



Grafické používateľské prostredie

Pojmy zavedené v 11. prednáške₍₁₎

- generická trieda
 - syntax
- typový parameter
 - konvencie

Pojmy zavedené v 11. prednáške₍₁₎

- možné riešenia v jazykoch
 - jedna generická trieda -> n tried pri preklade
 - jedna generická trieda -> n tried za behu
 - jedna generická trieda -> jedna trieda pri preklade

Pojmy zavedené v 11. prednáške₍₁₎

- výhody využitia generickej triedy
 - netreba pretypovávať
 - úplná typová kontrola pri preklade
 - nie je možné vložiť položku iného typu

Pojmy zavedené v 11. prednáške₍₁₎

- type erasure/odstraňovanie typov
 - problémy
 - použitie generickej triedy bez typových parametrov
- generické triedy a polymorfizmus
 - obmedzenia
 - syntax

Pojmy zavedené v 11. prednáške₍₁₎

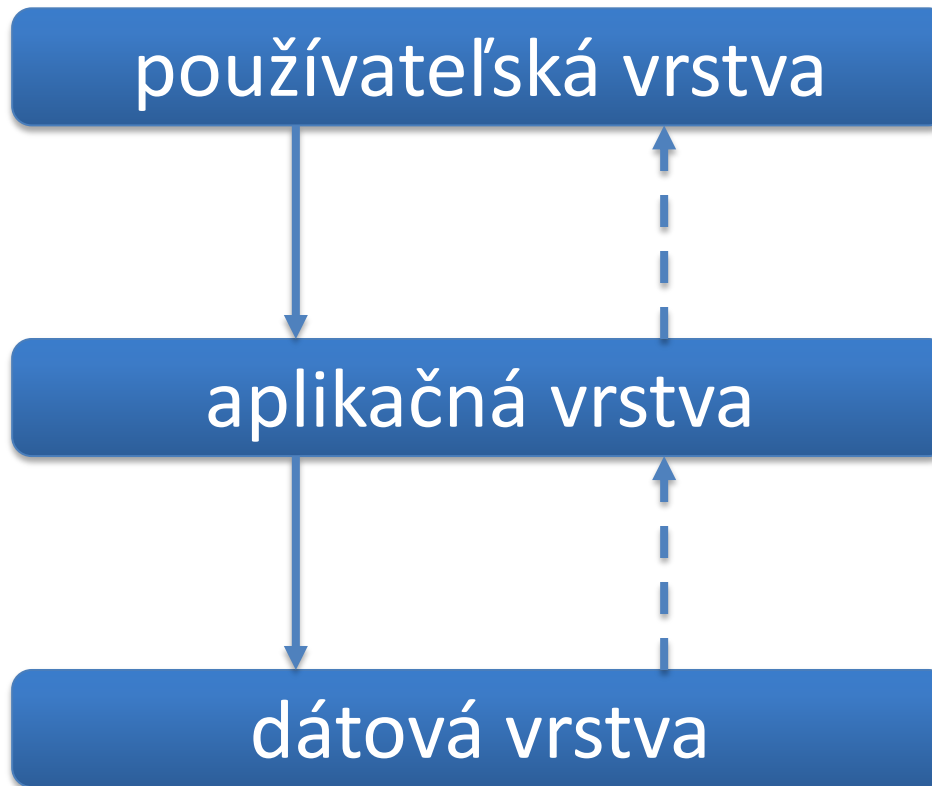
- iterátory a foreach
- generický interface
- generické metódy
 - automatické odvodzovanie typov
- divoké karty

Cieľ prednášky

- GUI
- vnorená trieda
- anonymná trieda
- príklad: kalkulačka

Trojvrstvový model aplikácie₍₁₎

- tri vrstvy sú voči sebe vo vzťahu klient – server



Trojvrstvový model aplikácie₍₂₎

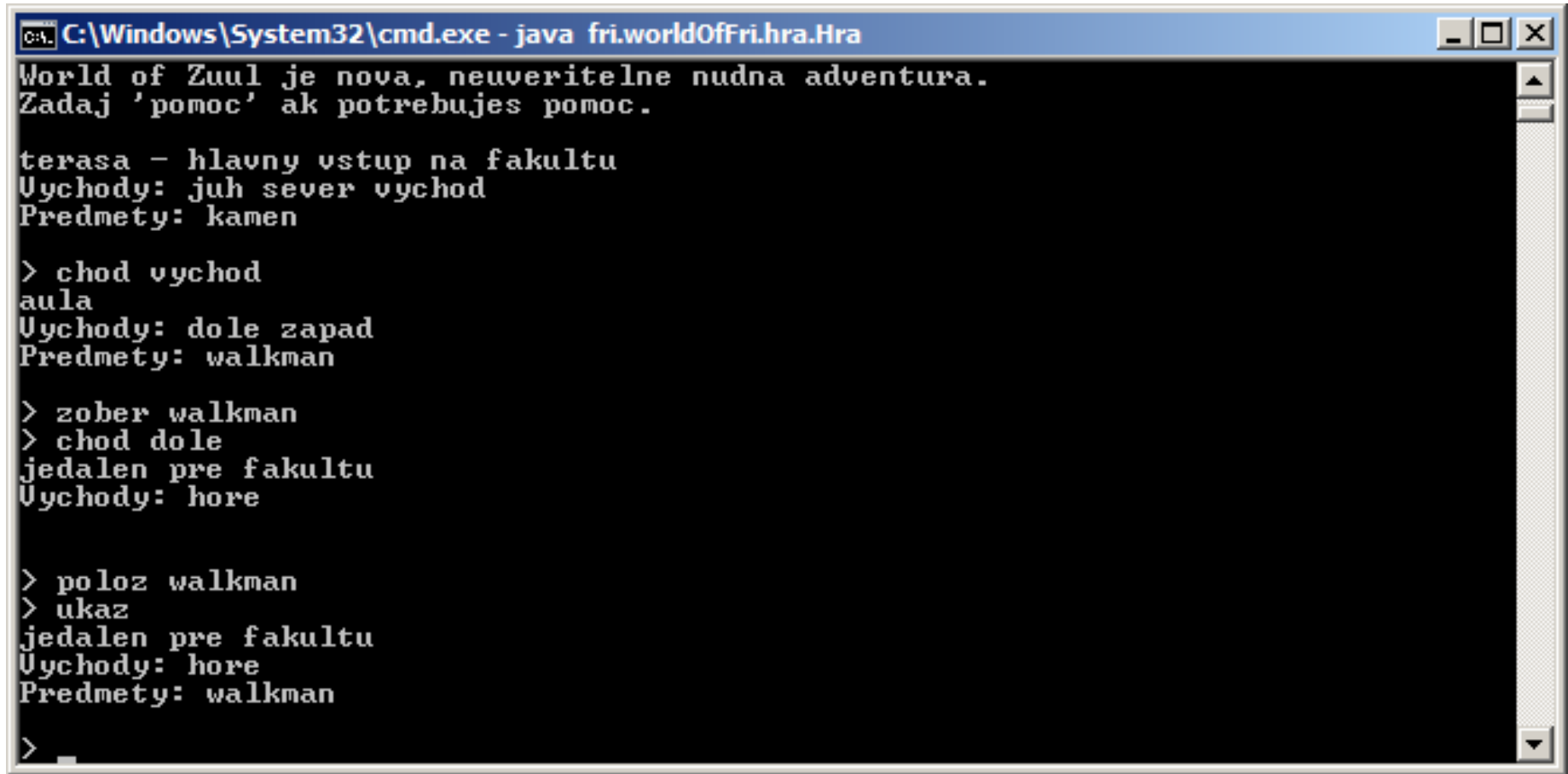
- tri skupiny – vrstvy – objektov aplikácie
- aplikačná (biznis) vrstva – objekty logiky aplikácie
- používateľská vrstva (GUI, TUI) – komunikácia človek – aplikácia, preberanie vstupných údajov, prezentácia výsledkov
- dátová vrstva – uloženie dát na opätovné použitie

Možnosti riešenia používateľskej vrstvy

- CLI – Command Line Interface
- TUI – Text User Interface
- GUI – Graphical User Interface

Command line interface

- použité v World of FRI



```
C:\Windows\System32\cmd.exe - java fri.worldOffri.hra.Hra
World of Zuul je nova, neuveritelne nudna aventura.
Zadaj 'pomoc' ak potrebujes pomoc.

terasa - hlavny vstup na fakultu
Uychody: juh sever vychod
Predmety: kamen

> chod vychod
aula
Uychody: dole zapad
Predmety: walkman

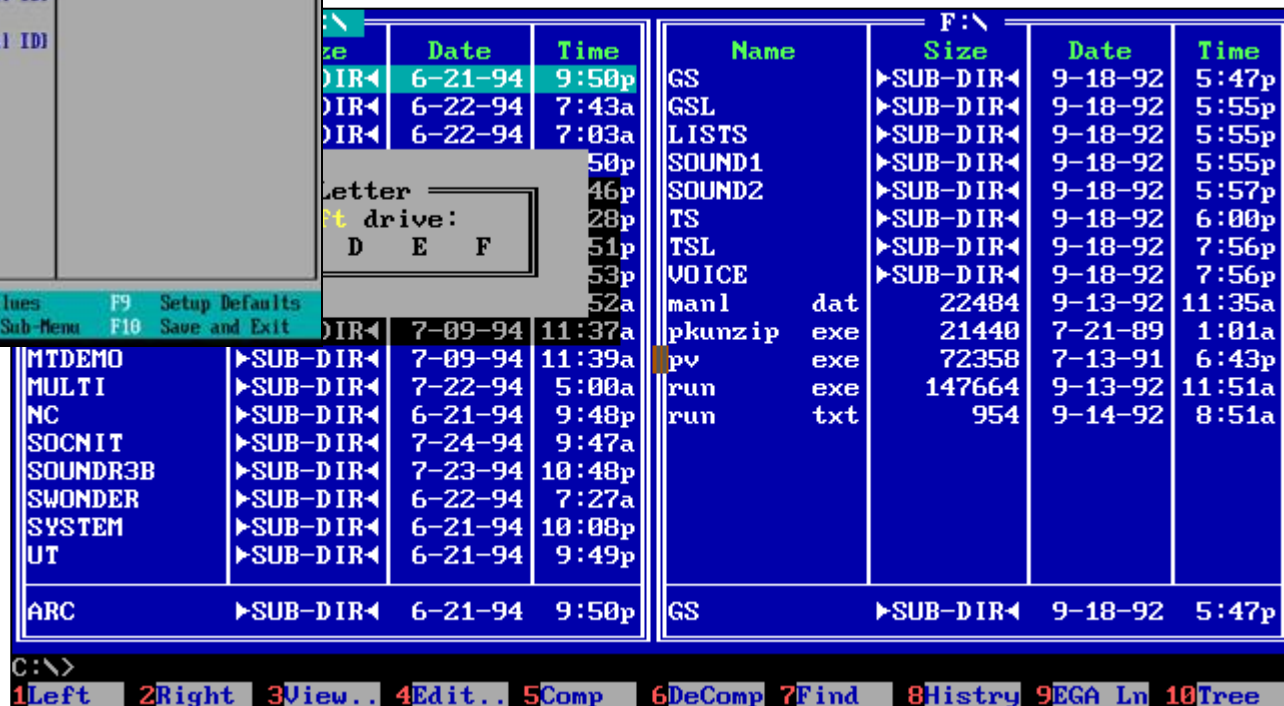
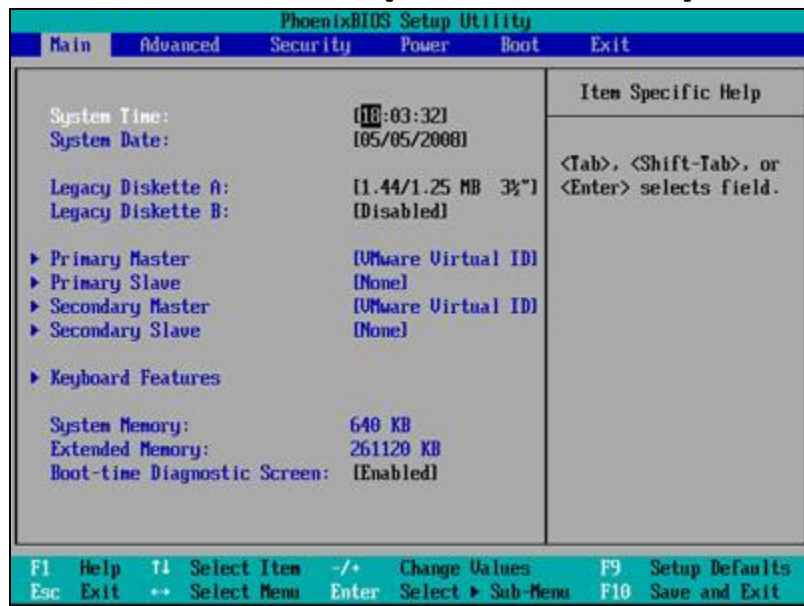
> zober walkman
> chod dole
jedalen pre fakultu
Uychody: hore

> poloz walkman
> ukaz
jedalen pre fakultu
Uychody: hore
Predmety: walkman

>
```

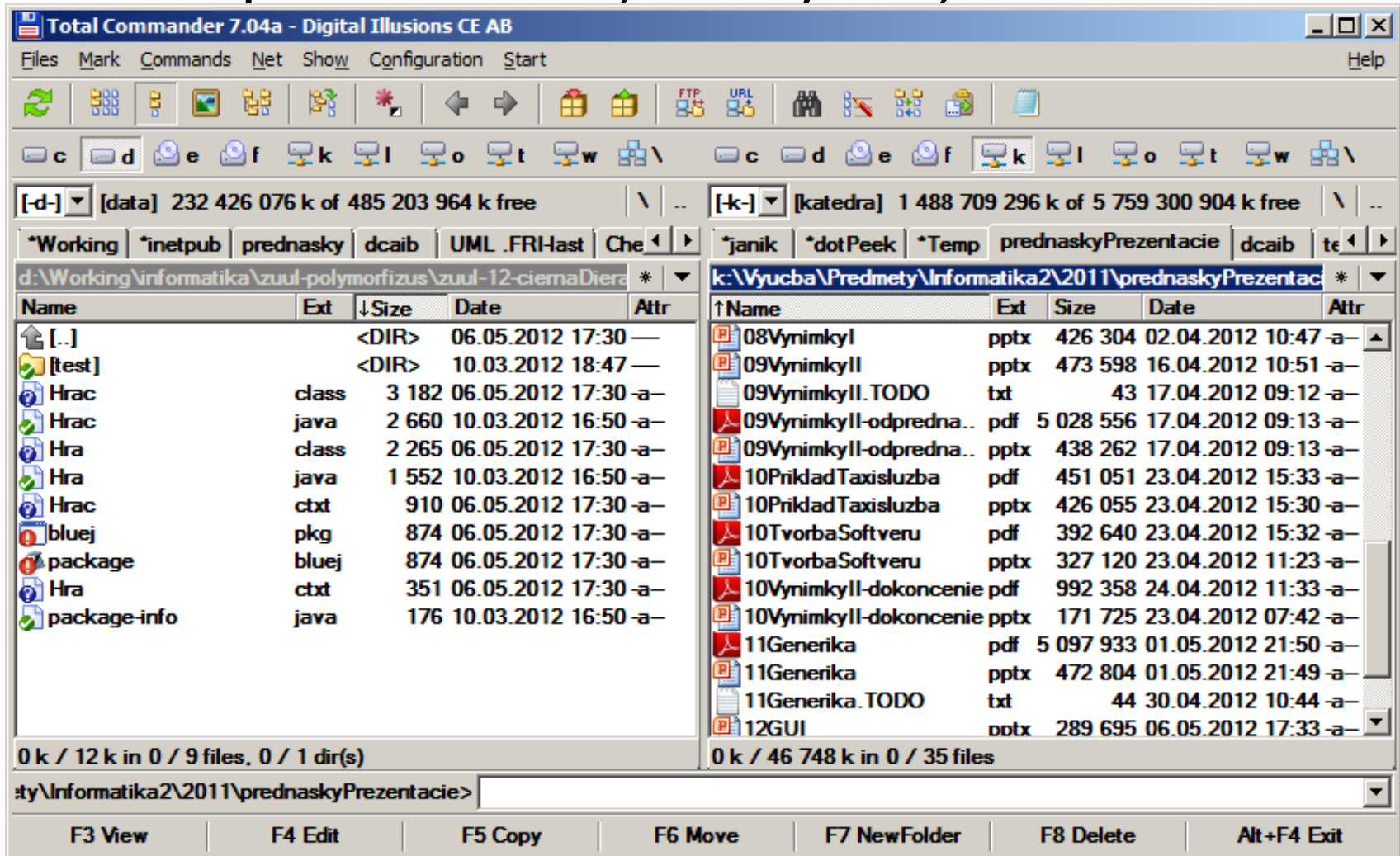
Text user interface

- BIOS setup, rôzne programy pre DOS



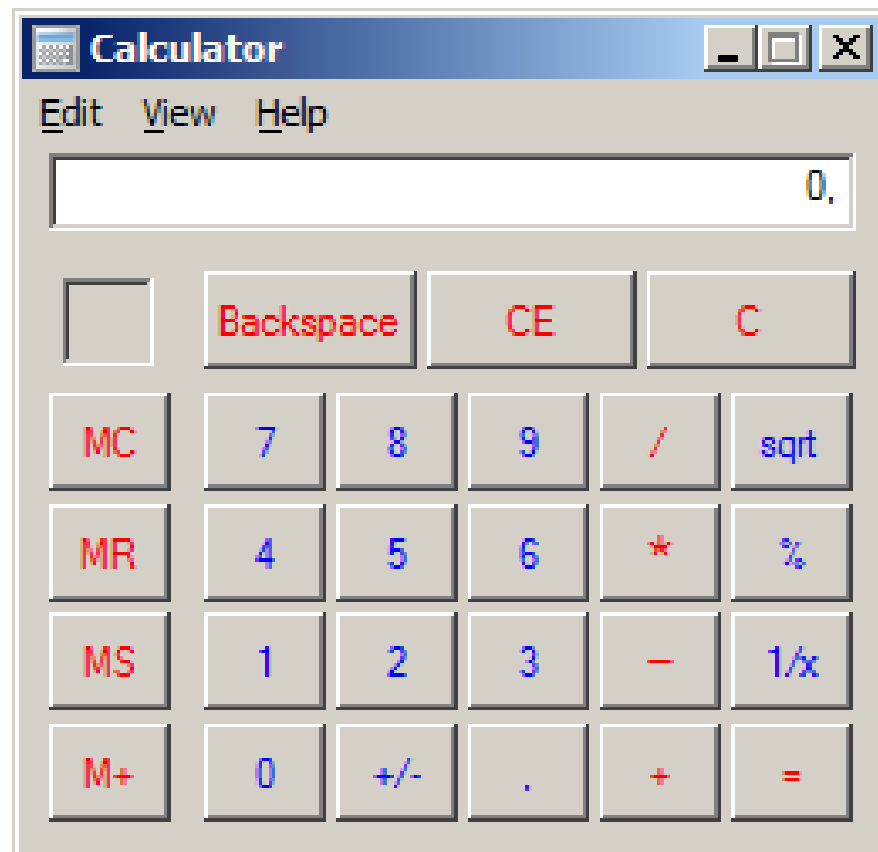
Graphical user interface

- softvér pre Windows, Linux/X11, MacOS X



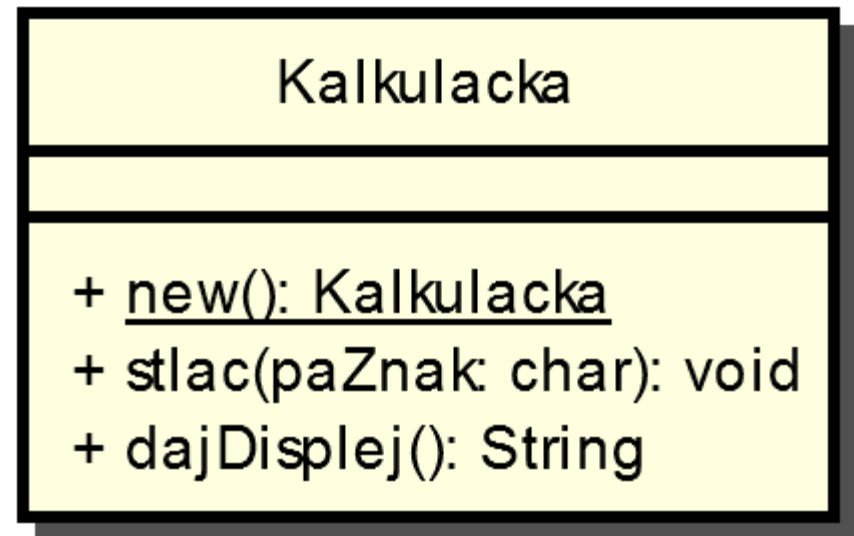
Zadanie – kalkulačka s GUI

- kalkulačka z prvého semestra – aplikačná vrstva
- dorobiť GUI
- konkurovať Microsoftu



Aplikačná vrstva

- displej
- čísla 0-9, desatinná čiarka
- operácie +, -, *, / a =

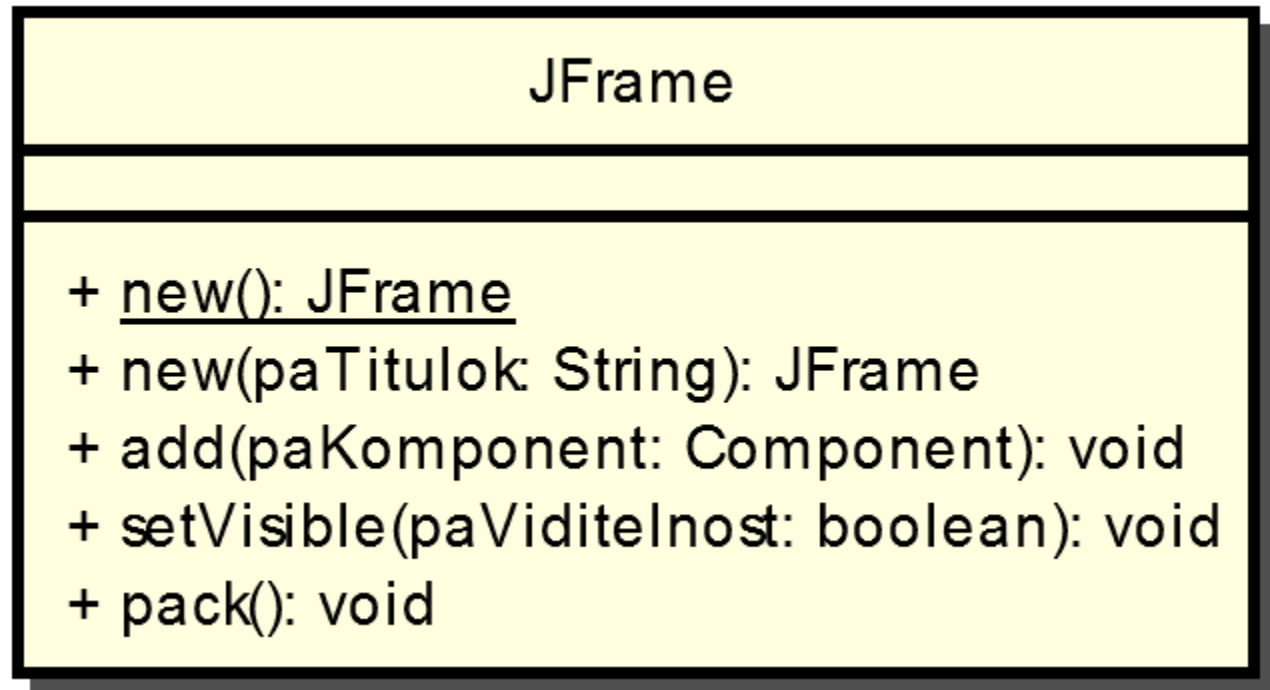


GUI v jazyku Java

- AWT – Abstract Window Toolkit – pôvodná knižnica pre jazyk Java
- Swing – nadstavba nad AWT

Okno v Swing₍₁₎

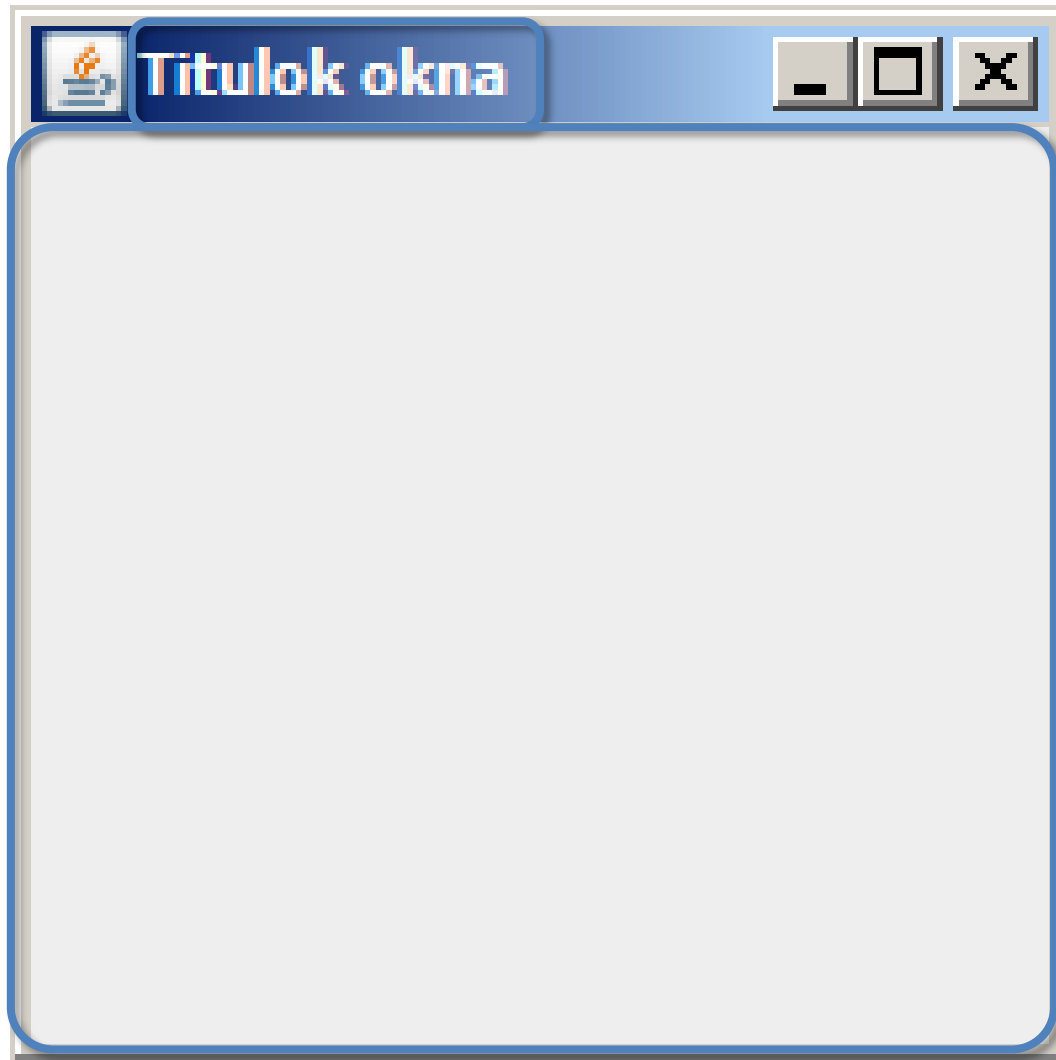
- časť GUI = objekt
- javax.swing.JFrame – okno aplikácie



Okno v Swing₍₂₎

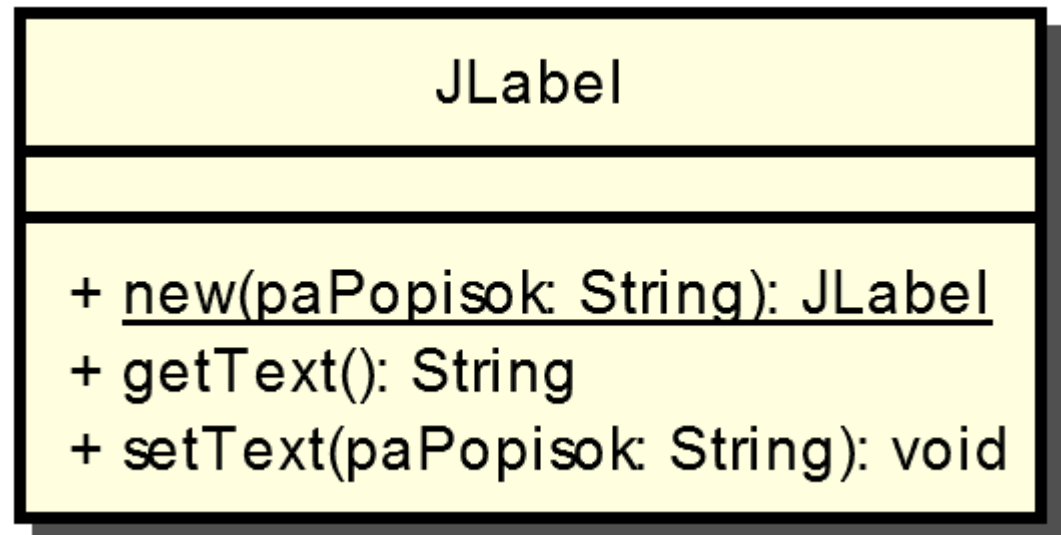
titulok

komponenty

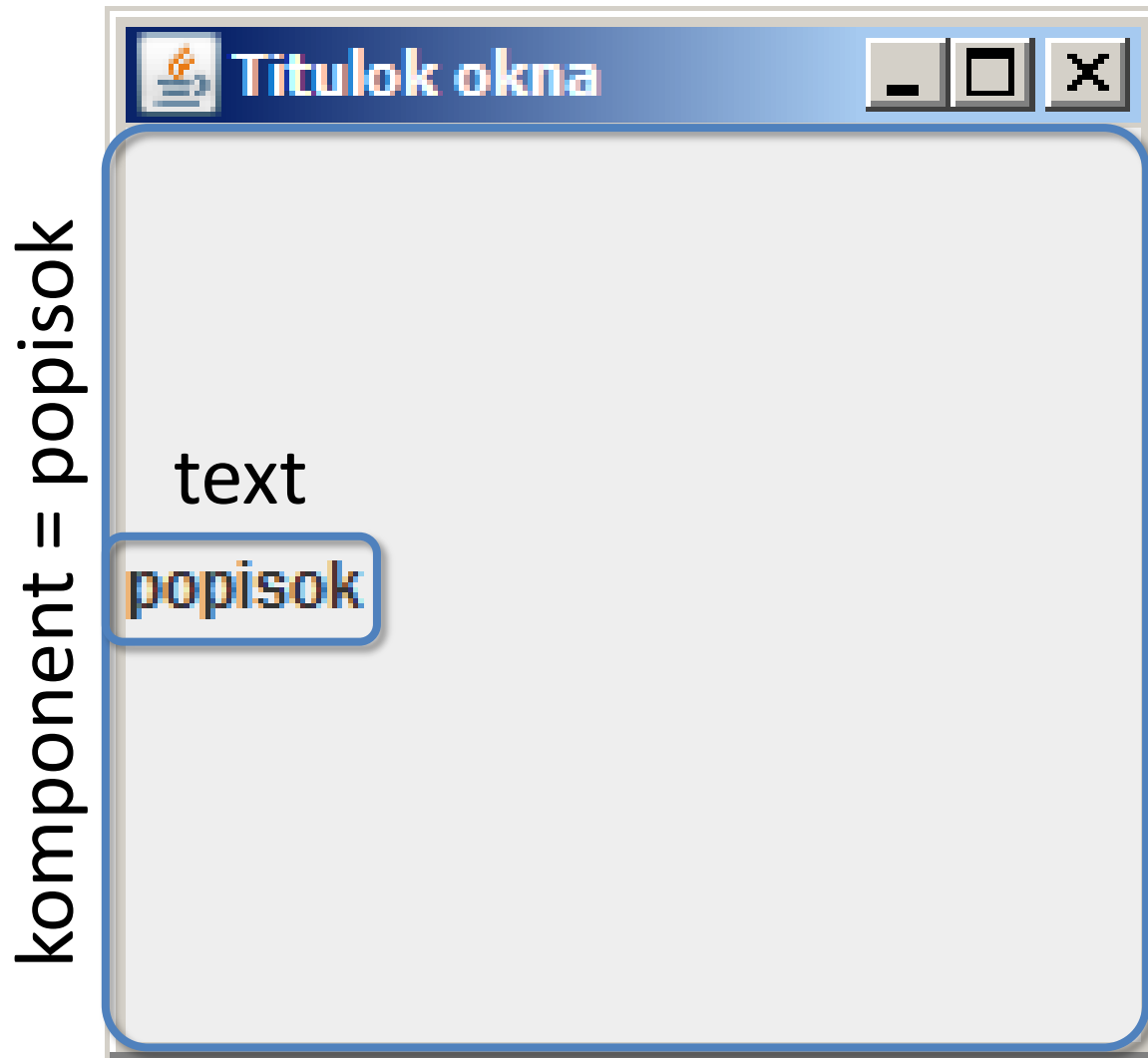


Popisok₍₁₎

- javax.swing.JLabel = popisok v okne

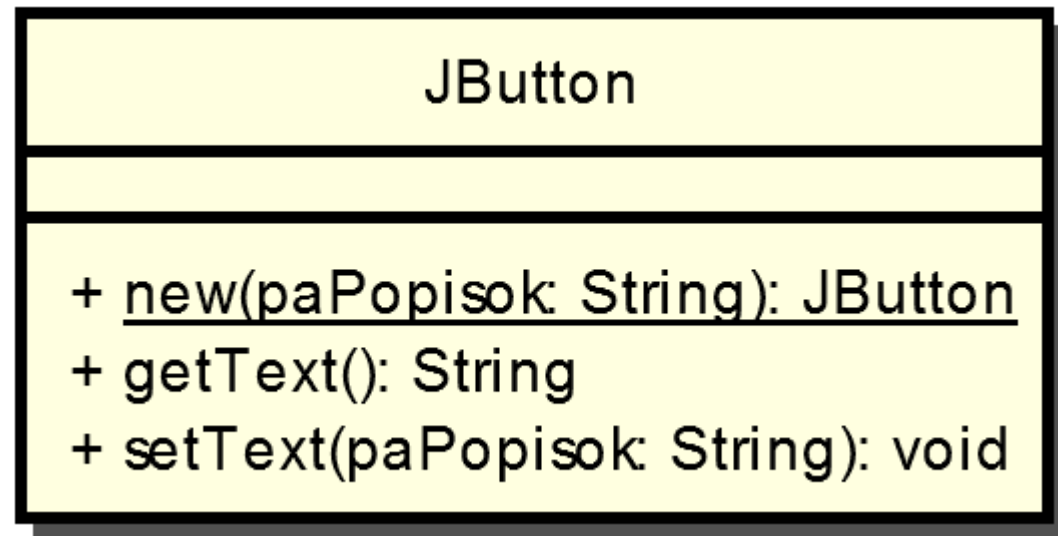


Popisok₍₂₎



Tlačidlo₍₁₎

- javax.swing.JButton = tlačidlo v okne



Tlačidlo₍₂₎

komponent = tlačidlo



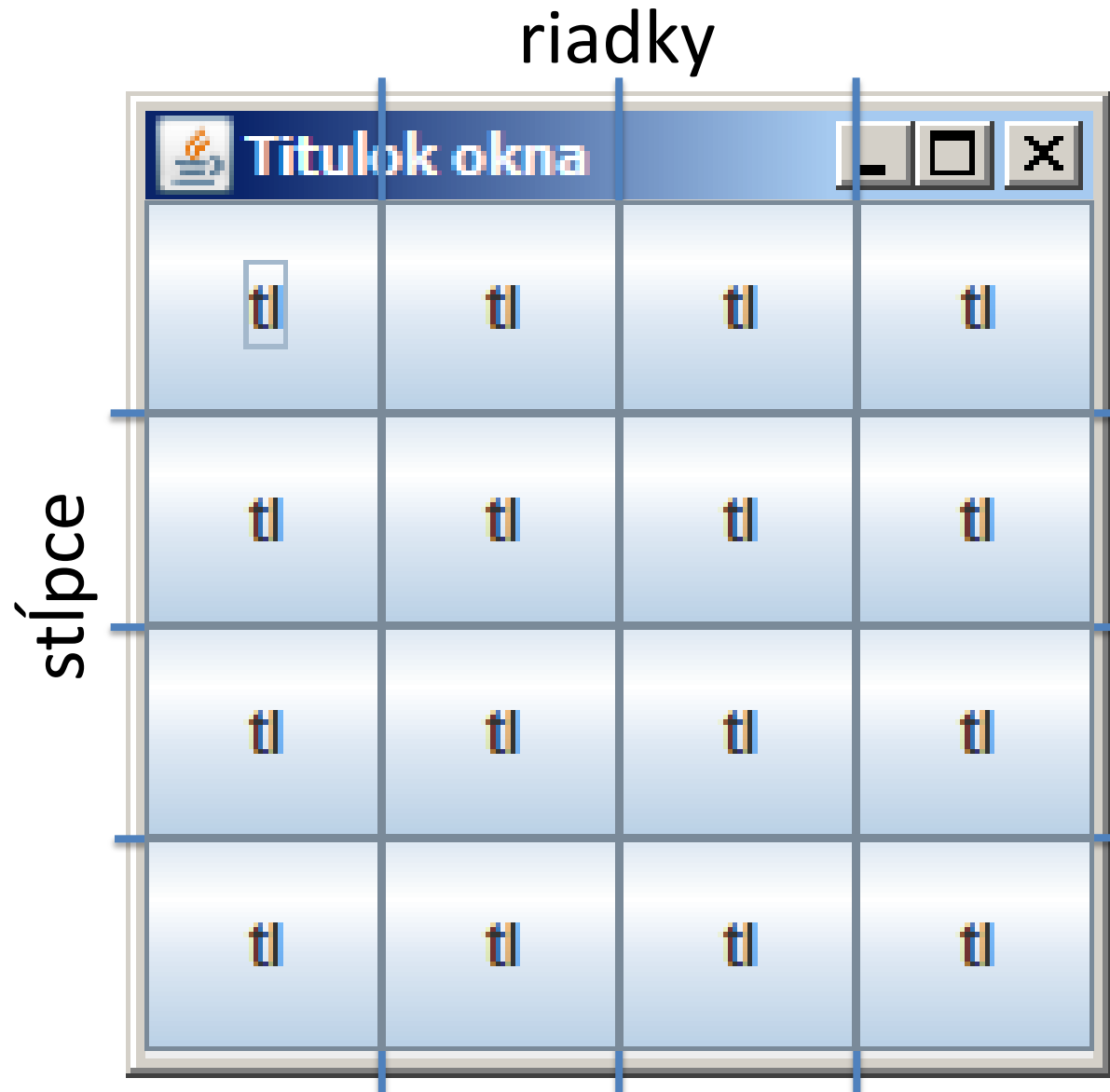
Kalkulačka

- 1x JFrame – hlavné okno kalkulačky
 - titulok = „Kalkulačka“
- 1x JLabel – displej
 - text = výsledok
- 16x JButton – klávesnica kalkulačky
 - 10 číslíc
 - desatinná čiarka
 - operátory +, -, *, /
 - operácia =

Viac prvkov v okne

- usporiadanie prvkov – Layout (rozmiestnenie)
- rôzne druhy

GridLayout₍₁₎

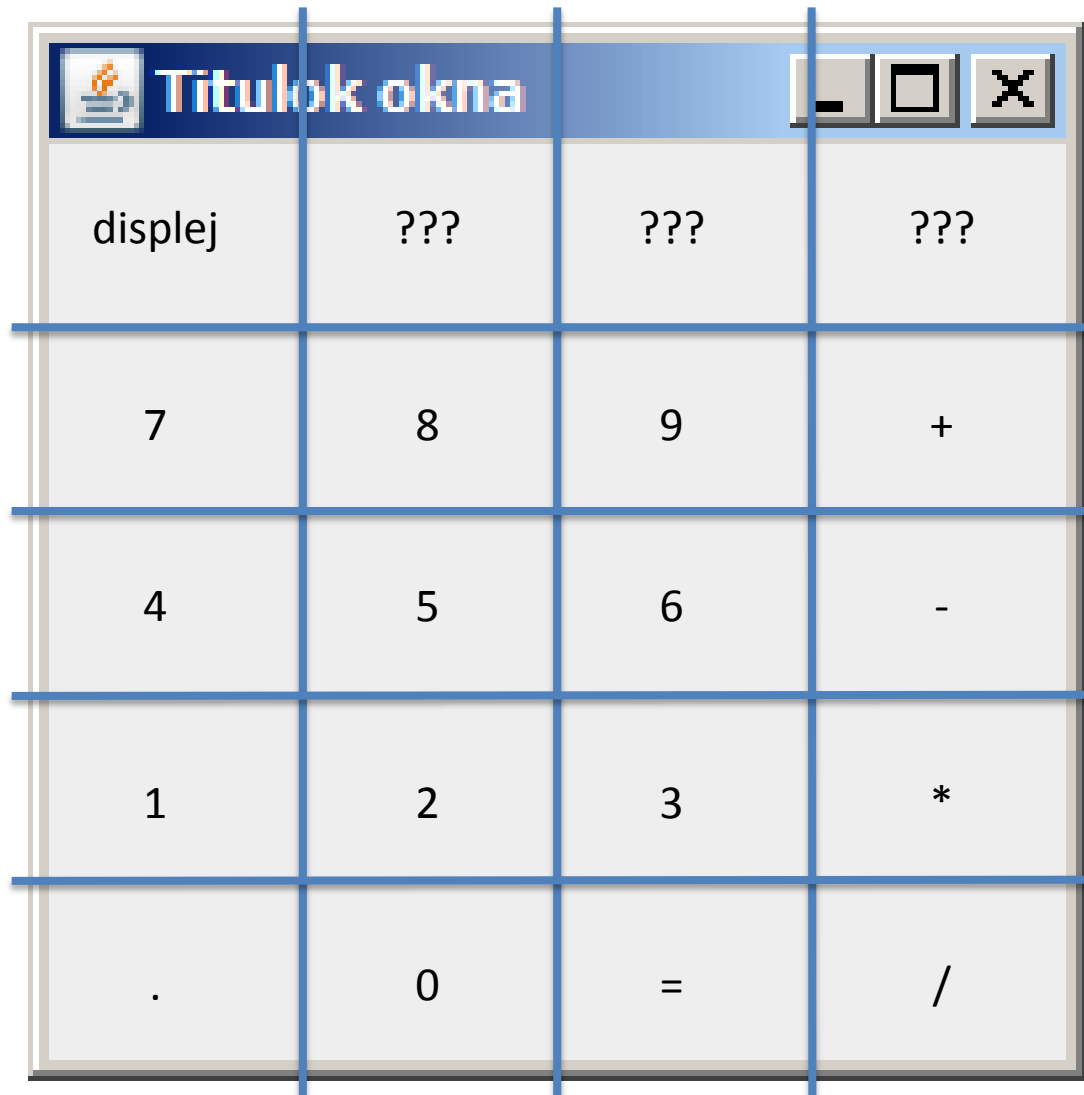


GridLayout₍₂₎

```
JFrame okno = new JFrame("Titulok okna");  
okno.setLayout(new GridLayout(4, 4));
```

```
okno.add(new JButton("t"));  
okno.add(new JButton("t"));  
okno.add(new JButton("t"));  
okno.add(new JButton("t"));  
okno.add(new JButton("t"));  
...
```

Kalkulačka pomocou GridLayout



Implementácia kalkulačky₍₁₎

```
public class Hlavna
{
    public static void main(String[] paParametre)
    {
        ...
    }
}
```

Implementácia kalkulačky₍₂₎

```
JFrame okno = new JFrame("Kalkulacka");
```

```
okno.setLayout(new GridLayout(5, 4));
```

```
okno.add(new JLabel("0"));
```

```
okno.add(new JLabel(""));
```

```
okno.add(new JLabel(""));
```

```
okno.add(new JLabel(""));
```

```
...
```

Implementácia kalkulačky₍₃₎

```
okno.add(new JButton("7"));  
okno.add(new JButton("8"));  
okno.add(new JButton("9"));  
okno.add(new JButton("+"));
```

...

```
okno.pack();  
okno.setVisible(true);
```

Možnosti programovania GUI

- všetko v main()
 - „slížový“, neprehľadný kód
 - komplikovaná údržba
- samostatná trieda pre každé okno
 - okno sa vytvorí v konštruktore
 - správy/metódy na obsluhu okna
 - napr. zobrazenie/skrytie okna
 - v main() len použitie objektu okna
- samostatná trieda pre časti okna
 - pri komplikovanom GUI

Implementácia Hlavna

```
public class Hlavna
{
    public static void main(String[] paParametre)
    {
        HlavneOkno o = new HlavneOkno();
        o.zobraz();
    }
}
```


Okno ako samostatná trieda

- možnosti implementácie
- potomok triedy JFrame – dedičnosť
- atribút – inštancia triedy JFrame – skladanie
- prvá možnosť – narušené zapuzdrenie
- zverejnené operácie s oknom
 - zmena titulku
 - pridávanie komponentov
 - ...

Implementácia HlavneOkno₍₁₎

```
public class HlavneOkno
{
    private JFrame aOkno;

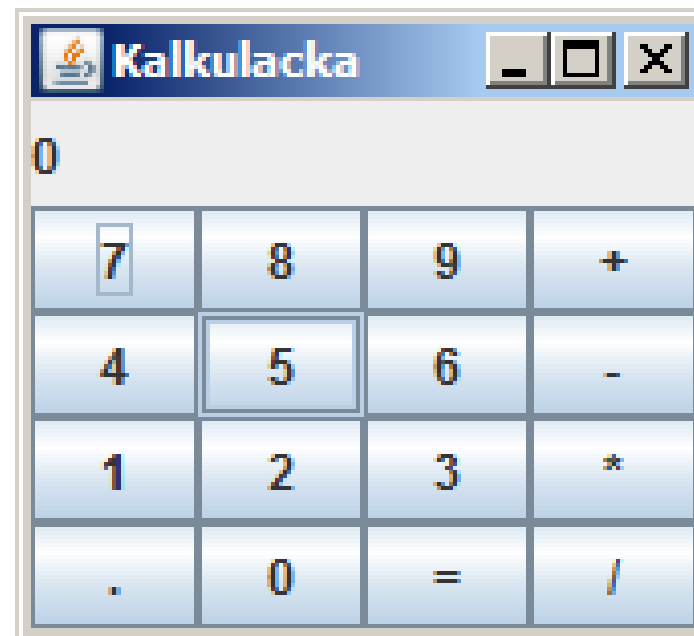
    ...
}
```

Implementácia HlavneOkno₍₂₎

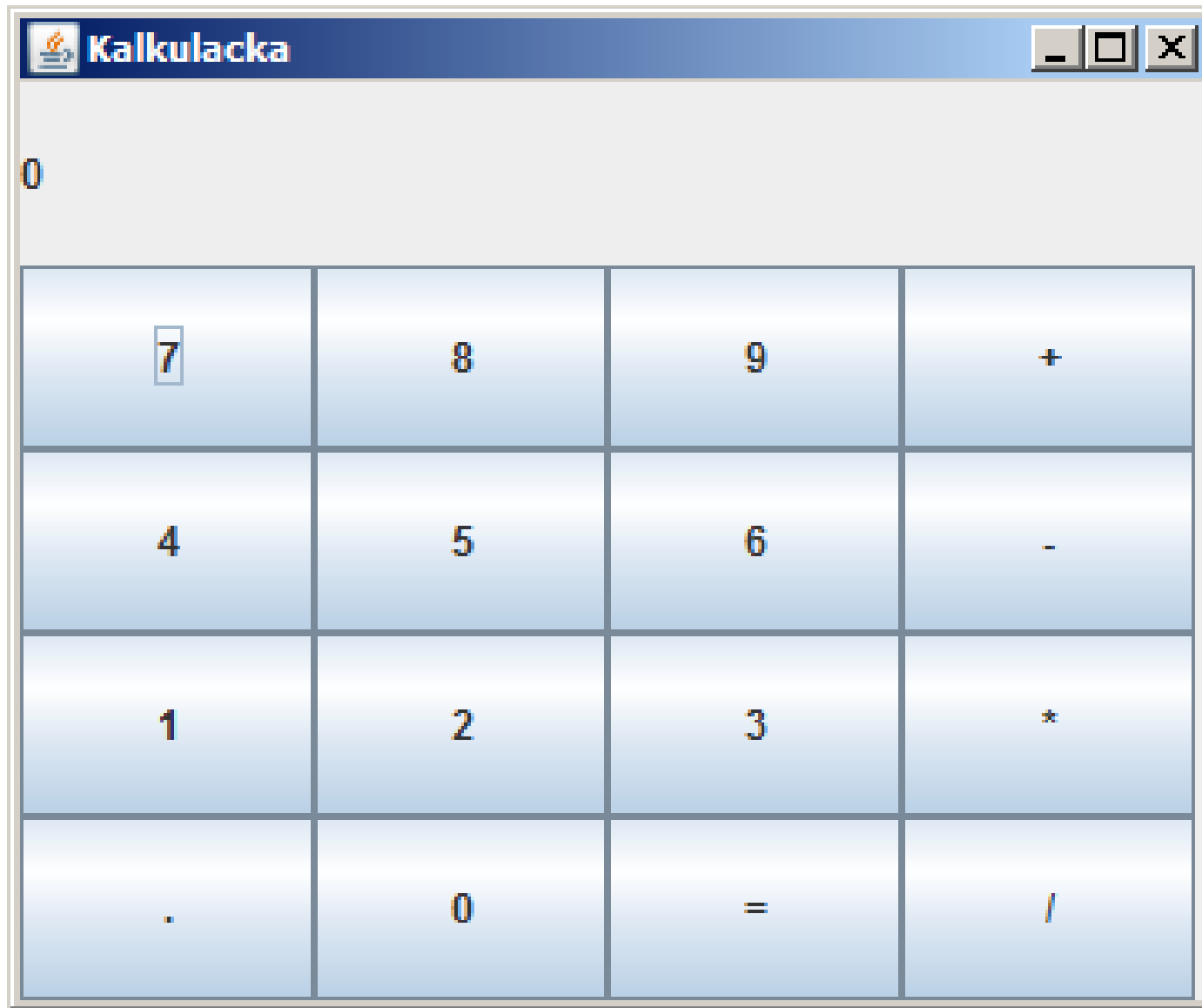
```
public HlavneOkno()
{
    aOkno = new JFrame("Kalkulacka");
    ...
}

public void zobraz()
{
    aOkno.setVisible(true);
}
```

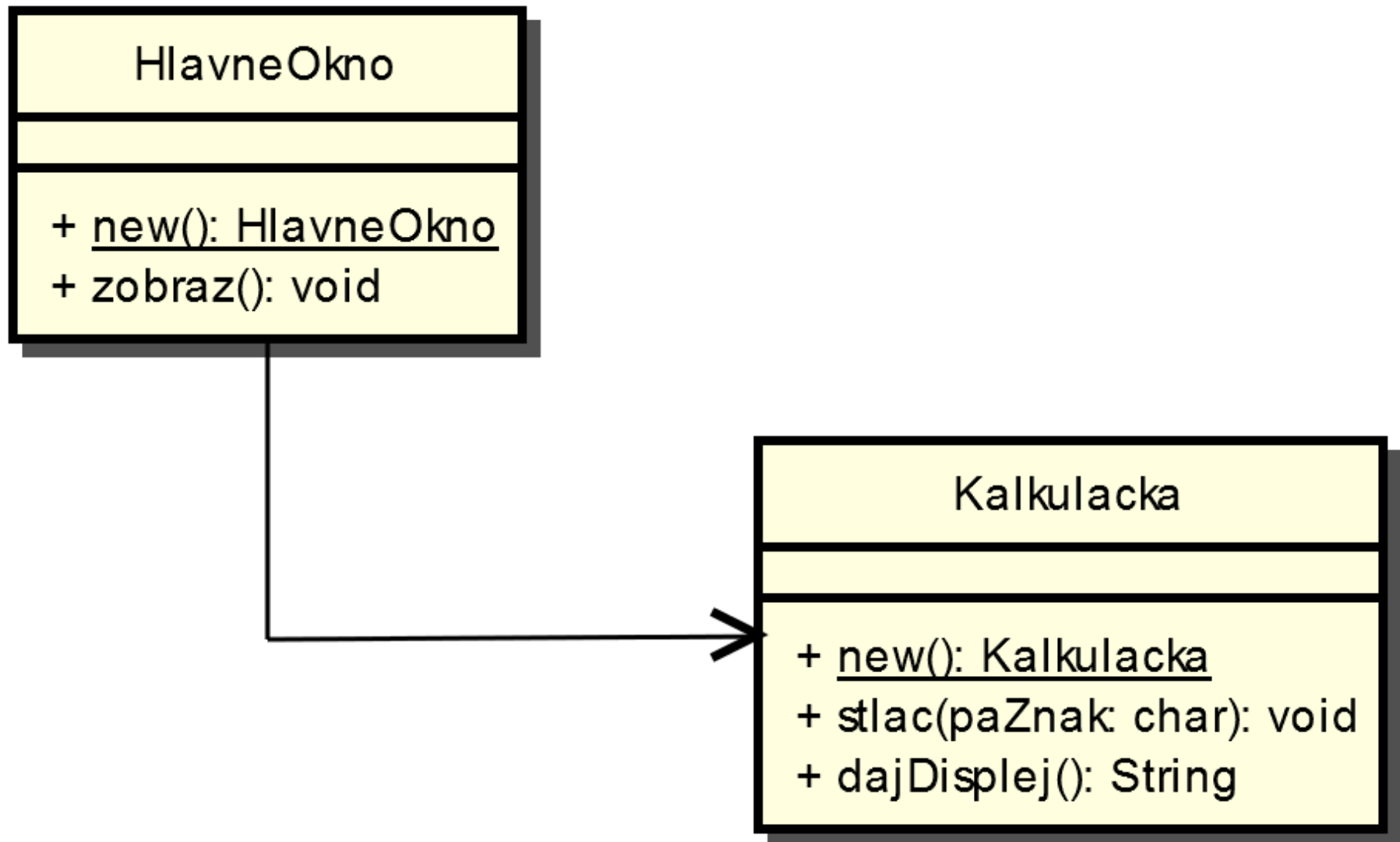
Výsledok



Zväčšovanie okna kalkulačky



Spolupráca okna s kalkulačkou



- udalosť
 - kliknutie myšou
 - pohyb myšou
 - stlačenie klávesy
 - použitie tlačidla
- pri udalosti pošle komponent, ktorý ju zaznamenal, správu registrovanému objektu

Registrácia na príjem udalostí

- poslucháč (listener) – objekt prijímajúci správy o udalosti
- interface *Listener
 - ActionListener
 - actionPerformed
 - MouseListener
 - mouseClicked, mouseEntered, mouseExited, ...
 - KeyListener
 - keyPressed, keyReleased, keyTyped
 - WindowListener
 - windowActivated, windowClosing, windowIconified, ...
 - ...

PosluchacTlacidiel₍₁₎

```
public void actionPerformed(ActionEvent e)
{
    aKalkulacka.stlac(aOperacia);
    aDisplej.setText(aKalkulacka.dajDisplej());
}
```

PosluchacTlacidiel₍₂₎

```
public class PosluchacTlacidiel
    implements ActionListener
{
    private Kalkulacka aKalkulacka;
    private JLabel aDisplej;
    private char aOperacia;

    ...
}
```

PosluchacTlacidiel₍₃₎

```
public PosluchacTlacidiel(char paOperacia,  
                           JLabel paDisplej, Kalkulacka paKalkulacka)  
{  
    aOperacia = paOperacia;  
    aDisplej = paDisplej;  
    aKalkulacka = paKalkulacka;  
}
```

Integrácia poslucháča do kalkulačky

- vytvorenie inštancie kalkulačky ako atribútu
- pre každé tlačidlo
 - vytvorenie príslušného poslucháča
 - nastavenie poslucháča ako ActionListener

Implementácia HlavneOkno₍₁₎

```
public class HlavneOkno
{
    private JFrame aOkno;
    private Kalkulacka aKalkulacka;
    private JLabel aDisplej;
    ...
}
```

Implementácia HlavneOkno₍₂₎

```
public HlavneOkno()  
{  
    aOkno = new JFrame("Kalkulacka");  
    aDisplej = new JLabel("0");  
    aKalkulacka = new Kalkulacka();  
    ...  
}
```

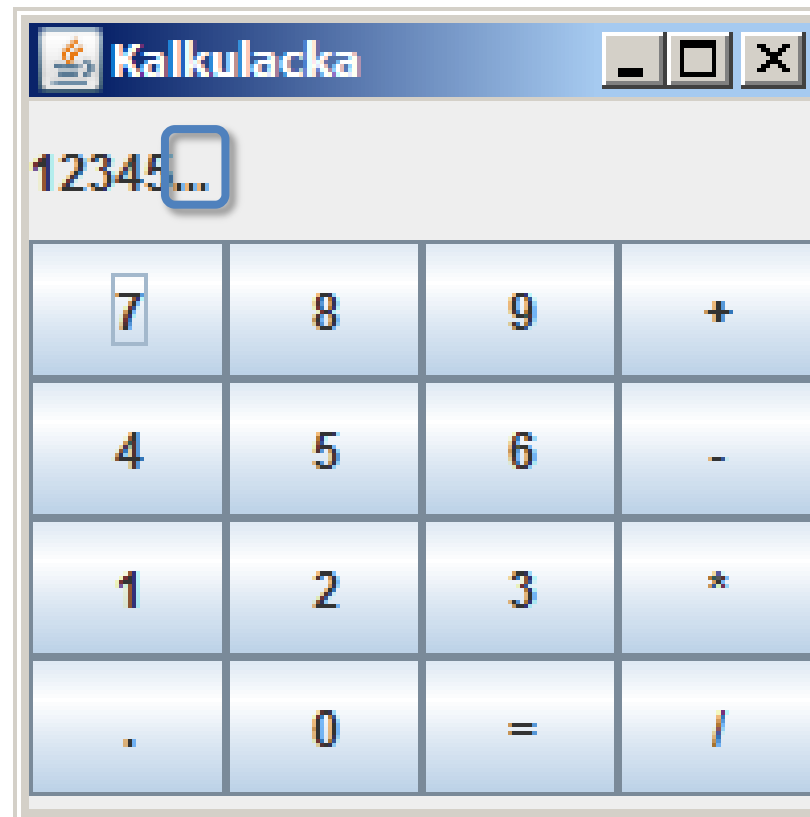
Implementácia HlavneOkno₍₃₎

```
aOkno.add(aDisplej);  
aOkno.add(new JLabel(""));  
aOkno.add(new JLabel(""));  
aOkno.add(new JLabel(""));  
...
```

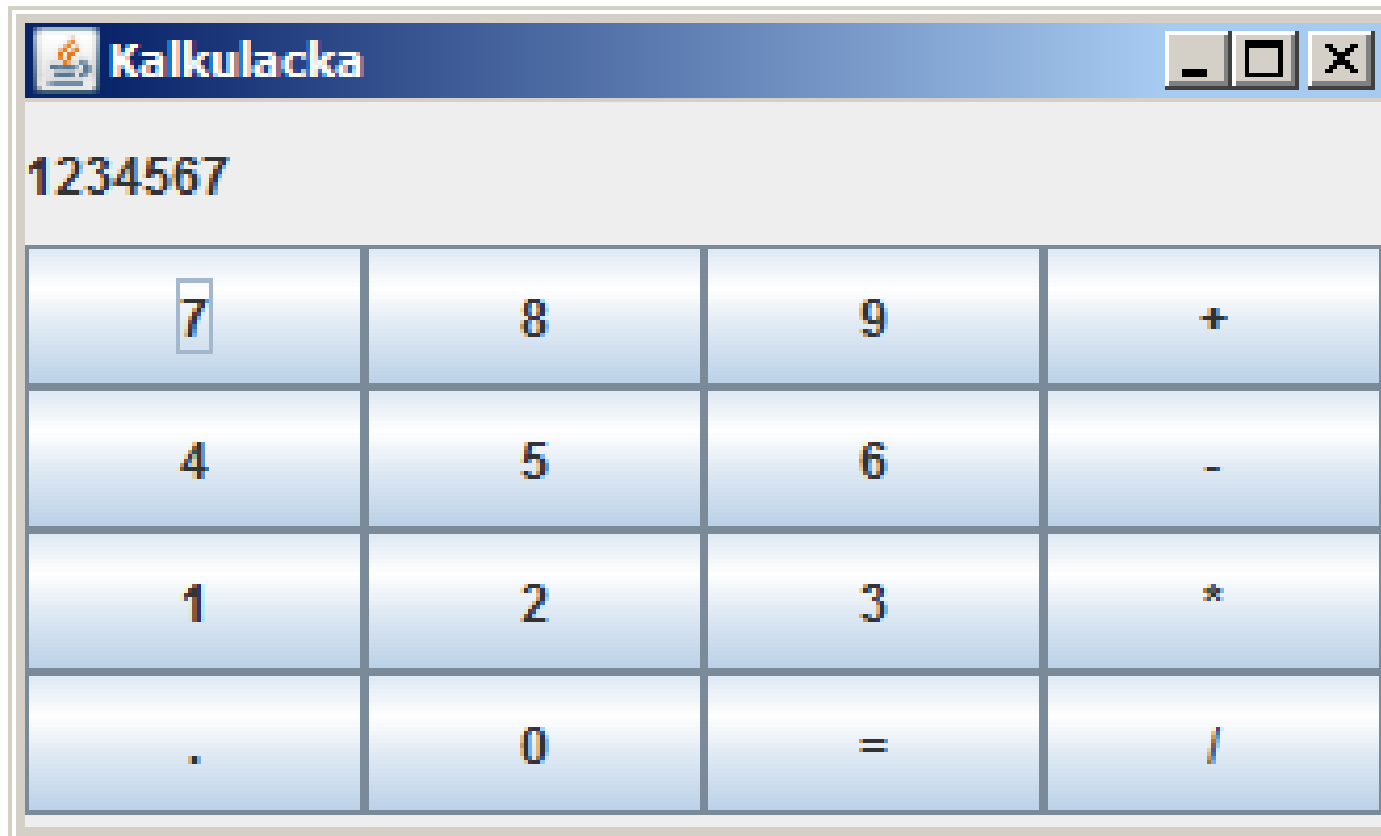
Implementácia HlavneOkno₍₄₎

```
JButton tlacidlo = new JButton("7");  
tlacidlo.addActionListener(  
    new PosluchacTlacidiel(  
        '7', aDisplej, aKalkulacka  
    ));  
aOkno.add(tlacidlo);
```

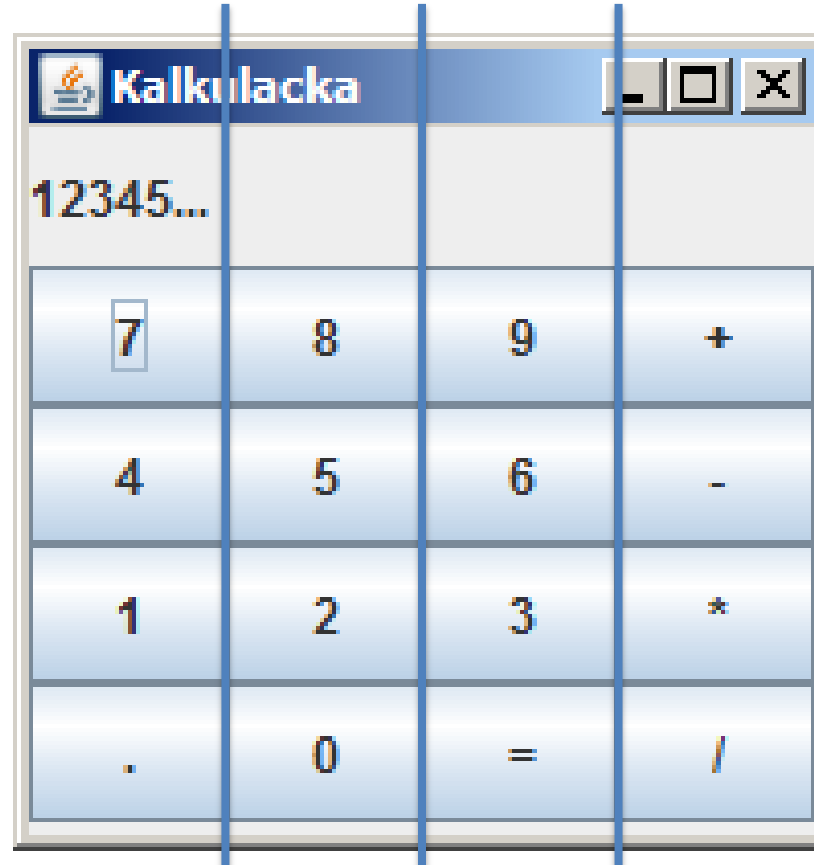

Problém s displejom₍₁₎



Problém s displejom₍₂₎



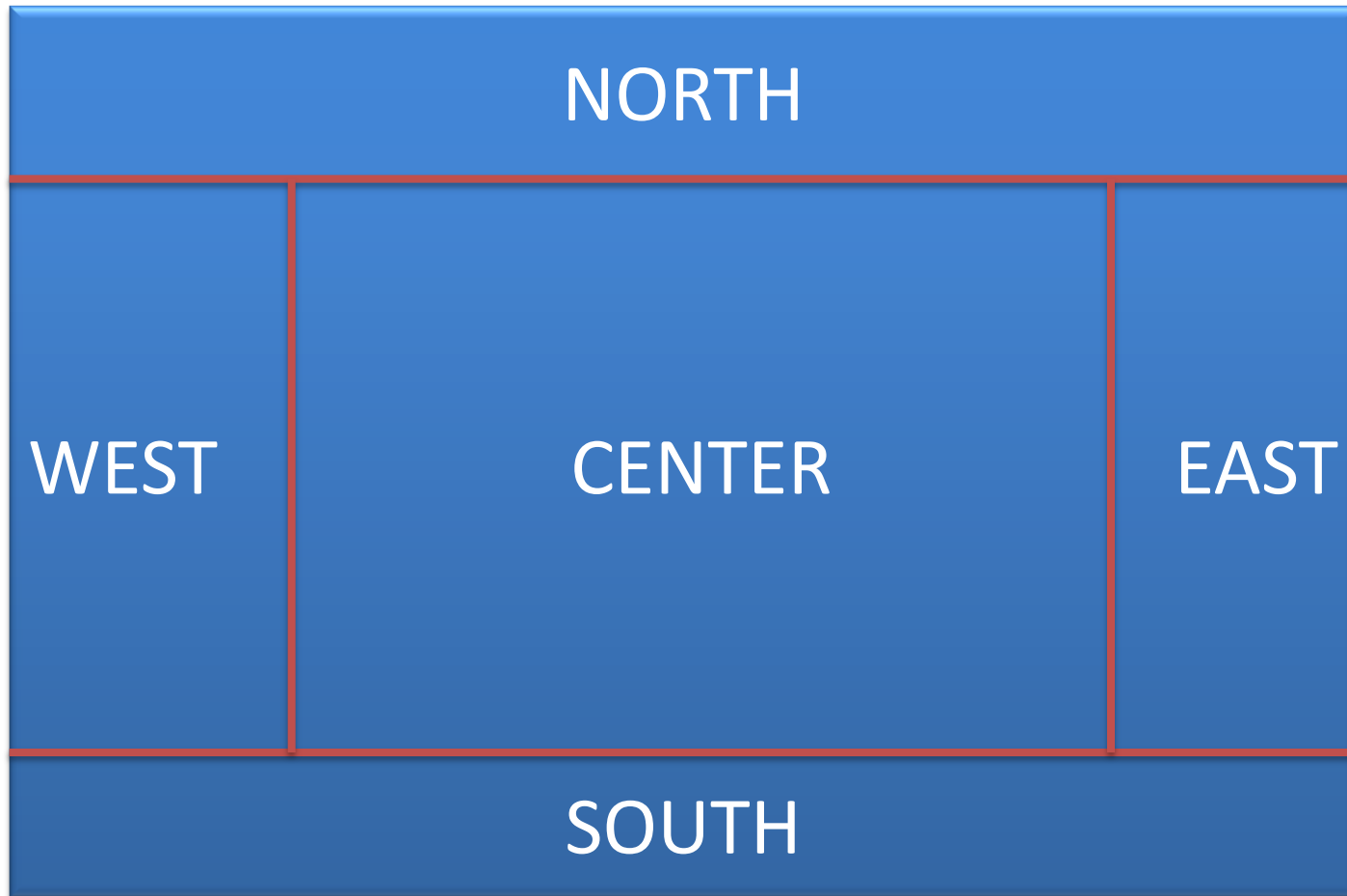
Problém objasnený



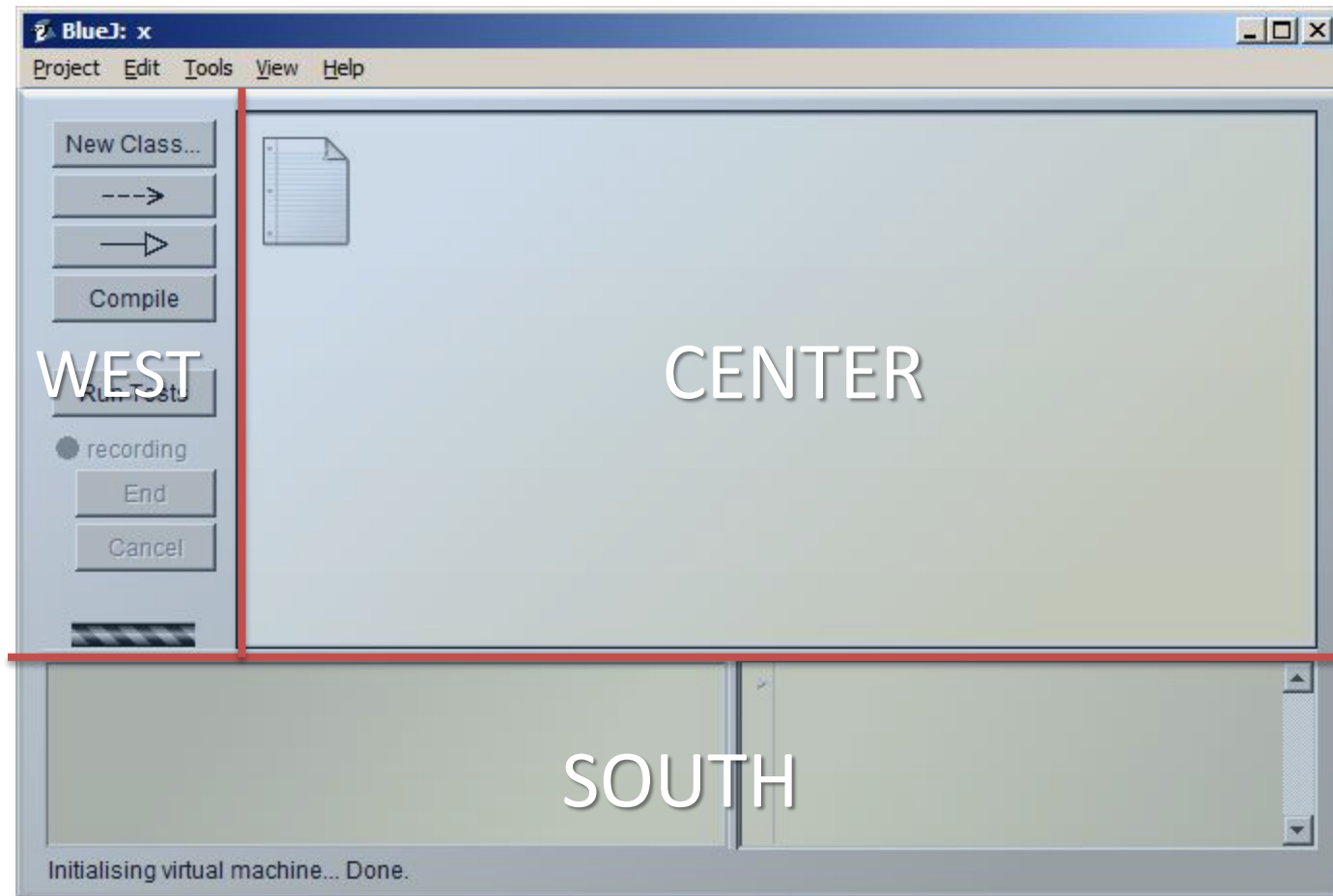
GUI komponenty – kontajnery

- každý komponent je kontajnerom
 - možnosť vkladať ďalšie komponenty
 - možnosť nastaviť layout
- komponent JPanel
 - všeobecný kontajner
- možnosť kombinovať layouty
 - každý kontajner môže mať vlastný layout
 - možnosť vnárať komponenty

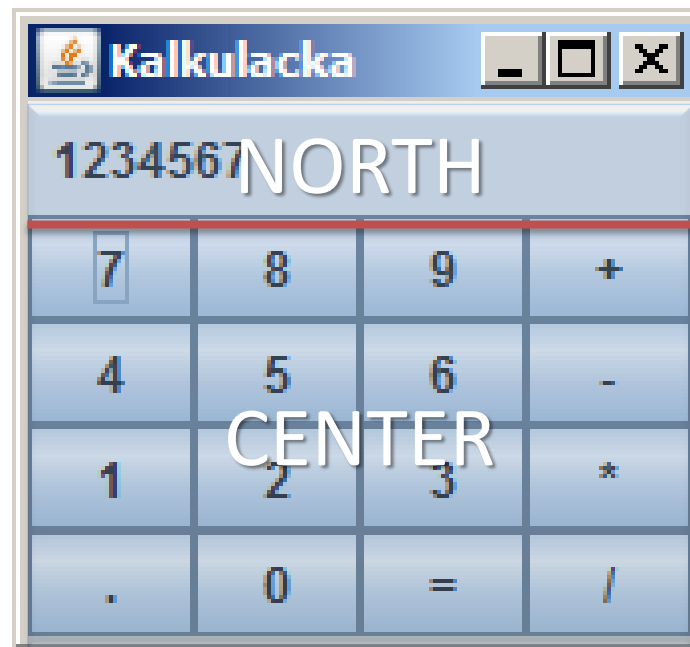
BorderLayout



BorderLayout v BlueJ



BorderLayout v Kalkulačke



Kombinácia viac Layout

```
aOkno.setLayout(new BorderLayout());
```

```
aDisplej = new JLabel("0");
```

```
aOkno.add(aDisplej, BorderLayout.NORTH);
```

```
aKlavesnica = new JPanel();
```

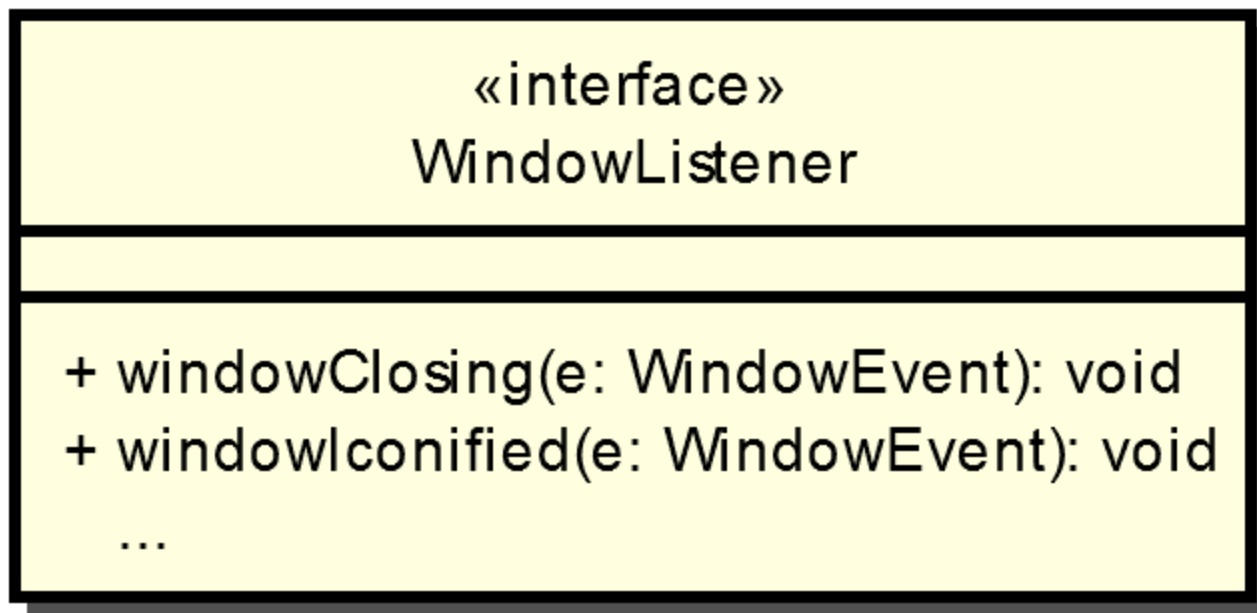
```
aOkno.add(aKlavesnica, BorderLayout.CENTER);
```

```
aKlavesnica.setLayout(new GridLayout(4, 4));
```


Ukončenie aplikácie

- po zavretí okna aplikácia beží ďalej
- jediná možnosť ukončenia
 - `System.exit(0);`
- nutnosť odchytiť udalosť „zatvorenie okna“
 - poslucháč: `WindowListener`
 - správa: `windowClosing`

Poslucháč WindowListener

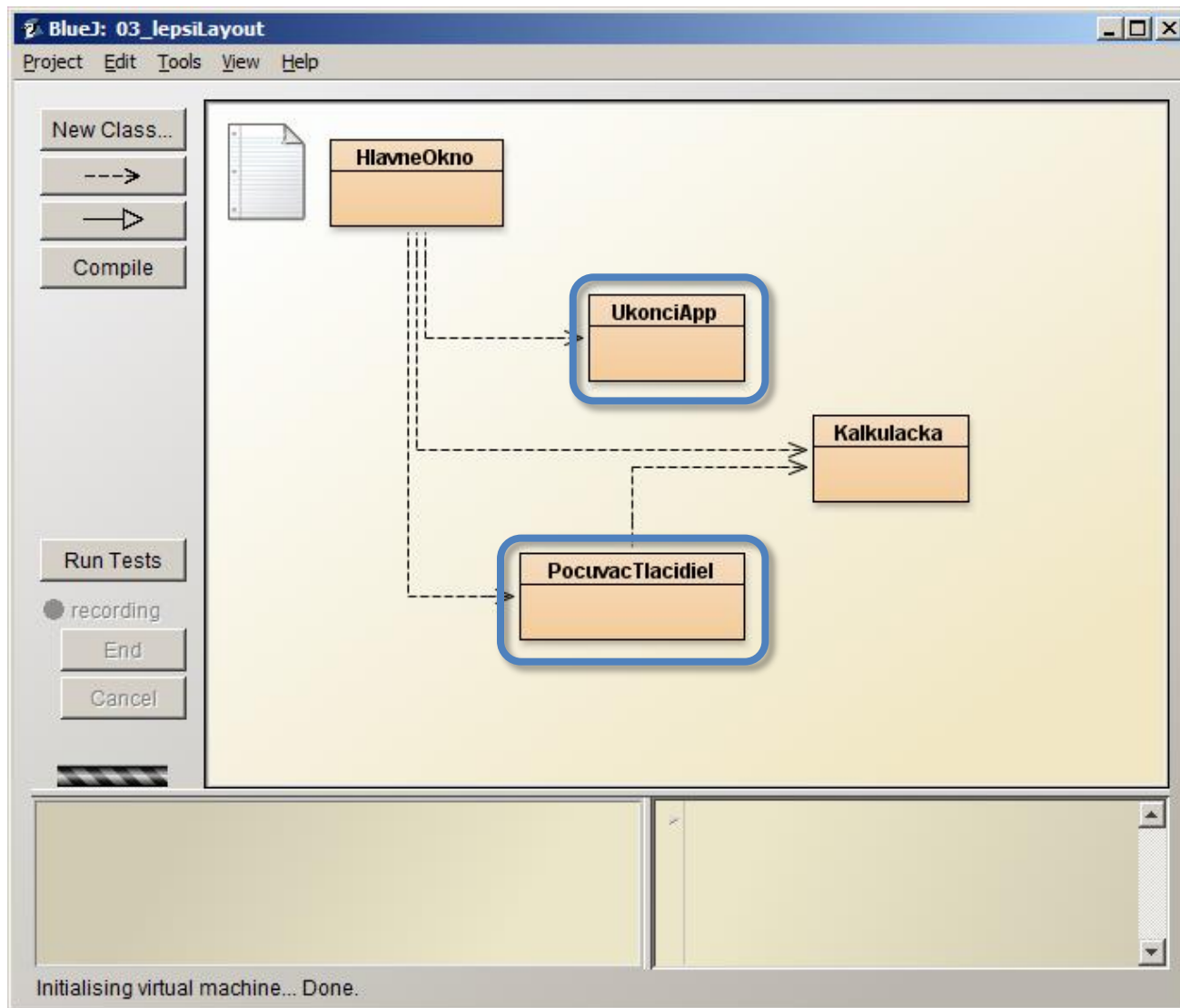


- WindowListener – 7 správ
- aby nebolo nutné ich implementovať
 - abstraktná trieda WindowAdapter
 - implementácia interface WindowListener
 - všetky metódy prázdne

PosluchacUkonci

```
public class PosluchacUkonci
    extends WindowAdapter
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
}
```

Projekt v BlueJ



Vnorená trieda

- s poslucháčmi pracuje len okno
- triedy nemusia byť prístupné mimo triedy okna
- vnorená trieda
 - definovaná ako časť inej triedy (vonkajšia)
 - modifikátory private, protected, public, nič
 - priami prístup ku atribútom vonkajšej triedy
 - vytvoriť inštanciu je možné len z inštancie vonkajšej alebo vnútornej triedy

PosluchacTlacidiel ako vnorená trieda

```
private char aOperacia;
```

```
public PosluchacTlacidiel(char paOperacia) {  
    aOperacia = paOperacia;  
}
```

```
public void actionPerformed(ActionEvent e) {  
    aKalkulacka.stlac(aOperacia);  
    aDisplej.setText(aKalkulacka.dajDisplej());  
}
```

Anonymná trieda

- zjednodušenie vnorenej triedy
- nedostáva identifikátor
- odvodená od inej triedy, alebo implementuje interface

Anonymná trieda ako potomok₍₁₎

```
private class AnonymnaTrieda extends Predok
{
    // definicie metod
}
...
premenna = new AnonymnaTrieda();
```

Anonymná trieda ako potomok₍₂₎

```
premenna = new Predok() {  
    // definície metod  
}
```

Anonymná trieda ako implementácia₍₁₎

```
private class AnonymnaTrieda implements Interface
{
    // definície metod
}
...
premenna = new AnonymnaTrieda();
```

Anonymná trieda ako implementácia₍₂₎

```
premenna = new Interface() {  
    // definície metód  
}
```

Príklad – ukončenie aplikácie

```
aOkno.addWindowListener(  
    new WindowAdapter ()  
    {  
        public void windowClosing(WindowEvent e)  
        {  
            System.exit(0);  
        }  
    }  
);
```

Uzávery (closure)

- nutnosť dodatočných informácií
 - operácia kalkulačky
- možnosť použitia lokálnych premenných z nadradeného menného priestoru v anonymnej triede
 - lokálna premenná musí byť označená ako final

Príklad – poslucháč tlačidiel₍₁₎

```
private void pridajTlacidlo(String paText)
{
    final char operacia = paText.charAt(0);
    JButton tlacidlo = new JButton(paText);
    ...
    aOkno.add(tlacidlo);
}
```

Príklad – poslucháč tlačidiel₍₂₎

```
tlacidlo.addActionListener(  
    new ActionListener() {  
        public void actionPerformed(ActionEvent e)  
        {  
            aKalkulacka.stlac(operacia);  
            aDisplej.setText(aKalkulacka.dajDisplej());  
        }  
    }  
);
```


Vzhľad SWING aplikácií

- možnosť meniť tému
- štandardne téma Metal
- možnosť nastavenia na System
 - na windows sa použije Windows téma
 - na linuxe GTK+ téma
 - na MacOS sa použije Macintosh téma
 - ...
- trieda `javax.swing.UIManager`

Nastavenie témy System

```
try {  
    UIManager.setLookAndFeel(  
        UIManager.getSystemLookAndFeelClassName());  
} catch (Exception ex) {  
    // nepodarilo sa načítať tému  
}
```

Vďaka za pozornosť