

**NAPOMENA:** Vežbe podrazumevaju da je student ovladao teorijom iz dela "Swing" (2-Prostorni raspored i događaji.ppt, 3-Komponente.ppt).

## Meniji u Java aplikacijama

Za dodavanje menija i podmenija na JFrame na raspolaganju su sledeće klase:

*JMenuBar* – Glavni meni aplikacije. Jedan *JFrame* može da ima samo jedan *JMenuBar*.

*JMenu* – Vertikalni podmeniji

*JMenuItem* – Stavke menija

U nastavku je dat primer:

```
// Kreiranje glavnog menija aplikacije
JMenuBar mainMenu = new JMenuBar();

// File meni
JMenu fileMenu = new JMenu("File");
// "Open" stavka menija
JMenuItem openMenuItem = new JMenuItem("Open...");
// Ctrl + O precica za stavku "Open"
openMenuItem.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
                                                    InputEvent.CTRL_MASK));
// Dodavanje stavke u meni
fileMenu.add(openMenuItem);

// Podmeni "insert"
JMenu insertMenu = new JMenu("Insert");

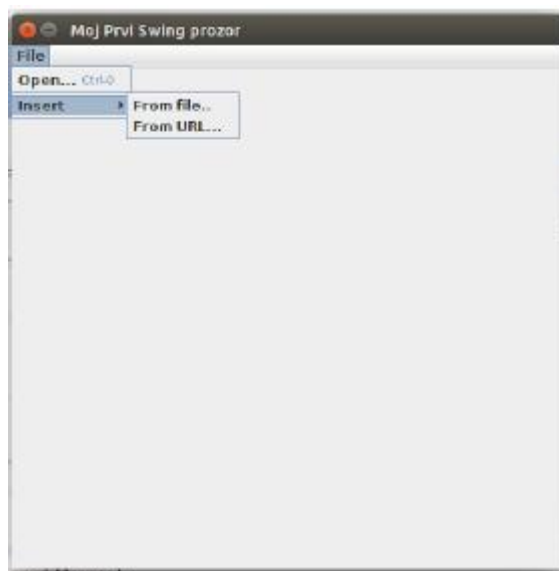
// Stavke podmenija
JMenuItem fileItem = new JMenuItem("From file..");
insertMenu.add(fileItem);
JMenuItem urlItem = new JMenuItem("From URL...");
insertMenu.add(urlItem);

// Dodavanje separatora
fileMenu.addSeparator();
// Dodavanje podmenija u meni
fileMenu.add(insertMenu);

// Dodavanje menija "File" u glavni meni
mainMenu.add(fileMenu);
// Postavljanje glavnog menija aplikacije
setJMenuBar(mainMenu);
```

Testiranje i odabrani paterni

Rezultat primera:



## Upravljanje prostornim rasporedom komponenti

Upravljanje prostornim rasporedom komponenti na kontejnerima (*Layout*-ima) se vrši uz pomoć klasa koje se zovu *Layout manager*-i. U zavisnosti od toga kakav raspored komponenti želimo, odabraćemo jedan od raspoloživih menagera. *Layout manager*-i se postavlja pozivom metode *setLayout()* klase *JFrame*. Uz *Swing* biblioteku dolazi nekoliko predefinisanih upravljača rasporedom komponenti. Neki od njih su navedeni u nastavku.

Više detalja o svim komponentama se može pronaći u zvaničnoj dokumentaciji na adresi <https://docs.oracle.com/javase/8/docs/api/index.html>

### ***BorderLayout***

Služe komponente po stranama sveta:



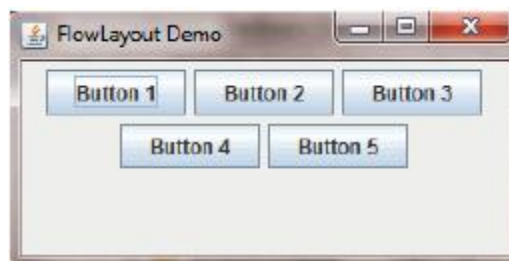
Prvo podesi veličinu severne i južne komponente uzimajući preferiranu visinu (postavljenu metodom *setPreferredSize()*), a za širinu uzima širinu kontejnera. Zatim podesi veličinu istočne i zapadne, poštujući preferiranu širinu, dok visinu proširi na ceo preostali prostor. Ostatak u potpunosti dodeljuje centralnoj komponenti. *BorderLayout* je podrazumevani *Layout manager* za *JFrame* kontejnere.

Primer upotrebe:

```
setLayout(new BorderLayout());  
add(new JButton("Novo dugme"), BorderLayout.NORTH);
```

### ***FlowLayout***

Služe komponente u red po horizontali poštujući preferirane veličine.



Kada više nema mesta, prelazi u novi red. Komponente mogu da se poravnaju u levo, desno ili centrirano. *FlowLayout* je podrazumevani manager za *JDialog* kontejnere.

Primer upotrebe:

```
setLayout(new FlowLayout());  
add(new JButton("Novo dugme"));
```

## ***GridLayout***

Raspoređuje komponente u mrežu sa zadatim brojem redova i kolona.



Ne poštuje preferirane, minimalne ni maksimalne veličine komponenti, već svim ćelijama dodeljuje iste dimenzije. Pozicija komponente (red i kolona) se određuju na osnovu redosleda ubacivanja komponente u kontejner.

Primer upotrebe:

```
setLayout(new GridLayout(4, 3));  
add(new JButton("Prvo"));  
add(new JButton("Drugo"));  
// ...
```

## ***GridBagLayout***

Kompleksnija i fleksibilnija varijanta *GridLayout*-a.



*GridBagLayout* omogućava da kolone budu različitih veličina kao i to da komponente zauzimaju više od jedne ćelije. Sadrži više opcija za raspored komponenti nego opšti *GridLayout*.

Primer upotrebe:

```
setLayout(new GridBagLayout());
GridBagConstraints con = new GridBagConstraints();
JLabel lblId = new JLabel("Id");
con.gridx = 0;
con.gridy = 0;
con.insets = new Insets(10, 10, 10, 10);
add(lblId, con);
```

## MigLayout

Razvijen od strane nezavisne firme MiG InfoCom. Ne dolazi sa *Swing* bibliotekom, već mora da se preuzme sa veb stranice (<http://www.miglayout.com/>) u uključi u projekat. Omogućava vrlo jednostavno kreiranje proizvoljnih rasporeda komponenti. Komponente smešta u ćelije matrice po uzoru na *GridLayout* i *GridBagLayout*. Napomena: na pomenutoj veb stranici se nalazi dokument sa uputstvom za upotrebu.

## JTable komponenta

*JTable* predstavlja *Swing* komponentu za tabelarni prikaz podataka. Podacima u *JTable* tabeli se upravlja instancom klase *TableModel* (najčešće se radi o implementaciji *DefaultTableModel*). *TableModel* je zadužen za inicijalizaciju podataka u tabeli, kao i za rukovanje izmenama.

Dobra praksa je ne dodavati *JTable* direktno u kontejner, već to uraditi putem *JScrollPane* instance kako bi, kada tabela posatane prevelika za prikaz svih podataka dobili mogućnost skrolovanja.

Primer upotrebe:

```
// Inicijalizacija podataka:
// Sadržaj zaglavlja tabele se definiše putem niza (ili Vector-a)
String[] zaglavlja = new String[] { "Ime", "Prezime", "Broj indeksa" };
// Sadržaj same tabele je matrica
Object[][] sadrzaj = new Object[5][zaglavlja.length];

for(int i=0; i<sadrzaj.length; i++) {
    sadrzaj[i][0] = "Ime";
    sadrzaj[i][1] = "Prezime";
    sadrzaj[i][2] = "Indeks";
}

// Instanciranje TableModel objekta sa definisanim podacima
DefaultTableModel model = new DefaultTableModel(sadrzaj, zaglavlja);
// Instanciranje tabele
JTable tabela = new JTable(model);

// Neka standardna podesavanja JTable komponente:
// Dozvoljeno selektovanje redova
tabela.setRowSelectionAllowed(true);
// Ali ne i selektovanje kolona
tabela.setColumnSelectionAllowed(false);
// Dozvoljeno selektovanje samo jednog reda odjednom
tabela.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
// Onemogućeno je direktno editovanje sadržaja u ćelijama
tabela.setDefaultEditor(Object.class, null);

// Dodavanje JScrollPane panela sa tabelom u kontejner
JScrollPane scrollPane = new JScrollPane(tabela);
add(scrollPane);
```

Dodavanje, izmena i brisanje se može vršiti pomoću apstraktnog *DefaultTableModel* objekta:

```
// Dodavanje novog reda u tabelu
DefaultTableModel model = (DefaultTableModel)tabela.getModel();
model.addRow(new String[] {"Ime", "Prezime", "Broj indeksa"});

// Uklanjanje reda se vrši putem indeksa
model.removeRow(5);

// Trenutno selektovani red se može dobiti metodom getSelectedRow()
// Metoda vraća -1 ako nista nije selektovano
int selektovaniRed = tabela.getSelectedRow();

String jednoIme = (String) tabela.getValueAt(1, 0);
tabela.setValueAt("Novo ime", 1, 0);
```

Međutim, u većini slučajeva je potrebno izmeniti podrazumevana podešavanja i stoga se kreira zasebna klasa koja proširuje *AbstractTableModel* ili *DefaultTableModel* klasu. Na ovaj način se podaci mogu čuvati u različitim tipovima kolekcija i mapa, i dopuniti funkcionalnosti.

```
public class IgracModel extends AbstractTableModel {

    private static final long serialVersionUID = -5519372712630599241L;

    private String[] columnNames = { "Id", "Ime", "Prezime",
        "Klub"};

    private List<Igrac> data = new ArrayList<Igrac>();

    public IgracModel(Map<Integer, Igrac> igraci) {
        for (Igrac i : igraci.values())
            data.add(i);
    }

    @Override
    public int getColumnCount() {
        return columnNames.length;
    }

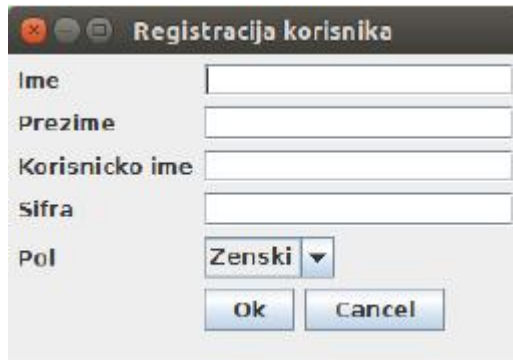
    @Override
    public int getRowCount() {
        return data.size();
    }

    @Override
    public String getColumnName(int col) {
        return columnNames[col];
    }

    @Override
    public Object getValueAt(int row, int col) {
        Igrac i = data.get(row);
        return i.toCell(col);
    }
}
```

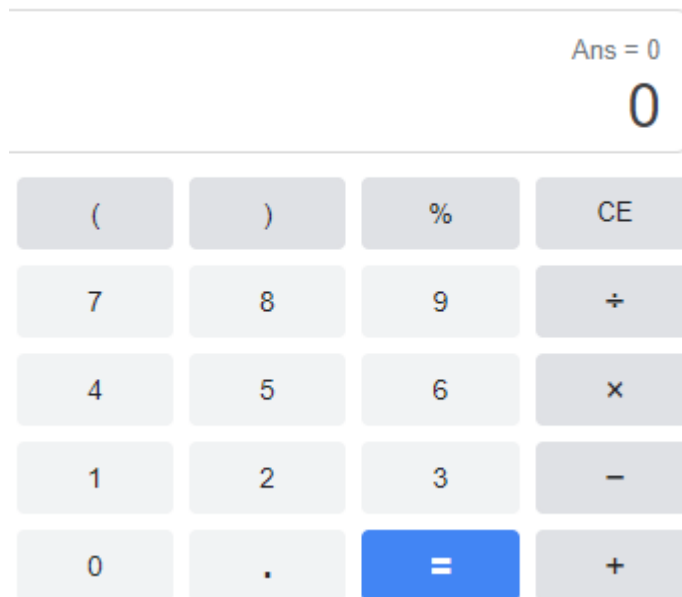
## Zadaci

- Zadatak 1. Kreirati *Java* formu sa glavnim menijem sa prve slike. Pored menija *File* dodati meni *Edit* sa stavkama *Cut*, *Copy* i *Paste*.
- Zadatak 2. Upotrebom MigLayout upravljača kreirati prozor sa rasporedom komponenti kao na slici:



The screenshot shows a Java Swing window titled "Registracija korisnika". It contains five text input fields labeled "Ime", "Prezime", "Korisnicko ime", and "Sifra". Below these is a dropdown menu for "Pol" with "Zenski" selected. At the bottom are "Ok" and "Cancel" buttons.

- Zadatak 3. Napraviti aplikaciju digitrona za operacije predstavljene na slici u nastavku. Omogućiti da se sve operacije unesu i putem tastature (namapirati tastere).



The screenshot shows a digital calculator application. It has a display area at the top showing "Ans = 0" and a large "0". Below the display is a grid of buttons: parentheses "(", ")", percentage "%", and "CE"; digits "7", "8", "9", and division "÷"; digits "4", "5", "6", and multiplication "×"; digits "1", "2", "3", and subtraction "-"; digits "0", decimal ".", a blue equals "=" button, and addition "+".