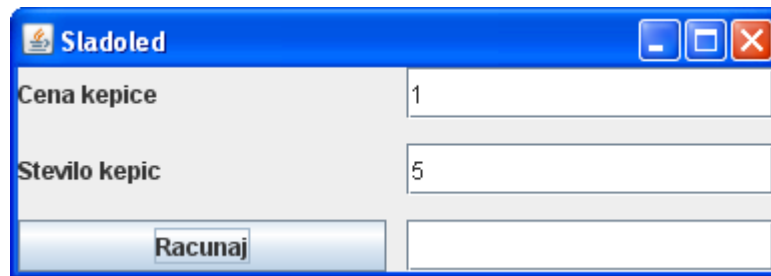


Uporabniški grafični vmesnik (GUI)

Programiranje 2, Tomaž Dobravec



Okna v Javi



Kaj bi se radi naučili

- ▶ okno (JFrame)
- ▶ komponente (JTextField, JLabel, JButton, ...)
- ▶ razporejanje
- ▶ akcije (pritisk na gumb, ...)



Okna v Javi

- ▶ Za delo z okni Java ponuja dve knjižnici:

`java.awt` (od verzije Java 1.0)

`javax.swing` (od verzije Java 1.2)

- ▶ AWT = Abstract Windowing Toolkit

- ▶ Swing (del Java Foundation Classes)

- ▶ JavaFX



Swing in AWT

- ▶ Razlike:
 - ▶ Swing ponuja več komponent
 - ▶ Swing ponuja lepše komponente
 - ▶ Swing komponente imajo lahko različne izgleda

AWT

Frame

Button

...

Swing

JFrame

JButton

J...



Swing in AWT

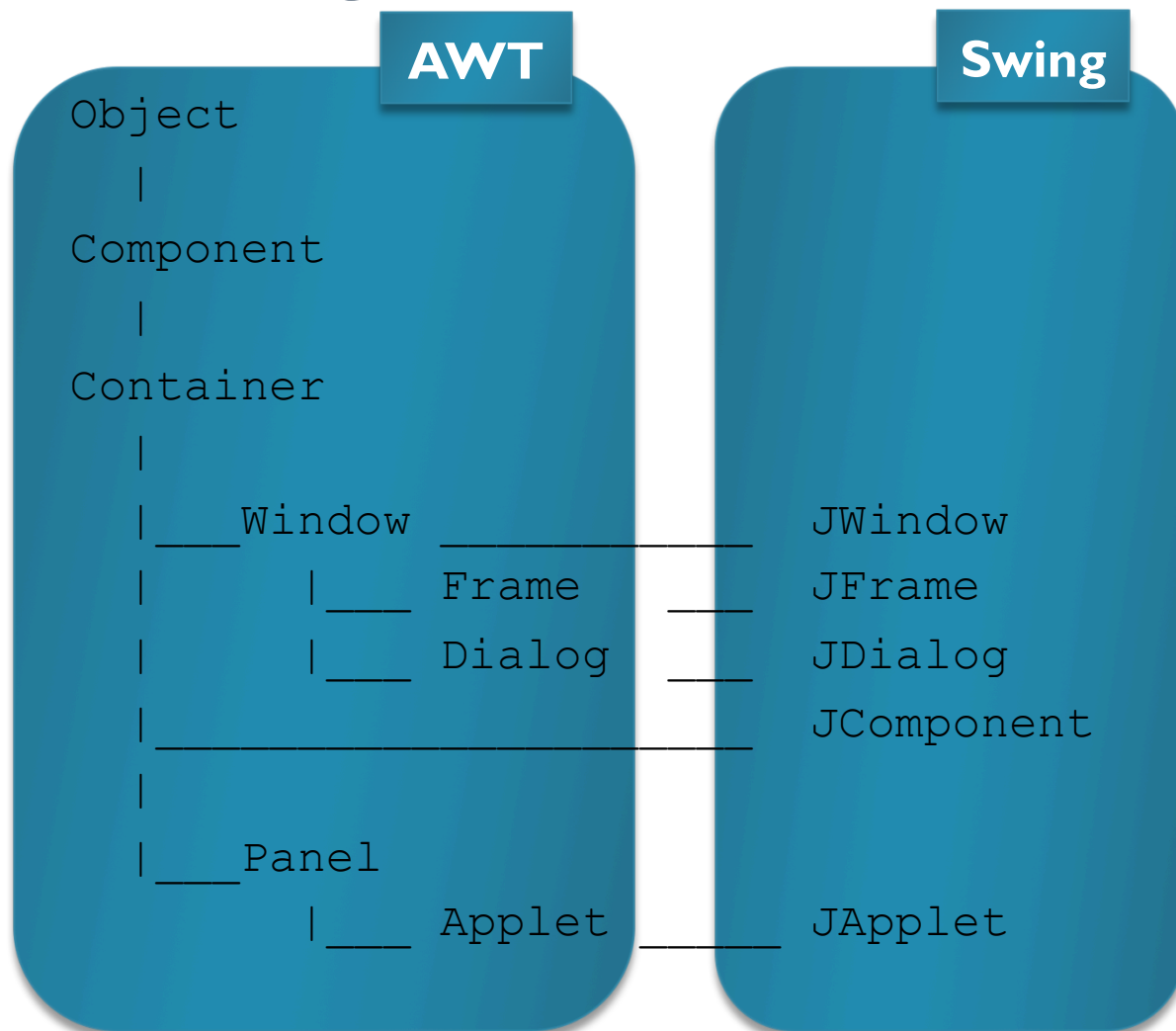
- ▶ Najpomembnejša razlika med AWT in Swing:

**AWT risanje vseh kontrol prepusti sistemu,
Swing večino kontrol izriše sam**

- ▶ AWT je močno odvisen od sistema, Swing le delno



Swing in AWT





Swing in AWT

- ▶ JWindow, JFrame, JDialog so izpeljani iz Window, Frame, Dialog, zato **niso** neodvisni od sistema.
- ▶ Vse ostale kontrole knjižnice Swing pa **so** neodvisne od sistema (za njihovo risanje poskrbi Java)!



Swing in AWT

- ▶ AWT še vedno obstaja, ker je Swing odvisen od njega.
- ▶ Priporočilo: Pri pisanju grafičnih programov uporabi Swing.
- ▶ V programu nikoli ne uporablaj hkrati AWT in Swing komponent.
- ▶ AWT uporabi le, če Swing željene funkcionalnosti nima.

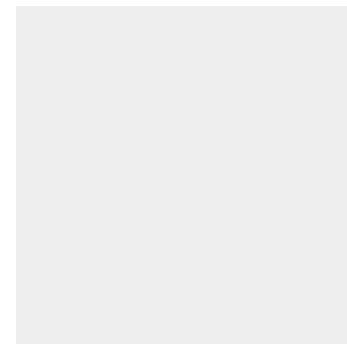


Osnovni vsebniki

- ▶ JFrame, JDialog in JWindow **so osnovni vsebniki (top-level containers).**

JWindow

- ▶ Okno brez glave in obrobe
- ▶ Uporablja se redko

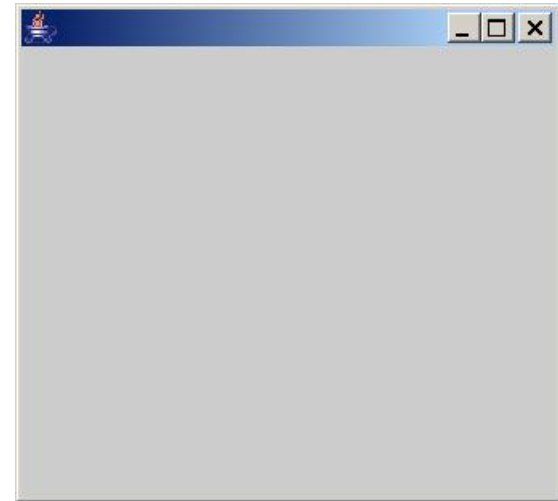




Razred JFrame

JFrame

- ▶ JFrame je glavno okno programa.
- ▶ Vsak Swing program naj bi imel vsaj eno JFrame okno

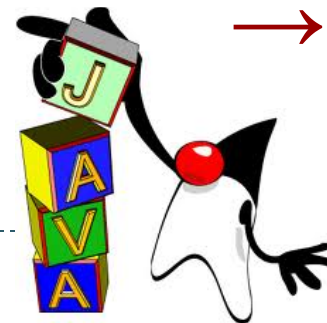


```
import javax.swing.*;  
  
JFrame okno = new JFrame("Test");  
okno.setVisible(true);
```

Naloga

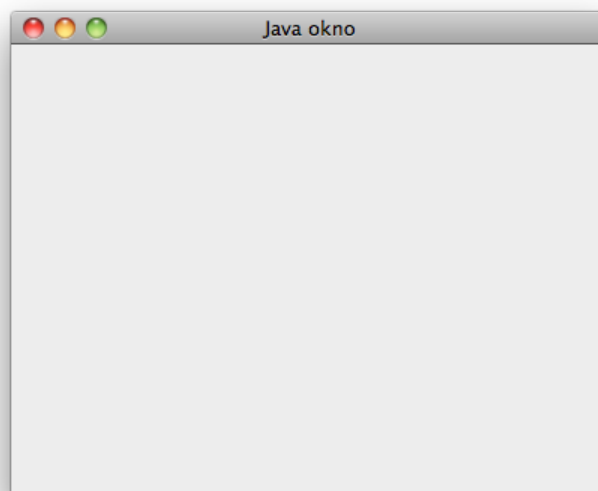
Izris JFrame okna

okna/JavaOkno.java



Napiši program, ki izriše okno, kot ga prikazuje slika, na dva načina:

- a) kodo napiši sam,
- b) kodo naj napiše Netbeans

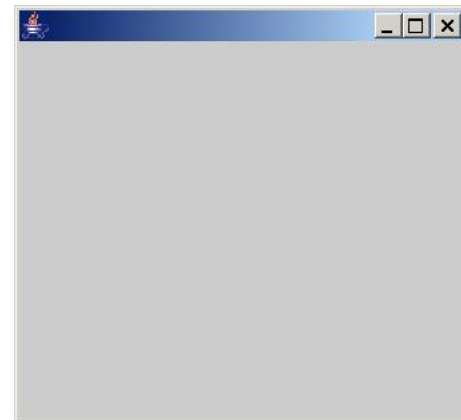


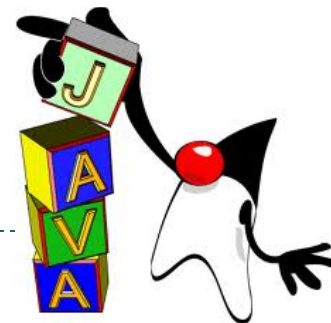


Razred JDialog

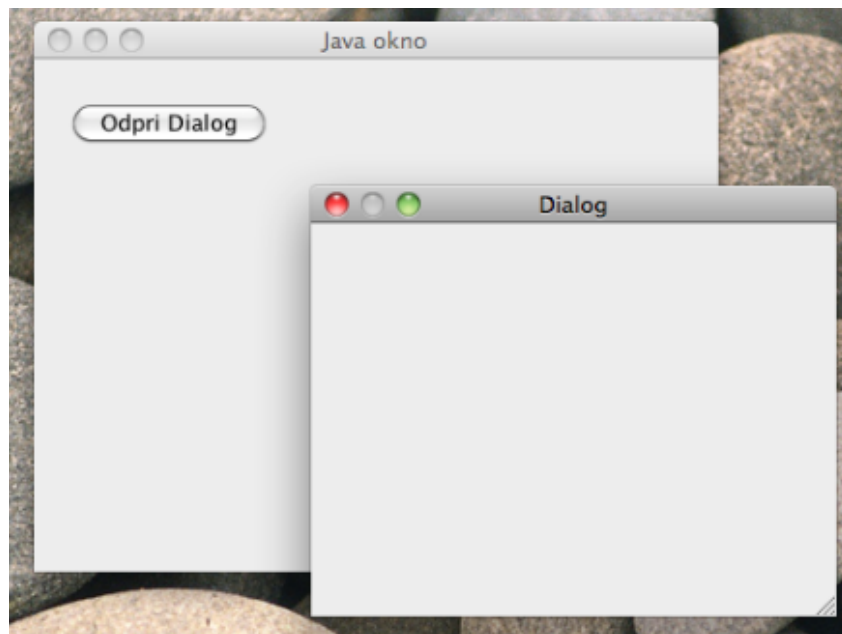
JDialog

- ▶ JDialog je okno, ki se odpre med izvajanjem programa
- ▶ Objekt tipa JDialog je odvisen od nekega JFrame objekta
- ▶ Objekt tipa JDialog se lahko prikaže tudi **modalno** (dokler se dialog ne zapre, glavnega programa ne moremo uporabljati).





S pomočjo Netbeansa napiši program, ki izriše okno z gumbom “Odpri dialog”. Ob kliku na gumb naj se odpre modalni dialog.





Metoda razreda JFrame

Nekatere metode razreda JFrame

```
public void setTitle(String title)
public void setSize(int width, int height)
public void setLocation(int x, int y)

public void setResizable(boolean resizable)
public boolean isResizable()
```





Swing komponente

- ▶ Primer komponent: napis, vpisno polje, gumb, ...

- ▶ Komponente prikažemo v oknu (npr. v JFrame ali v JApplet oknu).
- ▶ Swing komponente so naslednice razreda JComponent



Swing komponente

JComponent

| ____ JLabel

| ____ JList

| ____ JPanel

| ____ JMenuBar

| ____ JPopupMenu

| ____ JScrollPane

| ____ JAbstractButton

| | ____ JToggleButton

| | | ____ JCheckBox

| | | ____ JRadioButton

| | ____ JButton

| | ____ JMenuItem ____ JMenu

| ____ JTextComponent

 | ____ JTextArea

 | ____ JTextField ____ JPasswordField



Uporabniški grafični vmesnik (GUI)



Swing komponente

Za grafičen prikaz Swing komponent glej

A Visual Guide to Swing Components

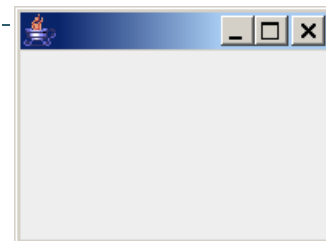
<http://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>



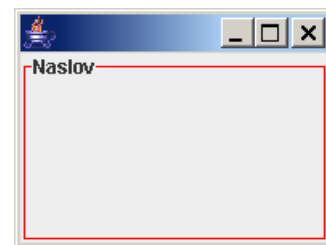


Komponenta JPanel

- ▶ Zelo uporabna komponenta
- ▶ JPanel je običajno neviden
- ▶ Na JPanel odlagamo druge komponente



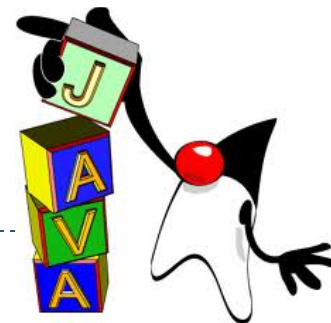
- ▶ JPanel ima lahko okvir (border)
- ▶ okvir ima lahko naslov (title)
 - ▶ uporabim `BorderFactory`



Naloga

Panel na oknu

okna/TestPanel.java



Glavnemu oknu dodaj panel, panelu dodaj okvir in spremeni barvo ozadja.





Look and Feel

- ▶ Swing komponentam lahko spremenimo videz
- ▶ Trije osnovni videzi: Metal, Motif, Windows

- ▶ **Primer uporabe:**

```
UIManager.getInstalledLookAndFeels();  
UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.Wi  
ndowsLookAndFeel");
```

```
// com.sun.java.swing.plaf.gtk.GTKLookAndFeel  
// com.sun.java.swing.plaf.motif.MotifLookAndFeel
```

```
SwingUtilities.updateComponentTreeUI(this);
```

Glej: [SwingSet2 demo](#)



Nekatere metode razreda JComponent

```
public void setSize(int width, int height)
```

```
public void setLocation(Point p)
```

```
public void setBounds(int x, int y,  
                      int width, int height)
```

```
public void setEnabled(boolean enabled)
```

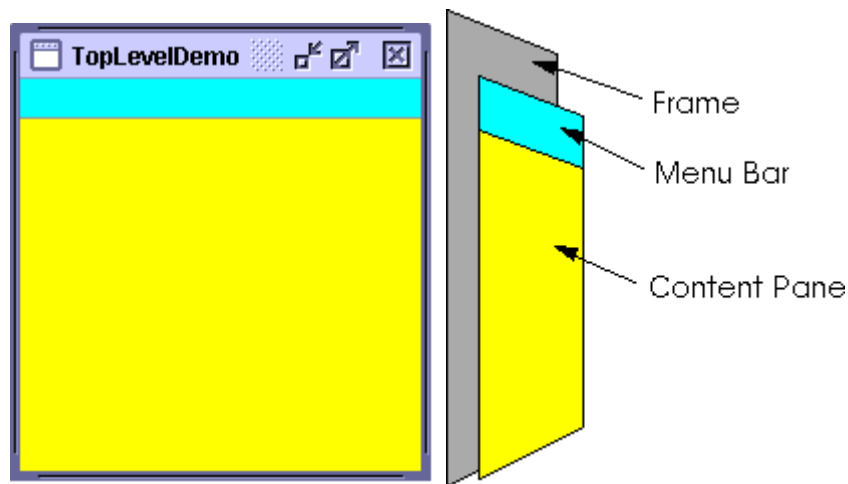
```
public void setForeground(Color fg)
```

```
public void setFont(Font font)
```





Osnovni vsebnik



- vsebnik (*angl. container*), na katerega odlagamo ostale komponente se imenuje “Content Pane”
- dobimo ga z ukazom

```
mojeOkno.getContentPane();
```





Razporejanje komponent

- ▶ Komponente razporejamo na dva načina:
 - ▶ absolutno pozicioniranje,
 - ▶ pozicioniranje s pomočjo razporejevalnika.



Absolutno pozicioniranje

- ▶ Javi povemo, da želimo komponente razporejati sami:

```
Container vsebnik = mojeOkno.getContentPane();  
vsebnik.setLayout(null);
```

- ▶ vsaki komponenti posebej določim položaj in velikost

```
mojGumb.setBounds(10,20,50,30); // položaj: x=10, y=20  
// velikost: širina=50, višina=30
```

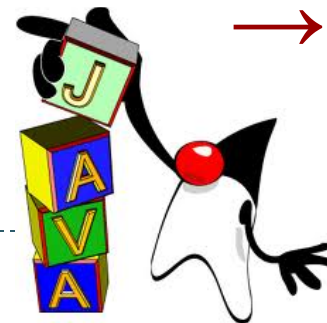
- ▶ komponento postavimo na okno

```
vsebnik.add(mojGumb);
```

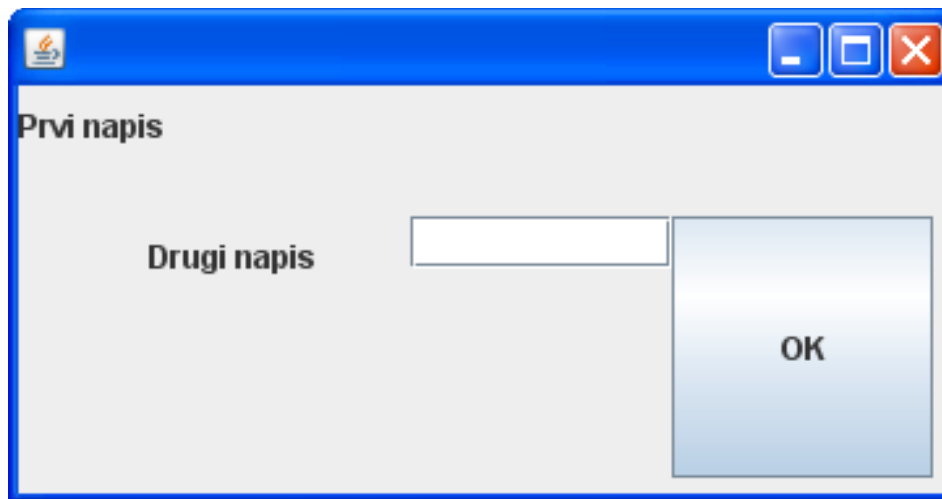

Naloga

Absolutno pozicioniranje

okna/AbsPos.java



Napiši program, ki z absolutnim pozicioniranjem komponent izriše okno, kot prikazuje spodnja slika



Uporabniški grafični vmesnik (GUI)



Razporejevalniki

- ▶ Razporejevalnik pomaga (sodeluje) pri razporejanju komponent
 - ▶ Komponente razporeja glede na
 - ▶ način razporejanja in
 - ▶ vrstni red dodajanja komponent.
 - ▶ Velikost in lego komponent določi razporejevalnik.
 - ▶ Uporabnik lahko določi le priporočeno velikost (*angl. preferred size*) posamezne komponente.
-
- ▶ Uporabniški grafični vmesnik (GUI)



Razporejevalniki

- ▶ `FlowLayout` ... razporejanje v vrsto
- ▶ `BorderLayout` ... razporejanje v pet con
(gor, levo, center, desno, dol)
- ▶ `GridLayout` ... razporejanje v mrežo,
(ena komponenta, eno polje)
- ▶ `GridBagLayout` ... razporejanje v mrežo
(komponenta lahko v več poljih)
- ▶ `CardLayout` ... razporejanje na eno mesto



Razporejevalnik `FlowLayout`

- ▶ Komponente se dodajajo ena za drugo v vrsto.



- ▶ Če je vrsta prekratka za naslednjo komponento, se ta doda v novo vrsto.
- ▶ Če je vrsta predolga, se komponente poravnajo na sredino.



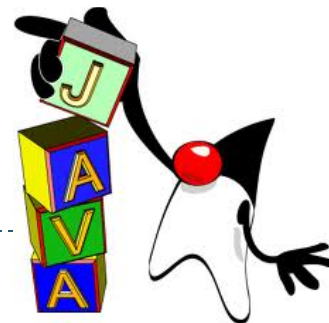
Razporejevalnik `FlowLayout`

- ▶ V konstruktorju za `FlowLayout` lahko določimo poravnavo (privzeto: center) in prostor okoli komponent
- ▶ `FlowLayout` je privzet razporejevalnik za `JPanel`

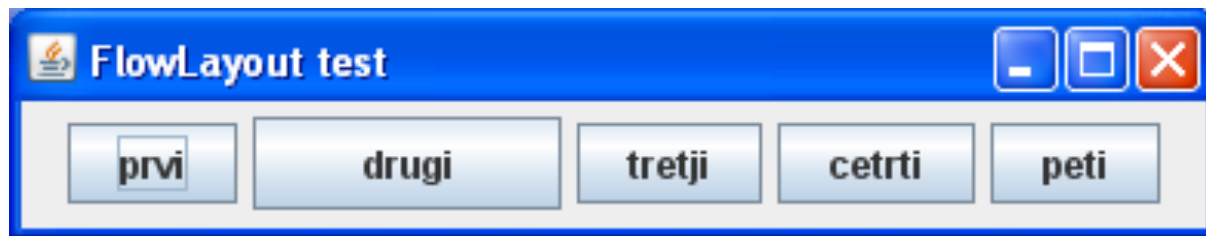
Naloga

Primer FlowLayout

okna/FlowLayout.java



Napiši program, ki na okno postavi pet gumbov, kot prikazuje spodnja slika

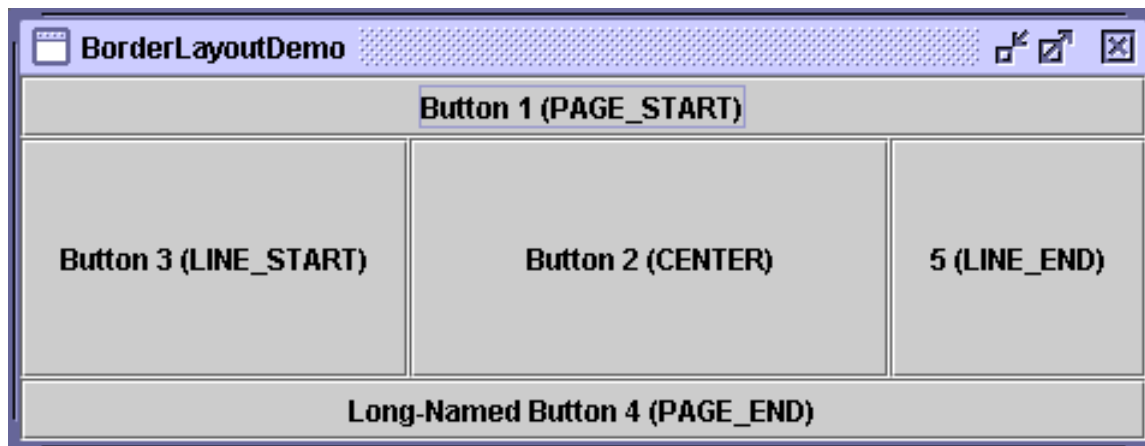


Uporabniški grafični vmesnik (GUI)



Razporejevalnik BorderLayout

- ▶ BorderLayout predivdeva pet con:
PAGE_START, PAGE_END, LINE_START, LINE_END in CENTER



- ▶ dodajanje komponente:

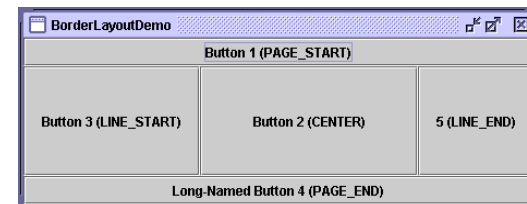
```
add(gumb, BorderLayout.PAGE_END);
```

- ▶ Uporabniški grafični vmesnik (GUI)



Razporejevalnik BorderLayout

- ▶ komponente lahko dodamo tudi na manj kot 5 con (na primer: samo v center in bottom)



- ▶ CENTER je glavni:
 - srednja cona zavzame ves prazen prostor
 - komponenta, za katero eksplicitno ne navedemo mesta:

`add (gumb) ;`

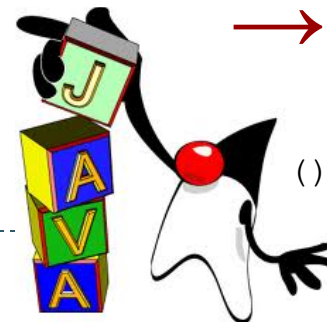
se doda v center

- ▶ BorderLayout je privzet razporejevalnik za JFrame, JDialog in JApplet

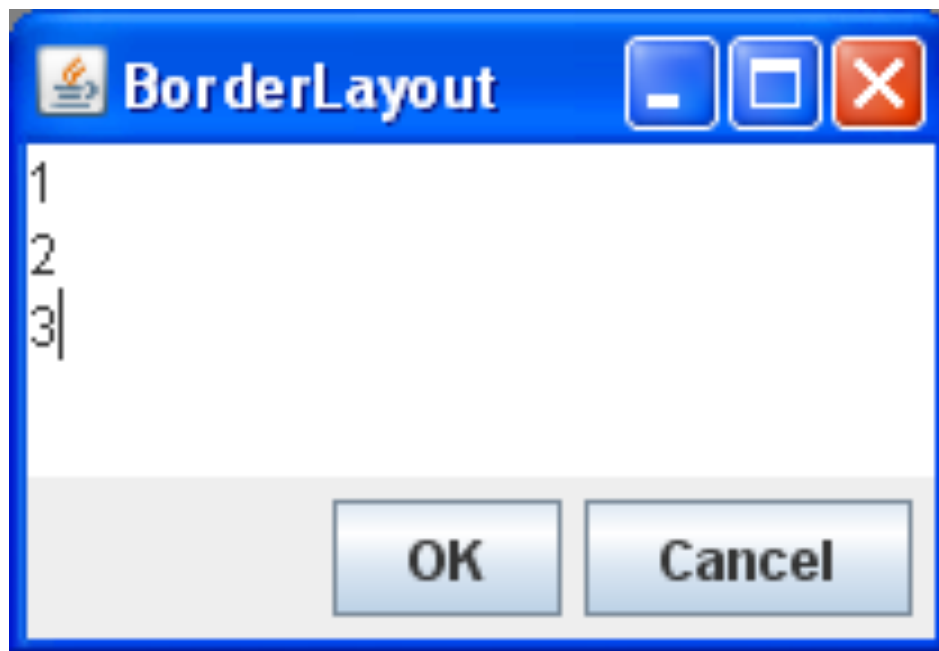
Naloga

Primer BorderLayout

okna/BorderLayout.java



Napiši program, ki nariše okno kot prikazuje spodnja slika (namig: za gumbe uporabi dodatni JPanel).



Uporabniški grafični vmesnik (GUI)



Razporejevalnik GridLayout

- ▶ mreža enakih celic
m stolpcev in n vrstic



- ▶ komponente se razporejajo v celice
vsaka komponenta v eno celico
- ▶ komponenta zasede ves prostor v celici
enako velike komponente



Razporejevalnik GridLayout

- ▶ Ukaz `add(komponenta)` doda komponento v naslednjo prosto celico.
- ▶ Polnjenje celic poteka od leve proti desni, od zgoraj navzdol.
- ▶ Preskakovanje celic ni možno!
- ▶ Neposreden vpis v celico ni možen.



Razporejevalnik `CardLayout`

- ▶ Vsebnik je razdeljen na več strani (kart); naenkrat je vidna le ena (zgornja) stran.

- ▶ Gradniki se dodajajo na posamezne strani.

- ▶ Posamezne karte prikazujemo z metodami

`next/previous/last/first/show`.



Razporejevalnik `GridBagLayout`

- ▶ Razporejanje v mrežo (podobno kot `GridLayout`)
- ▶ Ena komponenta lahko zasede več celic mreže.
- ▶ Natančno lahko določimo, v kateri celici bo posamezna komponenta.
- ▶ Vrstni red dodajanja ni pomemben.



Razporejevalnik GridBagLayout

- ▶ Pri dodajanju komponent podamo tudi omejitve (*angl. constraints*).
- ▶ Omejitve so objekt razreda GridBagConstraints

```
vsebnik.setLayout(new GridBagLayout());  
GridBagConstraints omejitve = new GridBagConstraints();
```

```
// nastavim omejitve  
omejitve.gridx=0;  
...
```

```
vsebnik.add(komponenta, omejitve);
```



Razporejevalnik GridBagLayout

► Pomen nekaterih atributov v GridBagConstraints

gridx

položaj (x in y koordinata celice)

gridy

gridwidth

**število celic (po širini in višini),
ki jih komponenta zaseda**

gridheight

fill

raztezanje komponent (H/V/B/N)

insets

prazen prostor okoli komponente

anchor

poravnava v celici (N/W/E/S/NW,...)

weightx

**utež, ki določa, kako naj se razdeli prostor med
posamezne celice mreže**

weighty



Razporejevalnik GridBagLayout

- ▶ **Fill – raztezanje (HORIZONTAL, VERTICAL, BOTH, NONE)**

```
omejitve.fill = GridBagConstraints.BOTH;
```

- ▶ **Insets - prazen prostor (zgoraj, levo, spodaj, desno)**

```
omejitve.insets = new Insets(10, 0, 0, 0);
```

- ▶ **Anchor (North, West, South, East, NorthWest, ...)**

```
omejitve.anchor = GridBagConstraints.NORTH;
```

-
- ▶ **Uporabniški grafični vmesnik (GUI)**



Razporejevalnik GridBagLayout

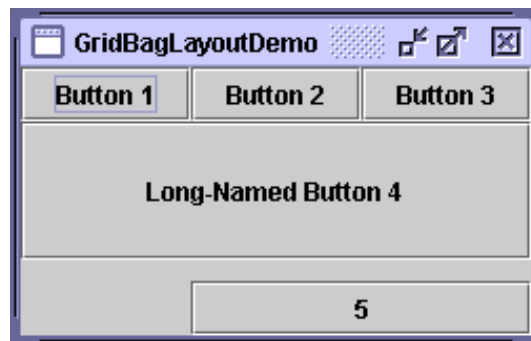
Button1

`gridx=0`

`gridy=0`

`gridwidth=1`

`gridheight=1`



Button5

Button4

`gridx=0`

`gridy=1`

`gridwidth=3`

`gridheight=1`

`fill=HORIZONTAL`

`gridx=1`

`gridy=2`

`gridwidth=2`

`gridheight=1`

`fill=HORIZONTAL`

`insets=(10,0,0,0)`

`weighty=1`

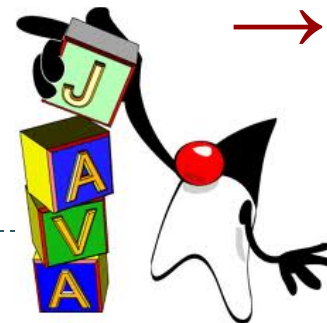
`anchor=SOUTH`



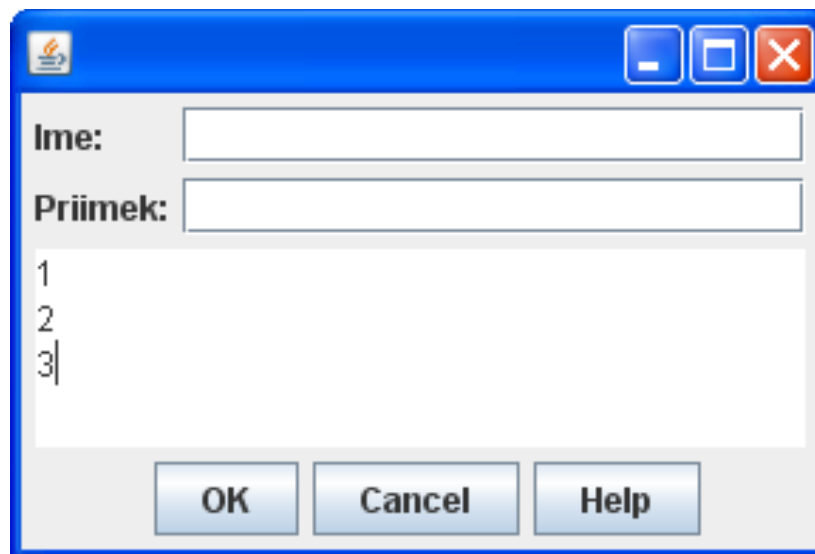
Naloga

Primer GridBagLayout

okna/GridBagLayout.java



Napiši program, ki nariše okno, kot prikazuje spodnja slika.



Uporabniški grafični vmesnik (GUI)



Razred `java.awt.Toolkit`

- ▶ Objekt razreda `Toolkit` dobimo s klicem statične metode

```
Toolkit tk = Toolkit.getDefaultToolkit();
```

- ▶ Nekatere metode:

```
public abstract void beep()
```

```
public boolean getLockingKeyState(int keyCode)
```

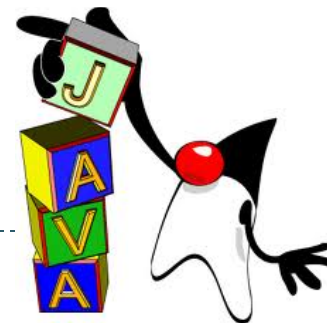
```
public void setLockingKeyState(int keyCode, boolean on)
```

```
public abstract Dimension getScreenSize()
```

Naloga

Okno čez ves zaslon

`okna/GridBagLayout.java`



Popravi program `okna/GridBagLayout.java` tako, da se bo okno ob zagonu programa raztezalo čez ves zaslon.



Uporabniški grafični vmesnik (GUI)



Zapiranje okna

- ▶ Okno (JFrame) se ob pritisku na križec samo skrije
- ▶ Program se ob tem ne konča (isto okno lahko kasneje s `setVisible()` ponovno prikažemo)
- ▶ Rešitev: uporabimo metodo `setDefaultCloseOperation()`

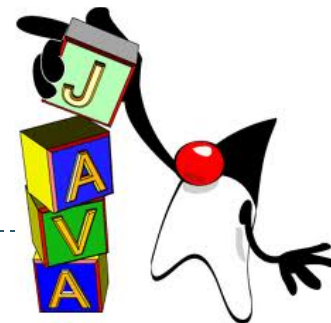
```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



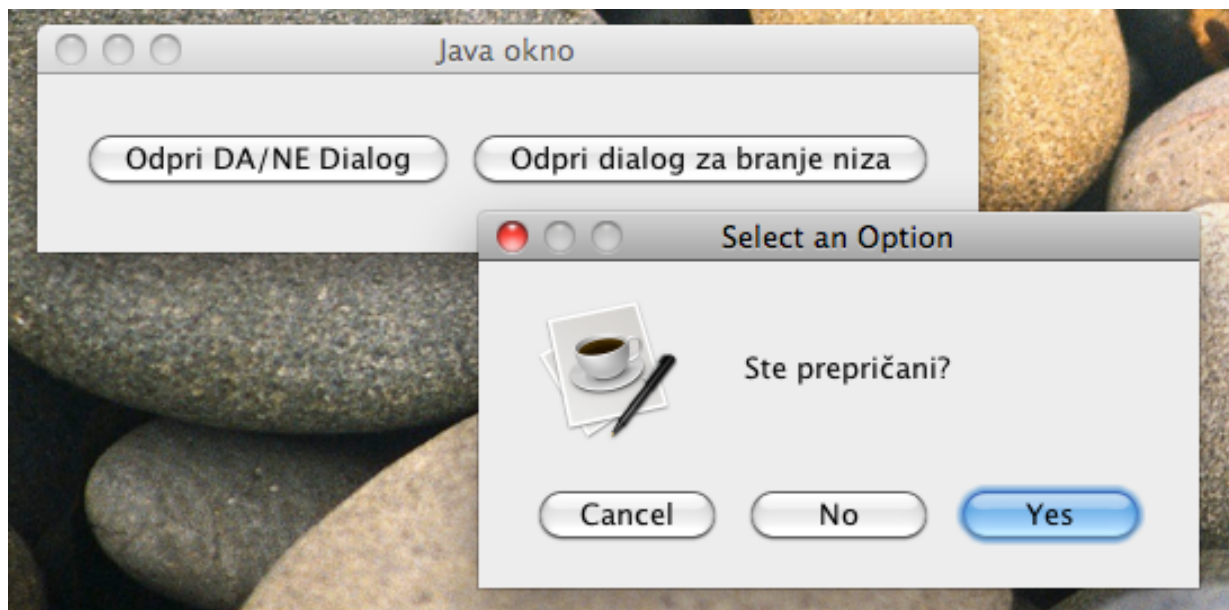
Okna razreda JOptionPane

- ▶ Razred JOptionPane uporabimo za prikaz standardnih dialogov za prikaz obvestil, branje niza, izbiro opcij, ...
- ▶ Nekateri JOptionPane metode:

Ime metode	Opis
showConfirmDialog	Potrditev (Da/Ne, ...)
showInputDialog	Vpis besedila
showMessageDialog	Okno s sporočilom
showOptionDialog	Splošen dialog (veliko opcij)



Napiši program, ki izriše okno (JFrame) z dvema gumboma. Ob pritisku na prvega, naj se odpre DA/NE dialog, ob pritisku na drugega pa dialog za branje niza...





Vsebina vpisnih polj

Vsebino vpisnega polja (`TextField`) preberemo z metodo `getText()`, nastavimo pa z metodo `setText()`.

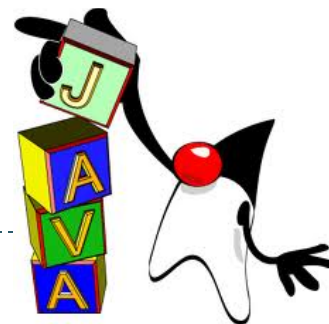
```
TextField polje = new TextField();  
polje.setText("Besedilo v polju");
```



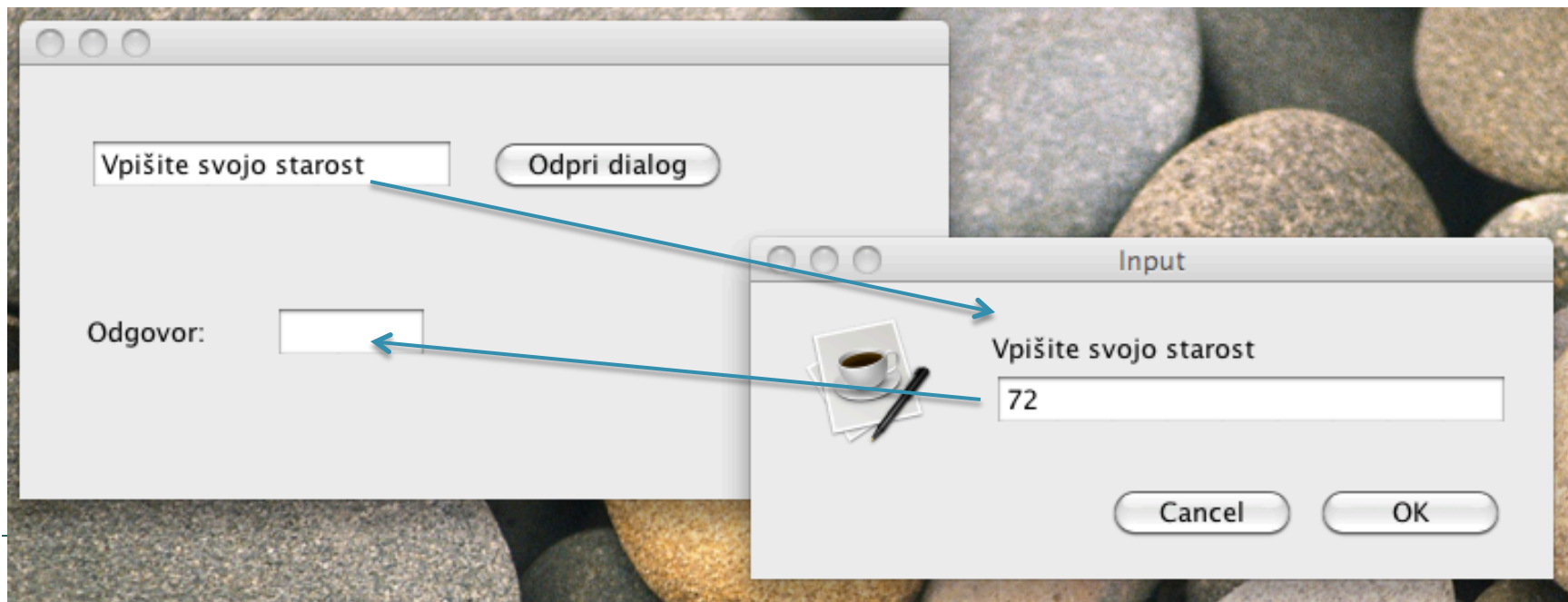
Naloga

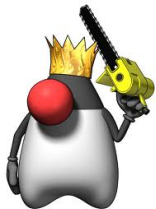
Branje in pisanje v vpisna polja

okna/SetGetText.java



Napišite program, ki ob kliku na gumb “Odpri dialog” odpre dialog za branje niza. Vprašanje v dialogu naj bo enako vsebini prvega vpisnega polja. Ob zaprtju dialoga, naj se odgovor vpiše v drugo vpisno polje.





Akcija ob pritisku na gumb

Če želimo, da se ob pritisku na gumb sproži neka akcija, moramo napisati in registrirati ustreznega poslušalca:

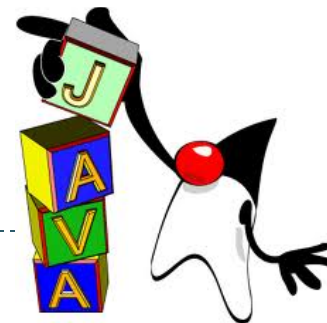
```
JButton gumb = new JButton("Klikni me");  
gumb.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent ae) {  
        // ... akcija, ki se zgodi ob kliku na gumb  
    }  
});
```



Naloga

Program za izračun cene sladoleda

okna/Sladoled.java



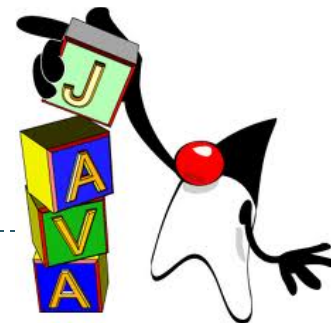
Napiši program za računanje cene sladoleda.

A screenshot of a Java Swing window titled "Sladoled". The window has a blue title bar with standard Windows-style controls (minimize, maximize, close). The main content area has a light gray background. It contains two labels: "Cena kepice" and "Stevilo kepic". Next to "Cena kepice" is a text input field containing the number "1". Next to "Stevilo kepic" is a text input field containing the number "5". At the bottom left, there is a button labeled "Racunaj". To the right of the button is an empty text input field for the result.

Cena kepice	1
Stevilo kepic	5
Racunaj	



Uporabniški grafični vmesnik (GUI)



Napiši preprost kalkulator, kot prikazuje spodnja slika;
kalkulator naj omogoča operacije $+$, $-$, $*$, $/$.

