



# GUI

2016/17

# GUI – GRAPHICAL USER INTERFACE

Aplikacije koje imaju samo tekstualni ulaz i izlaz

- nazivaju se konzolnim aplikacijama,
- upravljaju celim ekranom (odnosno prozorom koji simulira ceo ekran konzole).

Aplikacija sa grafičkim korisničkim interfejsom

- ne upravlja celim ekranom, već prozorima koje kreira,
- izuzetno može upravljati i celim ekranom (full screen mode),
- izlazni podaci se prikazuju u prozorima,
- ulazni podaci generišu događaje u prozorima.

Java je prvi široko rasprostranjeni programski jezik koji na standardan način podržava programiranje GUI.

# PAKETI

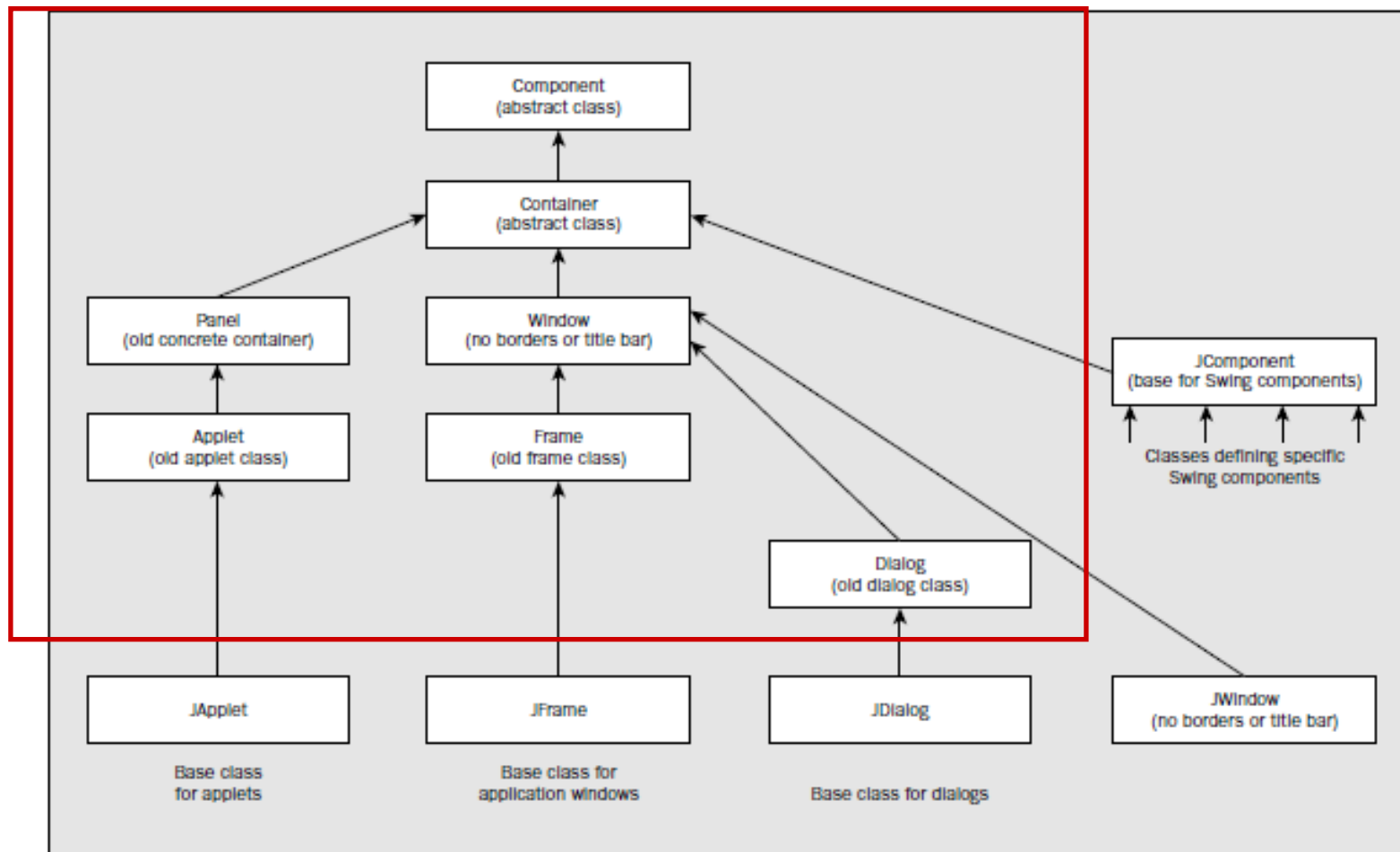
- Podrška za programiranje GUI-a nalazi se u `java.awt` paketu.
- Nešto noviji paket za programiranje GUI: `javax.swing`
  - sadrži neke nove i neke poboljšane komponente u odnosu na `java.awt`
  - pisan je u Javi pa je prenosivost potpuna
- `javafx`
- AWT je skraćenica od *Abstract Windowing Toolkit*
  - apstraktni alati za rad sa prozorima
  - apstraktni - ne zavise od konkretne platforme
- `java.awt` paket sadrži klase i interfejse koji podržavaju izlazne i ulazne aspekte GUI
- `java.awt` paket se koristi za programiranje
  - samostalnih aplikacija
  - apleta

# KOMPONENTE

- Bez obzira koji se paket koristi, struktura GUI-a je u osnovi ista.
- Ceo GUI se sastoji iz tzv. **grafičkih komponenti**.
- Grafičke komponente koje se pojavljuju na ekranu nazivaju se i **kontrolne** ili "**vidžiti**"  
ekranski tasteri (button), radio-dugmad (radio-button), polja za potvrde (checkbox), klizači(scrollbar), polja za tekst (text box), liste (list), padajuće liste (combo-box, choice)
- Svaki deo grafičkog interfejsa je implementiran kao **klasa nekog GUI paketa**.

# DEO STRUKTURE GUI KLASA

Komponenta je grafički entitet odgovarajućeg tipa koji može biti prikazan na ekranu i u Javi predstavlja objekat bilo koje klase koja je potklasa klase **Component**.



# DEO STRUKTURE GUI KLASA

- Klasa **Component**/**JComponent** je zajednička osnovna klasa za sve GUI kontrole i reprezentuje nešto što:
  - ima poziciju, veličinu, može se iscrtati na ekranu i prihvata ulazne događaje
- Klasa **Container** je izvedena iz Component
  - objekat Container može da sadrži druge komponente

Objekti svih klasa izvedenih iz klase Container mogu da sadrže druge objekte iz bilo koje klase koja je izvedena iz klase Component. Kako Container nasleđuje Component, sledi da **kontejneri mogu da sadrže druge kontejnere**. **Izuzetak je klasa Window i njene potklase**, jer one ne mogu biti sadržane u drugom kontejneru.

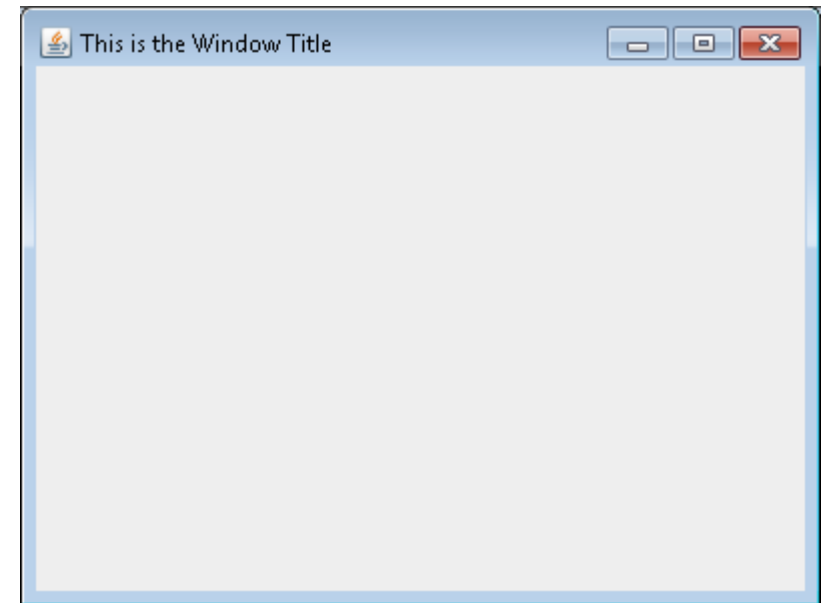
- Klasa **Window** je izvedena iz **Container** (sadrži komponente)
  - njeni objekti su prozori najvišeg nivoa bez linije naslova i ikona za upravljanje
  - nisu sadržani u drugim objektima
  - obuhvata metode za rad sa prozorima
- **JWindow** – swing verzija Window klase. Obično se koristi kao pomoćni prozor aplikacije.

# DEO STRUKTURE GUI KLASA

- Klasa **Frame** je izvedena iz Window
  - koristi se za kreiranje glavnog prozora aplikacije
  - objekat tipa Frame može da sadrži traku menija i da prikaže naslov
- **JFrame** – swing verzija Frame klase.
- Klasa **Dialog** je izvedena iz Window
  - koristi se za kreiranje prozora dijaloga
  - dijalog ima roditeljski prozor (dijalog nestaje sa ekrana kad se roditelj minimizuje)
  - može da bude **modalan** ili ne
  - ne sadrži traku menija
- **JDialog** swing verzija Dialog klase.
- **JApplet** – osnovna klasa za Java 2 applet – applet je program napravljen da radi kada je ugrađen u web-stranu.

# DEO STRUKTURE GUI KLASA

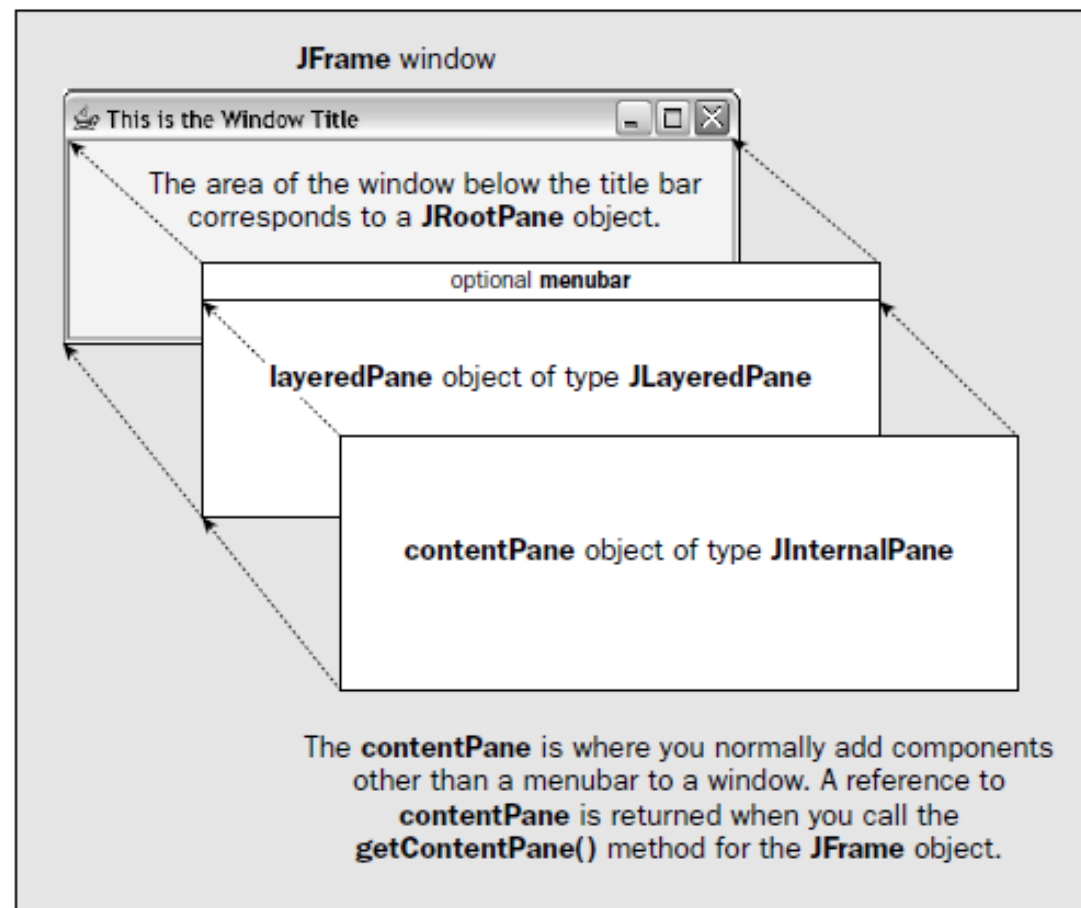
```
import javax.swing.JFrame;
public class TryWindow {
    public static void main(String[] args) {
        JFrame aWindow = new JFrame("This is the Window Title");
        aWindow.setBounds(0,0,400,300);
        aWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        aWindow.setVisible(true);
    }
}
```



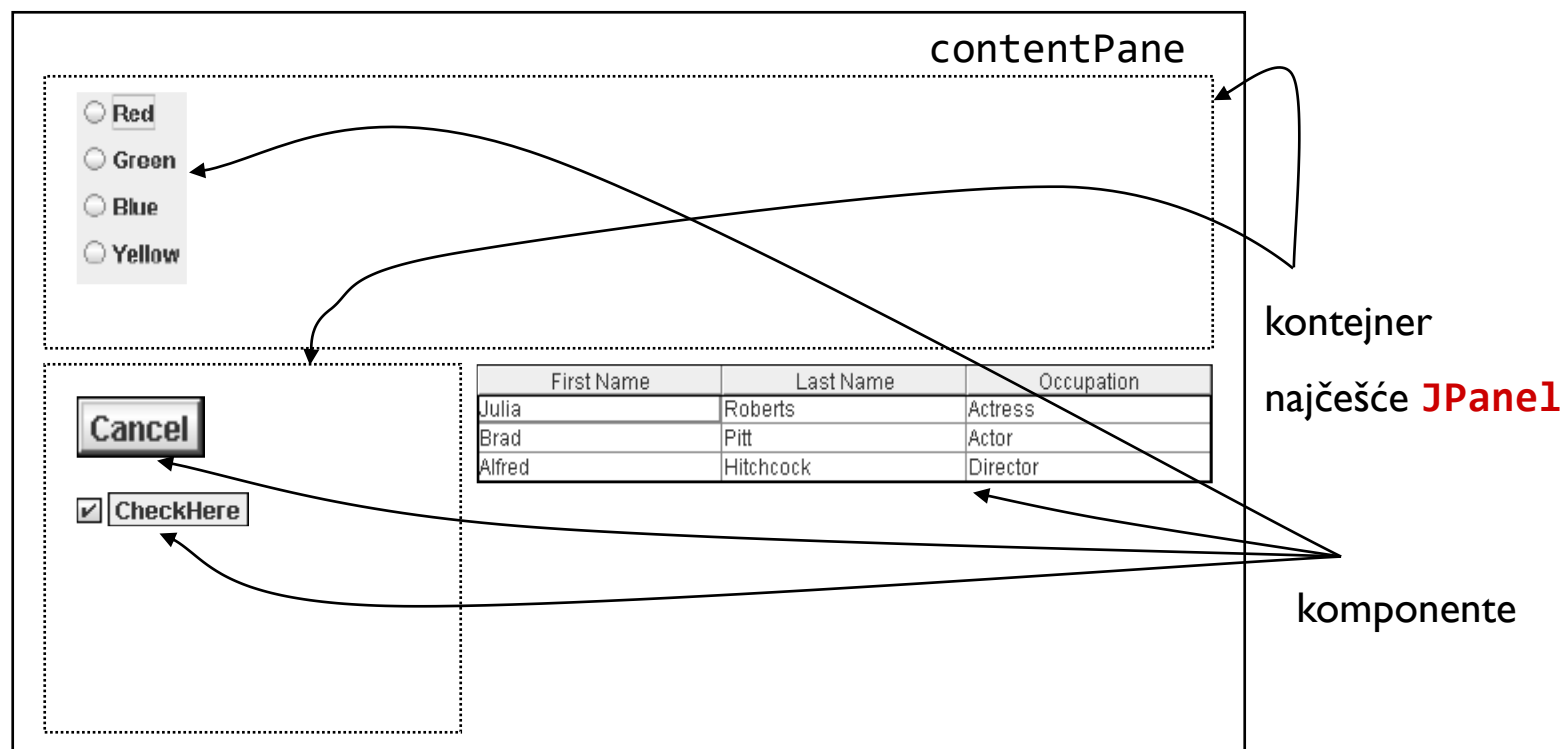


# SLAGALICA

- Samostalna GUI (swing) aplikacija, što se grafičkog dela tiče, najčešće sadrži jedan JFrame u koji se smeštaju kontrole.
- **contentPane** je Container koji zauzima centralni deo prozora i u njega se, najčešće, izuzimajući menije, smeštaju komponente (dugmići, padajuće liste itd.)



# SLAGALICA



- Kada se dodaju na prozor (JFrame), komponente se najčešće implementiraju kao atributi klase koja predstavlja prozor.

# SWING KOMPONENTE



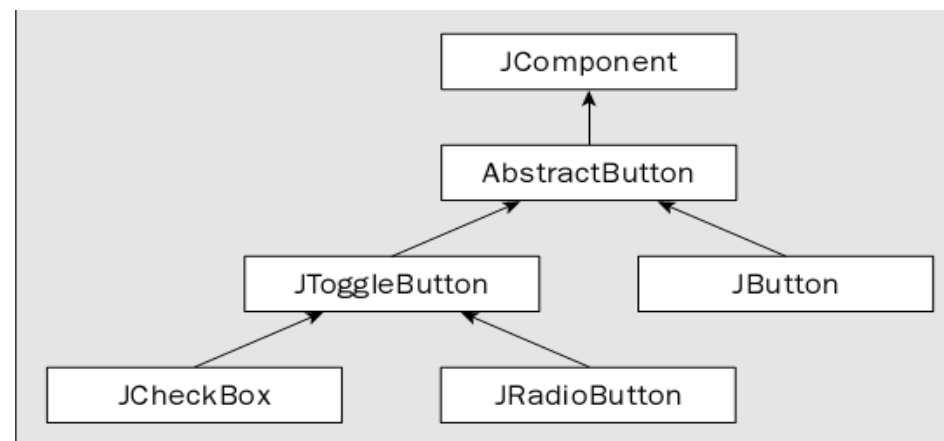
- **JButton** – regularno dugme, npr. OK/Cancel.

- **JToggleButton** – definiše dugme sa dva stanja – pritisnuto ili ne i iz njega su izvedeni:

- **JCheckBox** – dugme sa checkbox-om sa leve strane



- **JRadioButton** – dugmad koja operišu u grupi, tako da je samo jedno dugme pritisnuto u datom trenutku. Ovo grupisanje se postiže dodavanjem JRadioButton objekta ButtonGroup objektu



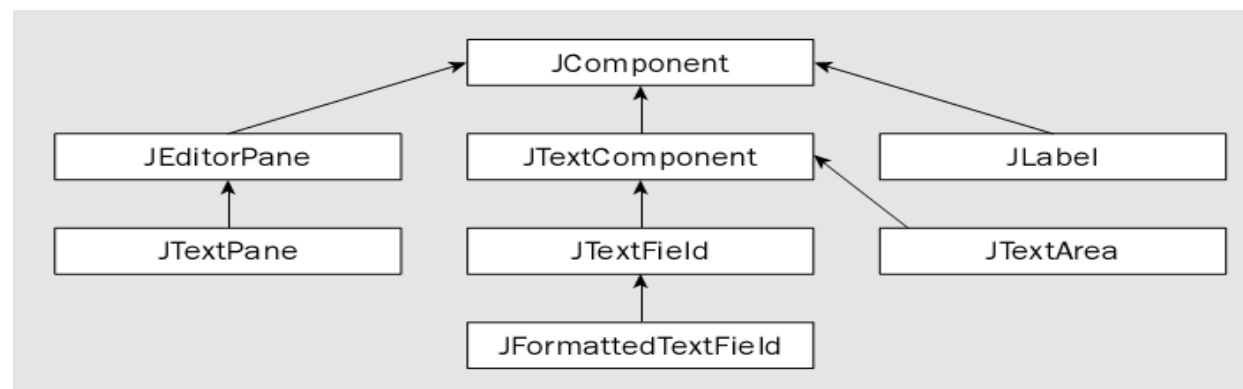
# SWING TEKST KOMPONENTE

I am a label

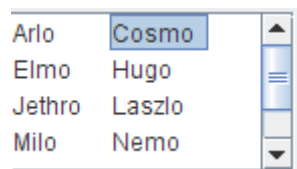
10-Apr-2004

'Tis a dog's delight to bark and bite  
And little birds to sing,  
And if you sit on a red hot brick,  
'Tis a sign of an early spring!

- **JLabel** – pasivna komponenta, ne može se editovati, služi za označavanje drugih komponenti
- **TextField** – kao JLabel samo se može editovati. Predstavlja jednu liniju teksta.
- **FormattedTextField** – to je TextField komponenta koja ima kontrolu formata podataka koji se unose/prikazuju.
- **TextArea** – komponenta koja dozvoljava unos višelinijskog teksta. Ne podržava direktno skrolovanje, ali to se može postići stavljanjem komponente TextArea u kontejner JScrollPane
- **EditorPane** i **TextPane** podržavaju sofisticiranije editovanje (HTML, slike itd).



# SWING KOMPONENTE



- **JList** – definiše ograničenu listu stavki
- **JTable** – definiše tabelu u kojoj se može selektovati kolona, vrsta ili pojedinačni element. Osim toga ova klasa automatski vodi računa o prevlačenju kolona mišem na novu poziciju.
- **javax.swing.borders** – sadrži osam klasa koje definišu različite vrste okvira za komponente

First Name	Last Name	Occupation
Julia	Roberts	Actress
Brad	Pitt	Actor
Alfred	Hitchcock	Director

# RASPOREĐIVANJE KOMPONENTI - KONTEJNERI

- Osnovno ‘sredstvo’ organizacije komponenti.
- Kontejner je objekat svakog tipa koji kao osnovnu klasu ima Container klasu.
- U kontejner sadrži:
  1. komponente

Komponente unutar kontejnera se prikazuju unutar površi koju zauzima kontejner na ekranu.  
Komponente su smeštene u listu, pa imaju svoje indekse

    - `int getComponentCount()` – vraća broj komponenti u kontejneru
    - `Component getComponent(int index)` – vraća komponentu koja se identifikuje datim indeksom. Radi se o indeksu niza, pa mora biti u opsegu `[0, count-1]`.
    - `Component[ ] getComponents()` – vraća niz komponenti iz kontejnera
    - Nekoliko verzija `add()` metoda – dodavanje komponente kontejneru
  2. definisanog menadžera postavke

Menadžer postavke je objekat neke od klasa tipa **Layout Manager**. On sadrži opis načina raspoređivanja komponenti u kontejneru.

    - `void setLayout(LayoutManager mgr)`
    - `LayoutManager getLayout()`

# DODAVANJE KOMPONENTE KONTEJNERU

- Klasa `Container` sadrži `add()` metod kojim se komponente dodaju kontejneru:
  - `Component add(Component c)` – dodaje komponentu `c` na kraj liste komponenti sačuvane u kontejneru. Vraća se `c`.
  - `Component add(Component c, int index)` – komponenta se smešta na datu poziciju. Ako je `index` -1 komponenta se dodaje na kraj liste, u suprotnom indeks mora biti ne manji od 0 i manji od trenutnog broja komponenti u kontejneru. Vraća se `c`.
  - postoje još 2 verzije

```
this.add(new Label("Zdravo, svete!"));
```

# RASPOREĐIVANJE

- Svi kontejneri imaju podrazumevani layout manager, ali se može izabrati drugačiji kada je potrebno.
- Klase koje definišu layout manager se nalaze u paketima `java.awt` i `javax.swing.imemanager`
- **Layout manager određuje poziciju i veličinu svih komponenti u kontejneru.**
  - Klase koje definišu layout manager-e implementiraju interfejs `LayoutManager`, tako da se promenljiva tipa `LayoutManager` može koristiti kao referenca na bilo koji tip layout manager-a.



# MENADŽERI POSTAVKI

- Menadžeri postavki u awt paketu:
  - `FlowLayout` – dodaju se komponente u sukcesivnim redovima – kad je red popunjen, počinje se sa novim; podrazumevan za `JPanel`.
  - `BorderLayout` – prozor je podeljen na 5 delova – north, south, east, west, center; podrazumevani za `ContentPane` u `JFrame`-u, `JDialog`, `JApplet`.
  - `CardLayout` – komponente se raspoređuju jedna iznad druge, kao špil karata. Jedino je komponenta na “vrhu” vidljiva u bilo kom trenutku.
  - `GridLayout` – komponente se raspoređuju u pravougaonu mrežu, pri čemu korisnik zadaje broj redova i kolona.
  - `GridBagLayout` – pravougaona mreža, pri čemu širina redova i kolona može da varira.
- Menadžeri postavki u swing paketu:
  - `BoxLayout` – raspoređuje komponente u red ili kolonu, pri čemu se iste skraćuju pre nego što pređu u drugi red ili kolonu. `BoxLayout` manager je podrazumevan za klasu `Box` kontejnera.
  - `SpringLayout` – omogućava pozicioniranje komponente uz ivicu kontejnera ili neke druge komponente

# CONTENTPANE | JPANEL

- Content Pane – kontejner
  - Centralni deo prozora
  - Na njega se dodaju ostale komponente (dugmići, polja za unos...)
- Koncept rasporedjivanja komponenti u contentPane-u:
  - u zavisnosti od potrebe komponente se grupišu tako što se dodaju nekom panelu, koji ima odgovarajući raspored, a onda se paneli smeštaju u veći kontejner
- JPanel klasa
  - Predstavlja površinu na koju se mogu dodati druge komponente
  - Najčešće korišćena klasa za 'grupisanje' komponenti u određenom rasporedu.

# CONTENTPANE I JPANEL

**cp - contentPane**

sa BorderLayout  
postavkom

**prvi - JPanel**

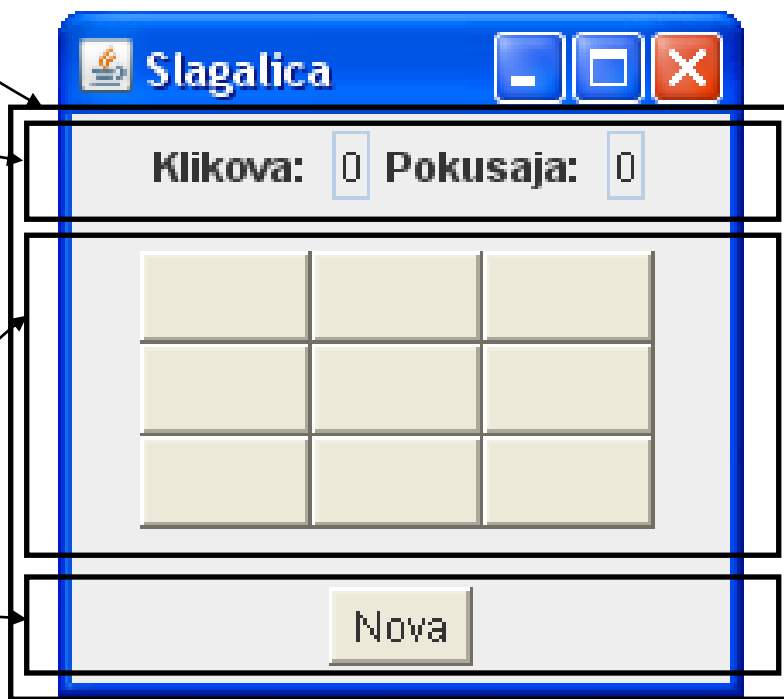
FlowLayout postavka  
sadrži dve labele i dva textfield-a

**drugi - JPanel**

GridLayout postavka  
sadrži 9 dugmića

**treći - JPanel**

FlowLayout postavka  
sadrži jedno dugme



```
cp.add(prvi, BorderLayout.NORTH)  
cp.add(drugi, BorderLayout.CENTER)  
cp.add(treci, BorderLayout.SOUTH)
```