji, potrebno je da ova metoda vrati null. Potrebno je obratiti pažnju na znak za kraj reda tj. povratni String treba da sadrži i znakove za kraj reda na kraju svakog reda.
- Javnu metodu koja kao ulazni argument prima dva String-a: prvi predstavlja ime tekstualnog fajla, a drugi neki tekst. Ova metoda upisuje ga tekst u tekstualni fajl sa zadatim imenom.

Napraviti vizuelnu klasu TekstEditor2GUI koja izgleda kao na slici (Slika 63). Centralni deo ekrana bi trebalo da bude editor za tekst. Kada se promeni dimenzija forme, trebalo bi da se ovaj centralni deo poveća/smanji a da ostale komponente ostanu na istom mestu (Slika 64). Kada se klikne na dugme "Obrisi", briše se tekst koji se nalazi u editoru. Kada se klikne na dugme "Učitaj", iz polja za unos se preuzima ime tekstualnog fajla, učitava se tekst iz tog fajla i prikazuje u okviru editora. Kada se klikne na dugme "Sačuvaj", sav tekst koji se trenutno nalazi u editoru se upisuje u fajl čije ime se nalazi u polju za unos. Učitavanje i upisivanje je potrebno vršiti pozivanjem metoda klase TekstDemo2. Kada se klikne na dugme "Izadji", prekida se izvršavanje programa.



Slika 63: TekstEditor2GUI



Slika 64: TekstEditor2GUI - povećan

Rešenje

```
import java.io.*;  
  
public class TekstDemo2 {  
  
    public String ucitaj(String fajl){  
        try{  
            BufferedReader in = new BufferedReader (new FileReader(fajl));  
  
            boolean kraj = false;  
            String tekst = "";
```



```

        while (!kraj){
            String pom = in.readLine();
            if (pom == null) kraj = true;
            else tekst=tekst+pom+"\n";
        }

        in.close();
        return tekst;
    } catch (Exception e){
        return null;
    }
}

public void upisi(String fajl, String tekst){
    try{
        PrintWriter out = new PrintWriter(new FileWriter(fajl));

        out.write(tekst);

        out.close();
    } catch (Exception e){
        System.out.println("GRESKA: "+e.getMessage());
    }
}
}

```

1. Kreirati nov grafički interfejs (vizuelnu klasu) sa imenom `TekstEditor2GUI`.
2. Promeniti naslov na formi tako da bude "Editor 2".
3. Potrebno je podesiti formu tako da ima `BorderLayout`.
4. Kliknuti na Swing containers u okviru palete, izabrati `JScrollPane` komponentu i dodati je u centralni deo border layout-a.
5. Kliknuti nazad na Swing components, izabrati `JTextArea`, dodati je na formu (tj. na `ScrollBar`) i dati joj ime `Editor`.
6. Kliknuti na Swing containers u okviru palete, izabrati `JPanel` komponentu i dodati je direktno u okviru Java Beans taba klikom na `jContentPane`. Panelu dati bilo koje ime. Panel je trenutno smešten na severni deo border layout-a, pa je to potrebno izmeniti. Promeniti panelov atribut `constraint` tako da ima vrednost "West" a ne "North" (jer se traži da ostale komponente budu uz levu ivicu forme).
7. Dodati na panel labelu sa bilo kojim imenom i naslovom "Ime fajla". Labelu dodati klikom na ikonu `JPanel` u okviru Java Beans tab-a. Podesiti veličinu labele na 100,16.
8. Dodati na panel polje za unos teksta sa imenom `PoljeZaUnosImena`. Polje dodati klikom na ikonu `JPanel` u okviru Java Beans tab-a ili direktno klikom na odgovarajuće mesto na formi. Podesiti veličinu polja na 100,20.
9. Dodati dugme sa imenom `UcitajDugme` i naslovom "Ucitaj". Dugme dodati klikom na ikonu `JPanel` u okviru Java Beans tab-a ili direktno klikom na odgovarajuće mesto na formi. Podesiti veličinu dugmeta na 100,26.
10. Po istom principu dodati još tri dugmeta: `ObrisiDugme` (naslov "Obrisi"), `SacuvajDugme` (naslov "Sacuvaj") i `IzadjiDugme` (naslov "Izadji"). Podesiti velicinu sva tri dugmeta na 100,26. Takođe, potrebno je podesiti da panel ima flow laoyut. Kada ovo uradite, komponente na panelu će da se rasporede na nepravilan način. Da bi se to rešilo, potrebno je podesiti da panel bude veličine 120,36. Po potrebi, "razvući" celu formu tako da se vide svi dugmići. Posle ovih izmena, forma je gotova i izgleda kao na slici (Slika 65).



Slika 65: `TekstEditor2GUI` - gotov

11. Dodati event handler za `UcitajDugme`:

```

public void actionPerformed(java.awt.event.ActionEvent e) {
    TekstDemo2 td = new TekstDemo2();
}

```

```
String ime = PoljeZaUnosImena.getText();
String tekst = td.ucitaj(ime);

Editor.setText(tekst);
}
```

12. Dodati event handler za ObrisiDugme:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    Editor.setText(null);
}
```

13. Dodati event handler za SacuvajDugme:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    TekstDemo2 td = new TekstDemo2();

    String tekst = Editor.getText();
    String ime = PoljeZaUnosImena.getText();

    td.upisi(ime, tekst);
}
```

14. Dodati event handler za IzadjiDugme:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
    System.exit(0);
}
```