

---

# GUI - Layout Manager-i

Bojan Tomić

# Layout Manager-i

---

- Content Pane
  - Centralni deo prozora
  - Na njega se dodaju ostale komponente (dugmići, polja za unos...)
  - To je objekat klase “`javax.swing.JPanel`”
- JPanel klasa
  - Predstavlja površinu na koju se mogu dodati druge komponente
  - Može da organizuje komponente koje se nalaze na njoj u specifičnom rasporedu

# Layout Manager-i

---

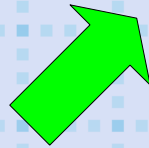
- Klase koje definišu raspored komponenti na JPanel površini
  - Null layout (negde se zove i absolute layout)
  - FlowLayout
  - GridLayout
  - BorderLayout
  - itd. (BoxLayout, CardLayout, GridBagLayout)
- Ove klase rešavaju i pitanje šta se dešava sa rasporedom komponenti kada se dimenzije forme izmene

# Null layout

---

- Panel kao layout zapravo dobija null vrednost
- Null layout
  - Omogućava da se komponente rasporede na bilo koji način tj. ne utiče na njihov raspored
  - Kada se dimenzije forme izmene, komponente ostaju na svom mestu, a povećava se ili smanjuje prostor oko njih.
  - Koristi se ako se prave forme sa veoma neobičnim rasporedom i ako se zna da se dimenzije forme neće menjati

# Null layout



# FlowLayout

---

- Klasa koja raspoređuje komponente na JPanel-u u jedan red
- Kada komponente više ne mogu da stanu u jedan red, popunjava se sledeći red, itd.
- Veličina konkretne komponente se može promeniti samo korišćenjem atributa “preferredSize” te komponente
- Kada se dimenzija prozora promeni, komponente se preslože po istom principu (ali neke zbog toga promene red)

# FlowLayout

---

- Atributi klase “java.awt.FlowLayout” omogućavaju dodatno prilagođavanje rasporeda
- alignment
  - LEFT - sve komponente se slažu tako da budu što bliže levoj ivici JPanel-a
  - CENTER - sve komponente se centriraju prema sredini JPanel-a
  - RIGHT - sve komponente se slažu tako da budu što bliže desnoj ivici JPanel-a

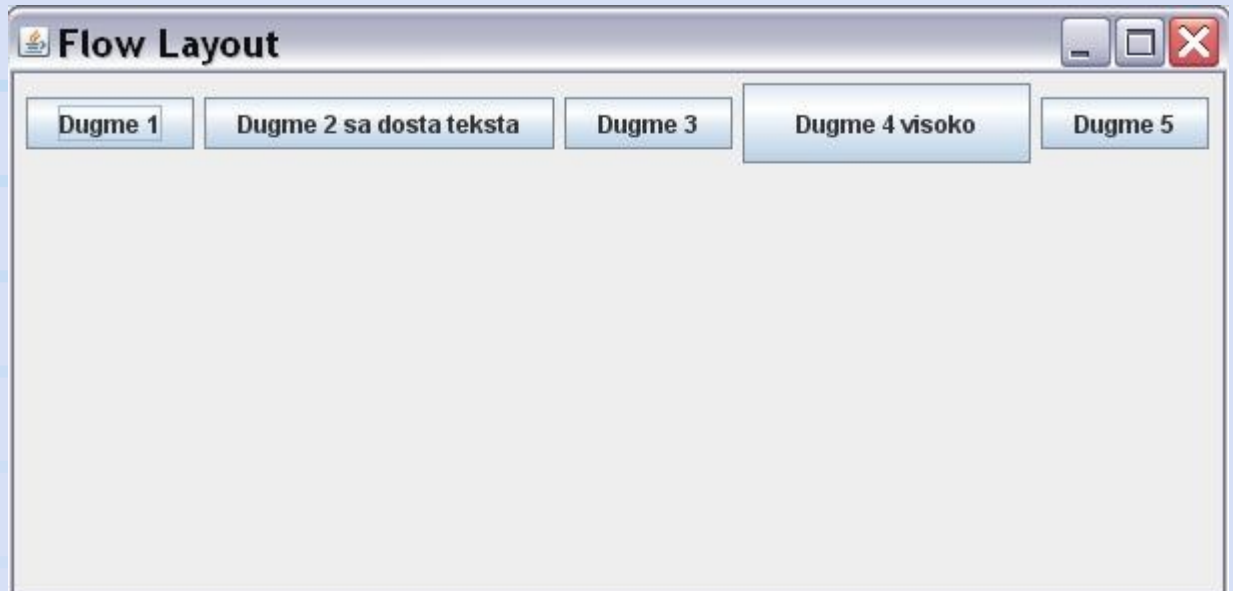
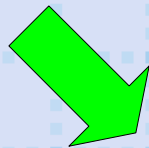
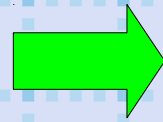
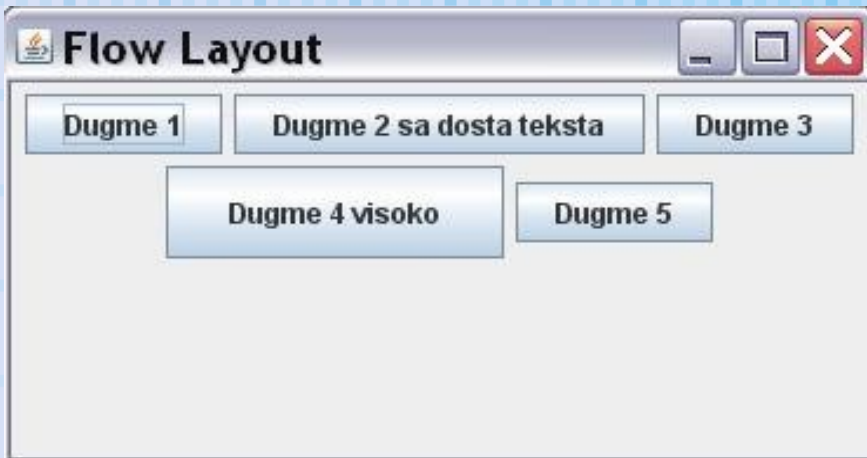
# FlowLayout

---

- `horizontalGap`
  - Horizontalni razmak između dve komponente u redu (u pikselima)
- `verticalGap`
  - Vertikalni razmak između dva reda komponenti (u pikselima)



# FlowLayout



# GridLayout

---

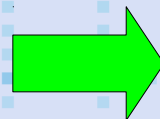
- Klasa koja raspoređuje komponente na JPanel-u u formi matrice
- Svaka komponenta staje u po jedno polje matrice, a sva polja matrice su iste veličine
- Veličina konkretne komponente se ne može promeniti - sve komponente moraju da budu iste veličine
- Kada se dimenzija prozora promeni, komponente se preslože po istom principu (veličina polja matrice se poveća ili smanji)

# GridLayout

---

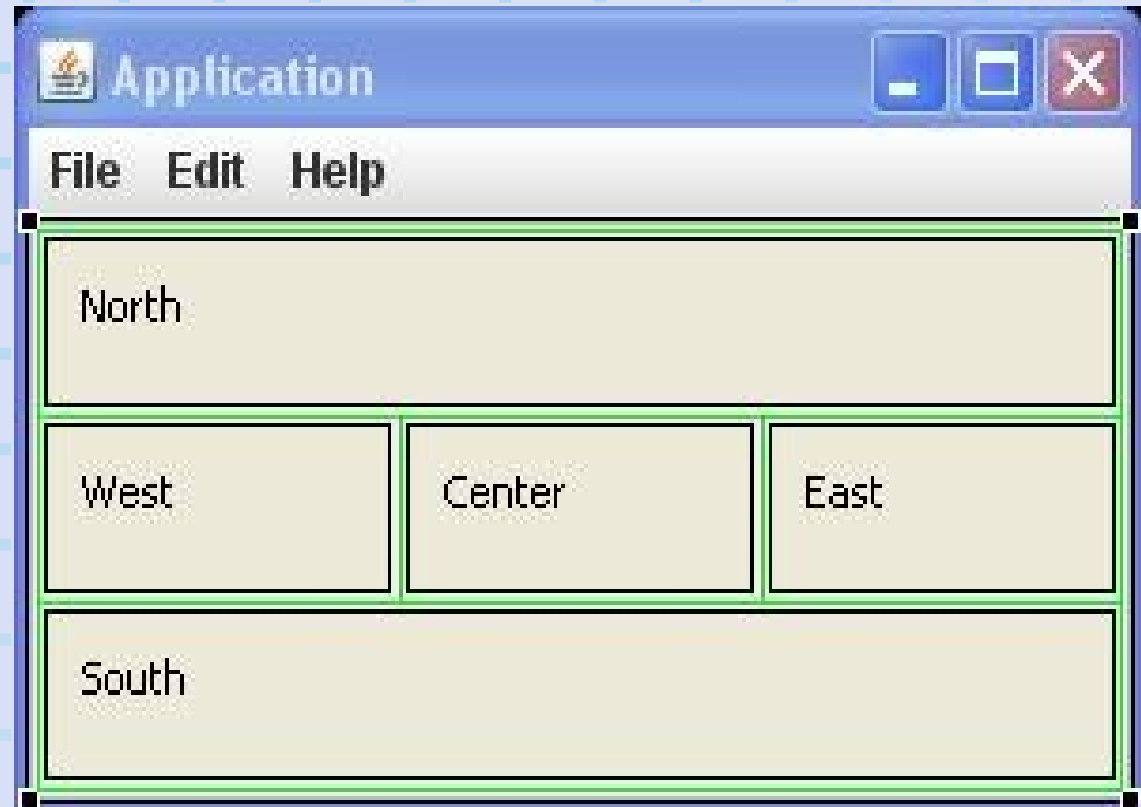
- Atributi klase “java.awt.GridLayout”
- columns
  - broj kolona matrice (ako se ostavi 0, napravi se potreban broj kolona kako se dodaju komponente)
- rows
  - broj redova matrice (ako se ostavi 0, napravi se potreban broj redova kako se dodaju komponente)
  - Broj kolona i redova ne smeju istovremeno biti 0
- horizontalGap
- verticalGap

# GridLayout



# BorderLayout

- Deli JPanel na pet delova
  - NORTH
  - SOUTH
  - EAST
  - WEST
  - CENTER
- Komponente se mogu dodavati u bilo koji deo

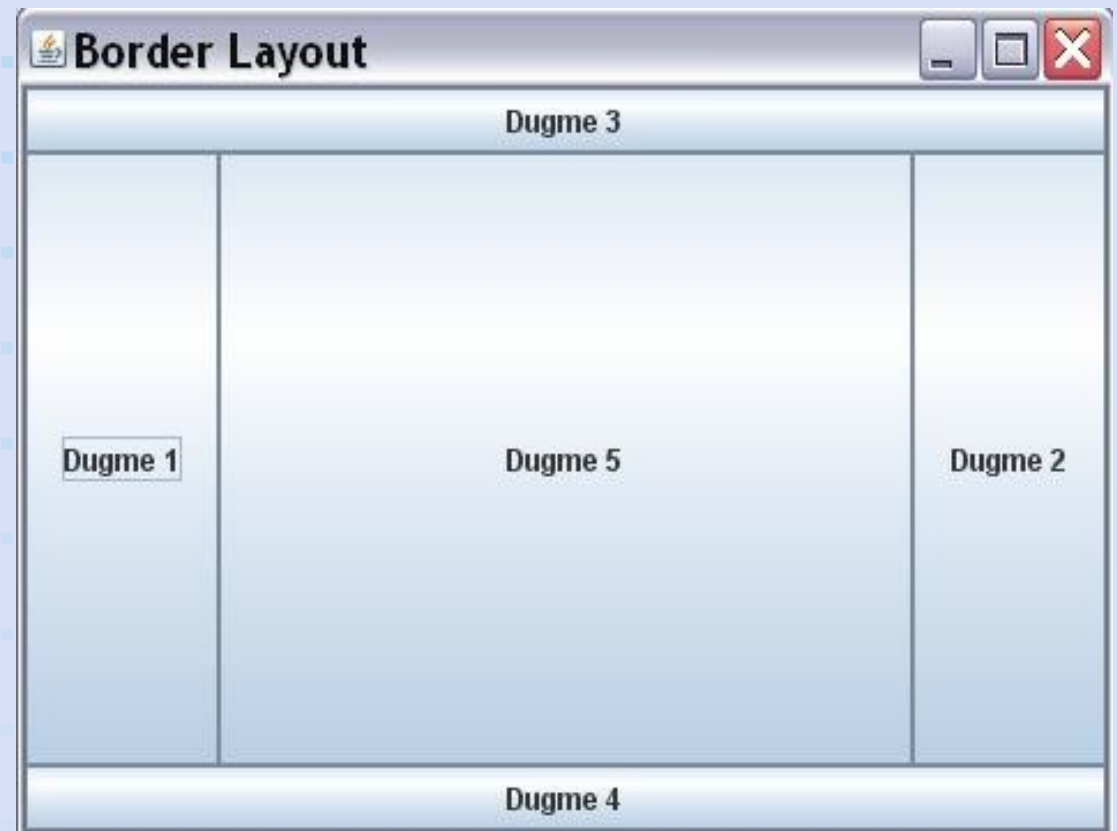
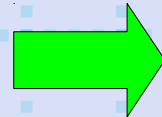


# BorderLayout

---

- Kada se forma poveća, povećava se samo centralni deo a ostali delovi ostaju iste veličine
- Veličina konkretne komponente se može promeniti samo korišćenjem atributa “preferredSize” te komponente i to samo ako komponenta nije u centralnom delu
- Atributi
  - horizontalGap
  - verticalGap

# BorderLayout



# BoxLayout (X axis, Y axis)

---

- Klasa koja raspoređuje komponente na JPanel-u u jedan red: horizontalno (X axis) ili vertikalno (Y axis)
- Veoma sličan FlowLayout-u ali dosta primitivniji
- Za razliku od FlowLayout, kada komponente više ne mogu da stanu u jedan red NE popunjava se sledeći red već se jednostavno ne vide.
- Veličina komponente se menja korišćenjem atributa “preferredSize”

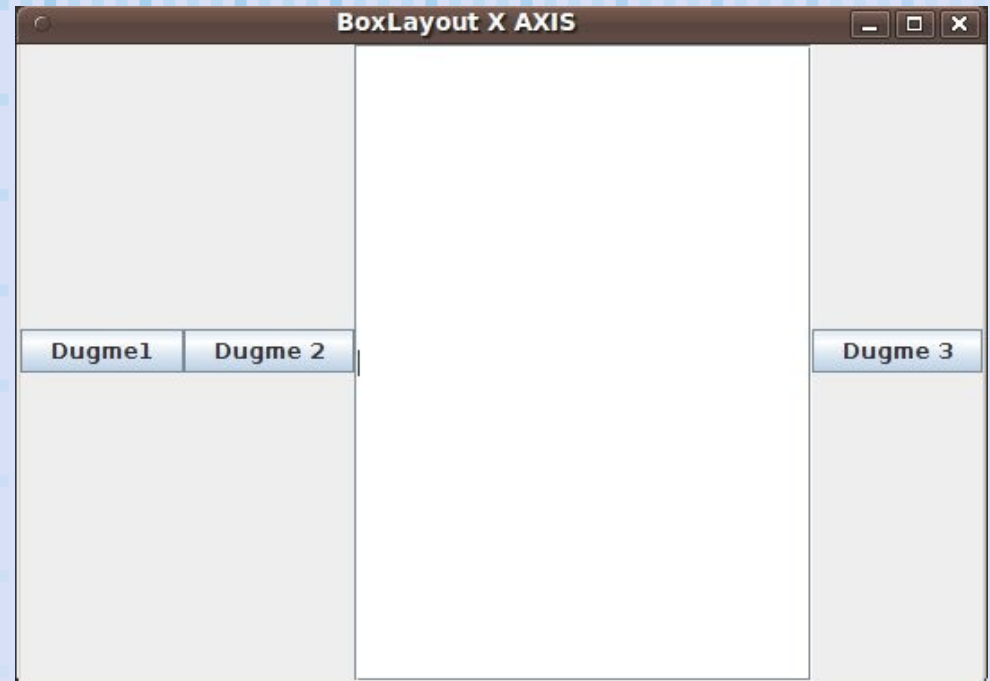
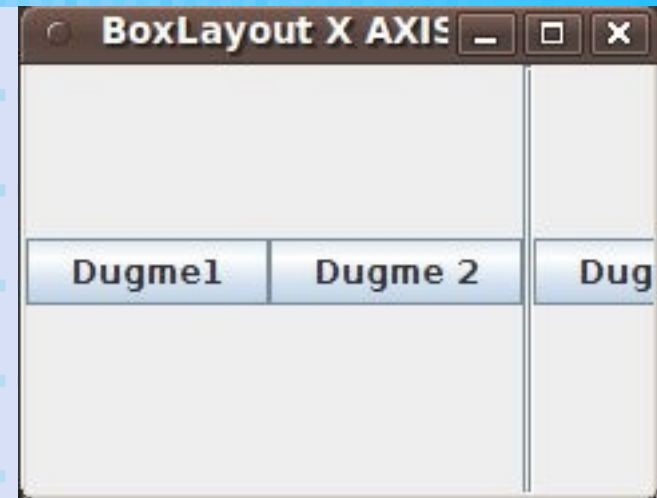
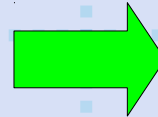
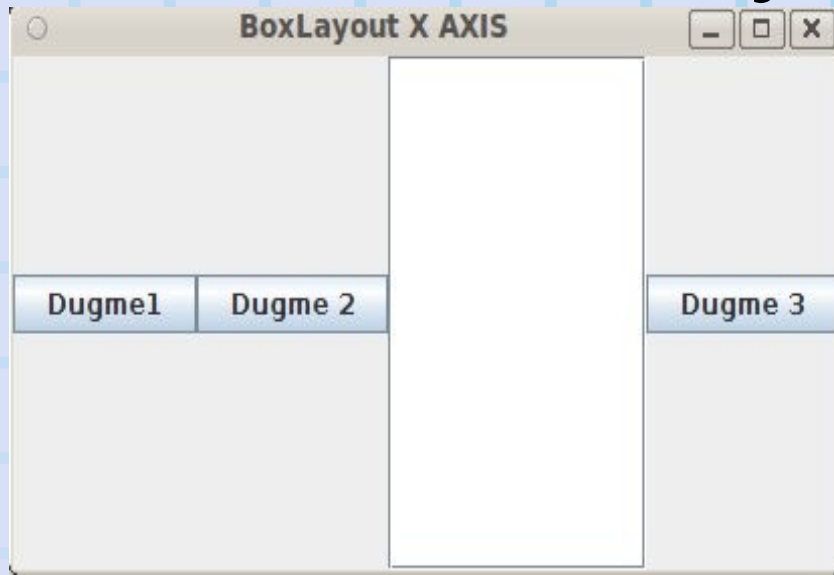


# BoxLayout (X axis, Y axis)

---

- Kada se dimenzija prozora smanji, komponente se “odseku”
- Kada se se dimenzija prozora poveća, neke komponente se povećaju (tipa JTextField i JTextArea) a ostale ostanu iste veličine.
- NEMA dodatno podešavanje rasporeda.

# BoxLayout (X axis, Y axis)



# CardLayout

---

- Klasa koja raspoređuje komponente na JPanel-u kao “karte” koje su složene jedna preko druge
- Svaka komponenta predstavlja jednu “kartu”
- U svakom trenutku se vidi samo jedna “karta” a ostale su sakrivene
- Veoma je slično efektu JTabbedPane komponente, ali karte nemaju jezičke (korisnik ne može sam da ih menja)
- Efekat: prikaz različitih komponenti na istom prostoru u zavisnosti od potrebe

# CardLayout

---

- Najbolji način da se koristi CardLayout je da se JPanel-i koriste kao karte i da se na svaki stave komponente
- Tada svaki JPanel može da ima svoj layout i može se postići da prozor izgleda značajno drugačije u odnosu na to koja “karta” se vidi
- Metoda za dodavanje “karte”:

```
addLayoutComponent (komponenta, nazivKarte)
```

- Metoda za prikaz jedne “karte”:

```
show ( panelSaCardLayout, nazivKarte)
```

# Kombinovanje Layout Manager-a

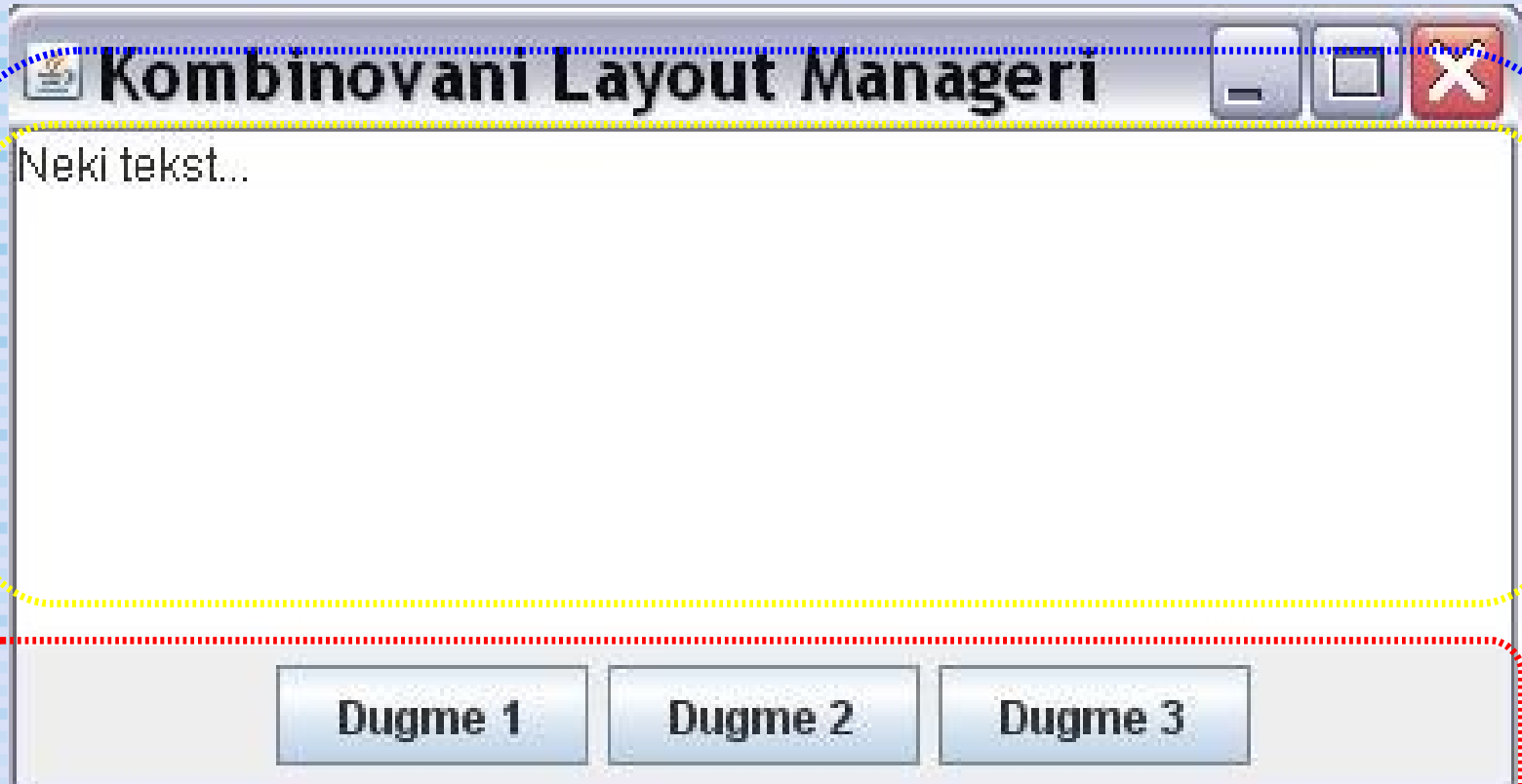
---

- Kombinovanjem različitih Layout Manager-a može da se dobije izuzetno složen GUI
- Svaki JPanel ima svog Layout Manager-a
- Postavljanje više JPanel-a jedan preko drugog
- Na primer
  - ContentPane - BorderLayout
  - CENTER - JTextArea
  - SOUTH - JPanel sa FlowLayout i tri dugmeta

# Kombinovanje Layout Manager-a

ContentPane  
(BorderLayout)

JTextArea  
(CENTER  
deo  
ContentPa  
ne-a)



JPanel (FlowLayout) sa tri dugmeta pozicioniran  
na SOUTH delu ContentPane-a

# Kombinovanje Layout Manager-a

