

Informatika 2

Grafické používateľské rozhranie



Pojmy zavedené v 7. prednáške (1)

- štandardný vstup a výstup
- textové súbory
- binárne súbory
- objektové prúdy

Pojmy zavedené v 7. prednáške (2)

- štandardný vstup a výstup
- čítanie: objekt System.in a Scanner
- zapisovanie: objekt PrintStream
 - System.out
 - System.err
- členenie na riadky

Pojmy zavedené v 7. prednáške (3)

- súbory
- trieda File
- spolupráca so súborovým systémom
- práca so súborom
 - otvorenie – vytvorenie príslušnej inštancie
 - čítanie alebo zapisovanie
 - zatvorenie – správa close()

Pojmy zavedené v 7. prednáške (4)

- textové súbory
 - čítateľné pre človeka
- čítanie: Scanner (+ Scanner)+ File
- zápisovanie: PrintWriter (+ File)
- formátovanie textového súboru
- bežne členenie na riadky

Pojmy zavedené v 7. prednáške (5)

- binárne súbory
 - nečitateľné pre človeka
- čítanie: `DataInputStream` + `FileInputStream` (+ `File`)
- zapisovanie: `DataOutputStream` + `FileOutputStream` (+ `File`)
- `String` a primitívne dátové typy

Pojmy zavedené v 7. prednáške (6)

- objektové prúdy
 - nečitateľné pre človeka
- čítanie: `ObjectInputStream` + `FileInputStream` (+ `File`)
- zapisovanie: `ObjectOutputStream` + `FileOutputStream` (+ `File`)
- celé objekty

Cieľ prednášky

- GUI
- vnorená trieda
- anonymná trieda

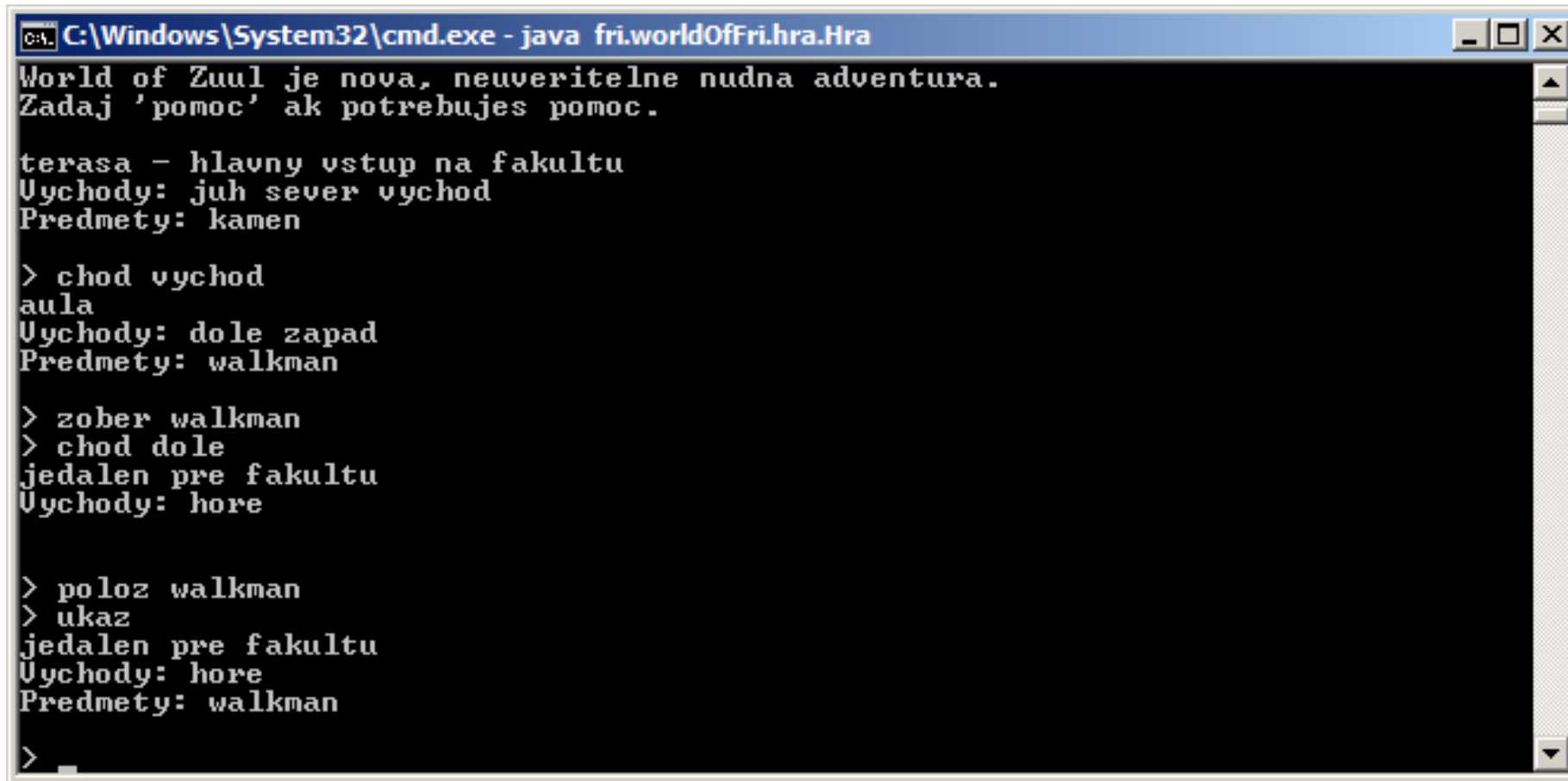
- príklad: kalkulačka

Možnosti riešenia používateľskej vrstvy

- CLI – Command Line Interface
- TUI – Text User Interface
- GUI – Graphical User Interface
- ...
- Web UI/PWA
- Mobile UI
- ...

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

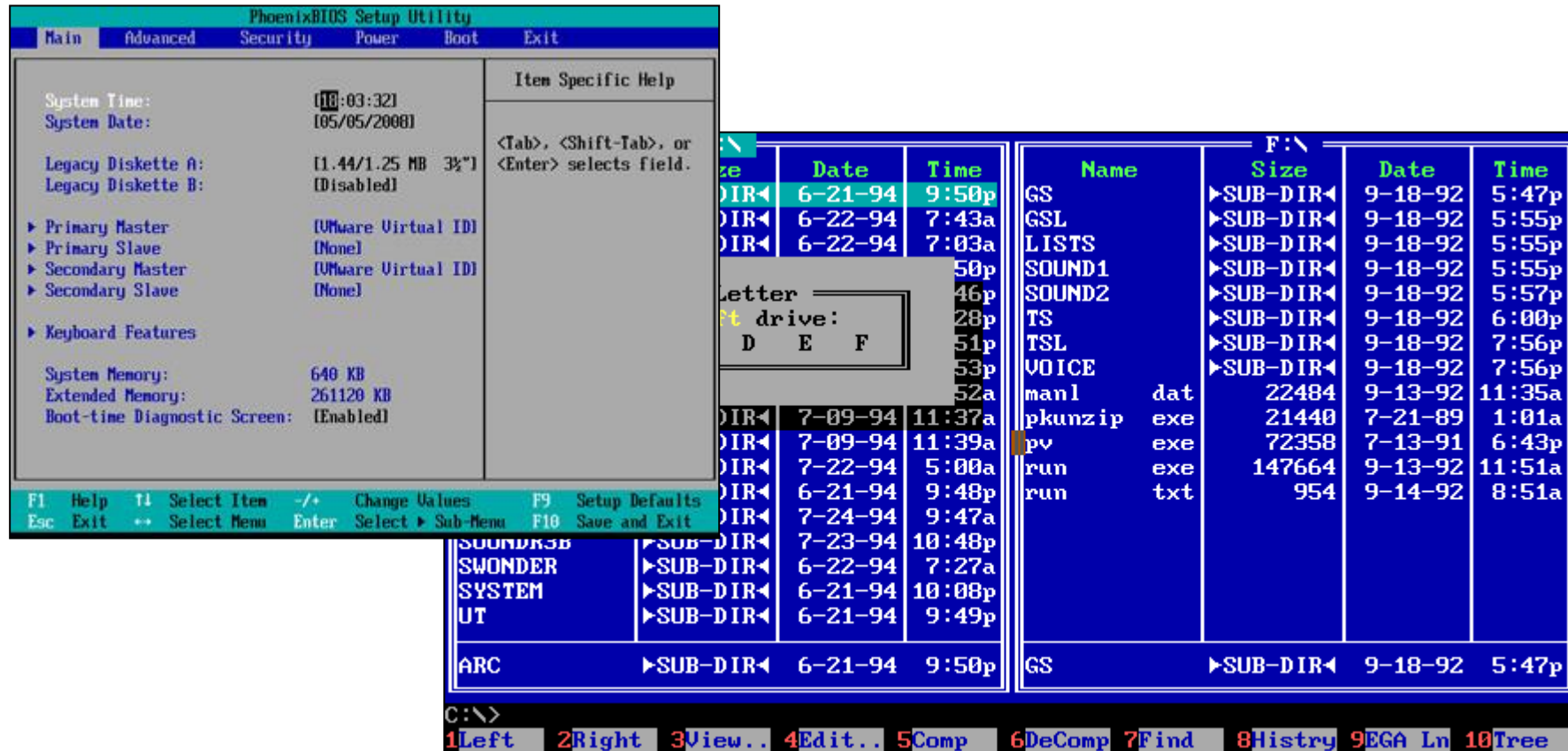
> zober walkman
> chod dole
jedalen pre fakultu
Uychody: hore

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

>
```

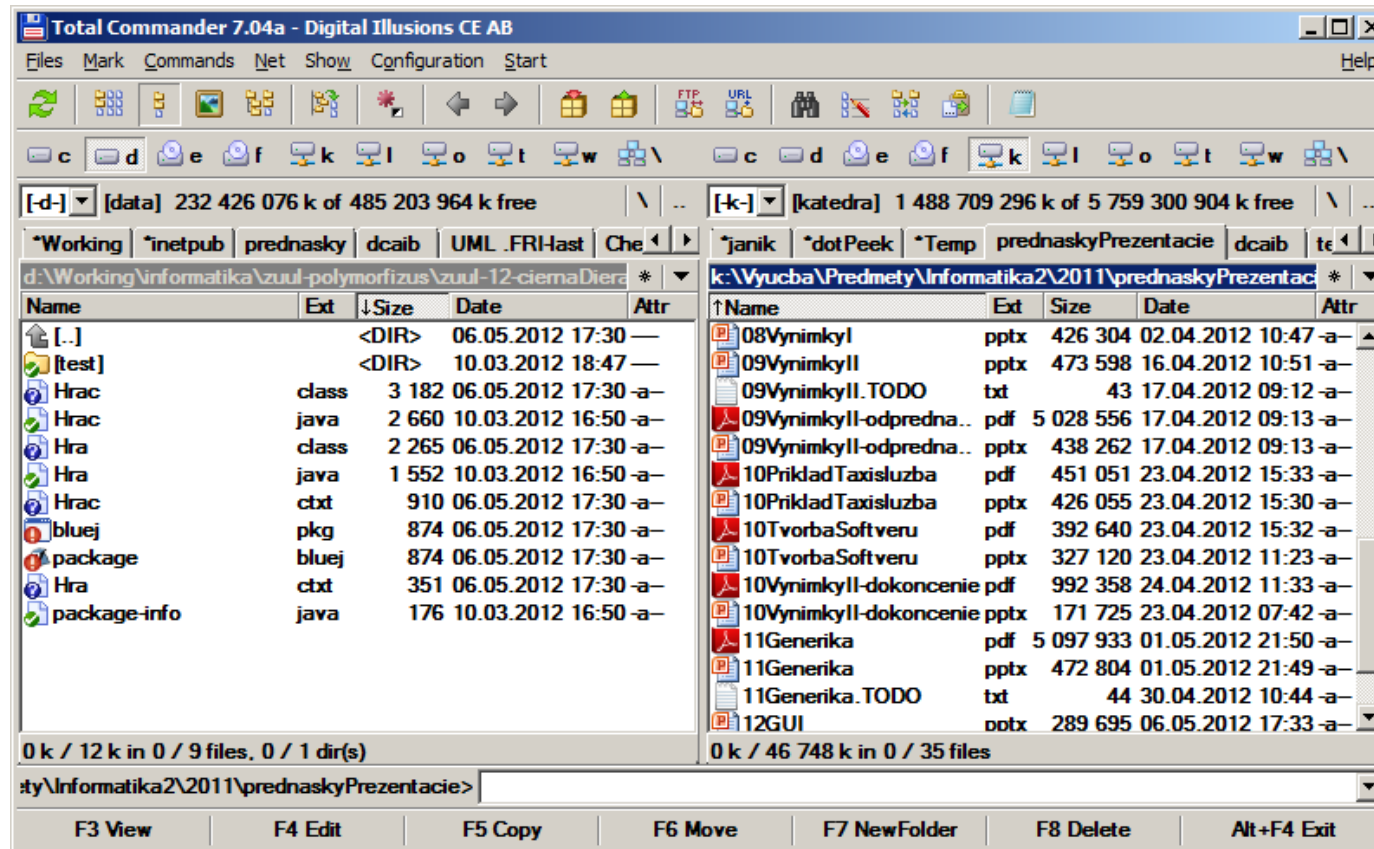
Text user interface

- BIOS setup, rôzne programy pre DOS



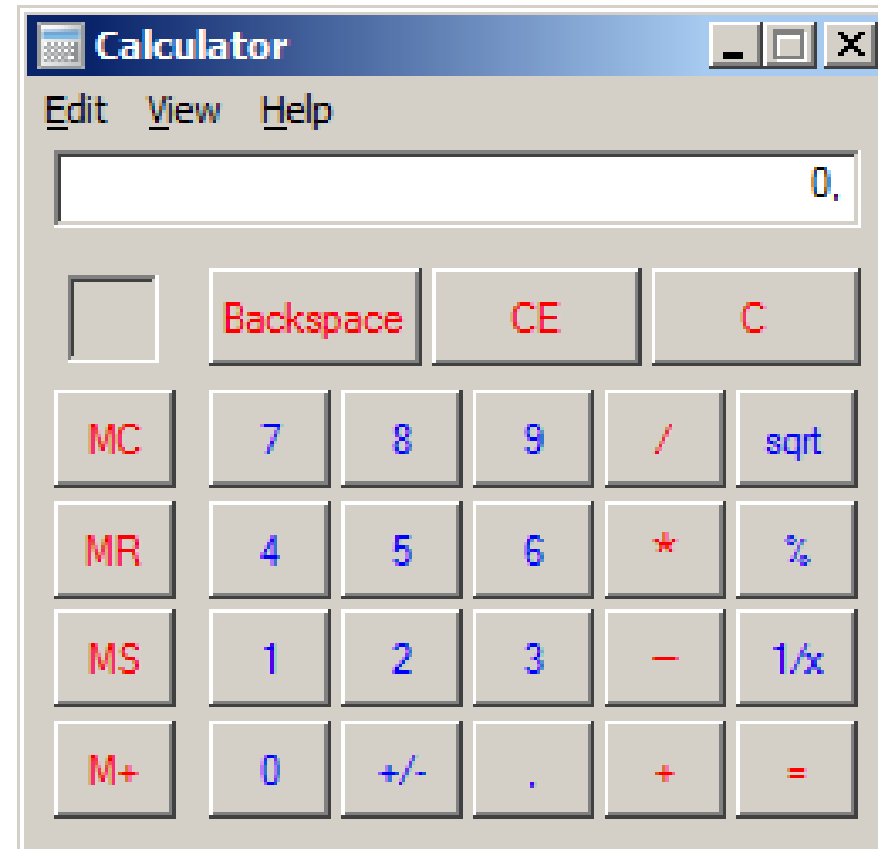
Graphical user interface

- softvér pre Windows, Linux/X11, MacOS X



Zadanie – kalkulačka s GUI

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



Logika kalkulačky

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

Kalkulacka
+ <u>new(): Kalkulacka</u> + stlac(znak: char): void + getDisplej(): String

GUI v jazyku Java

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

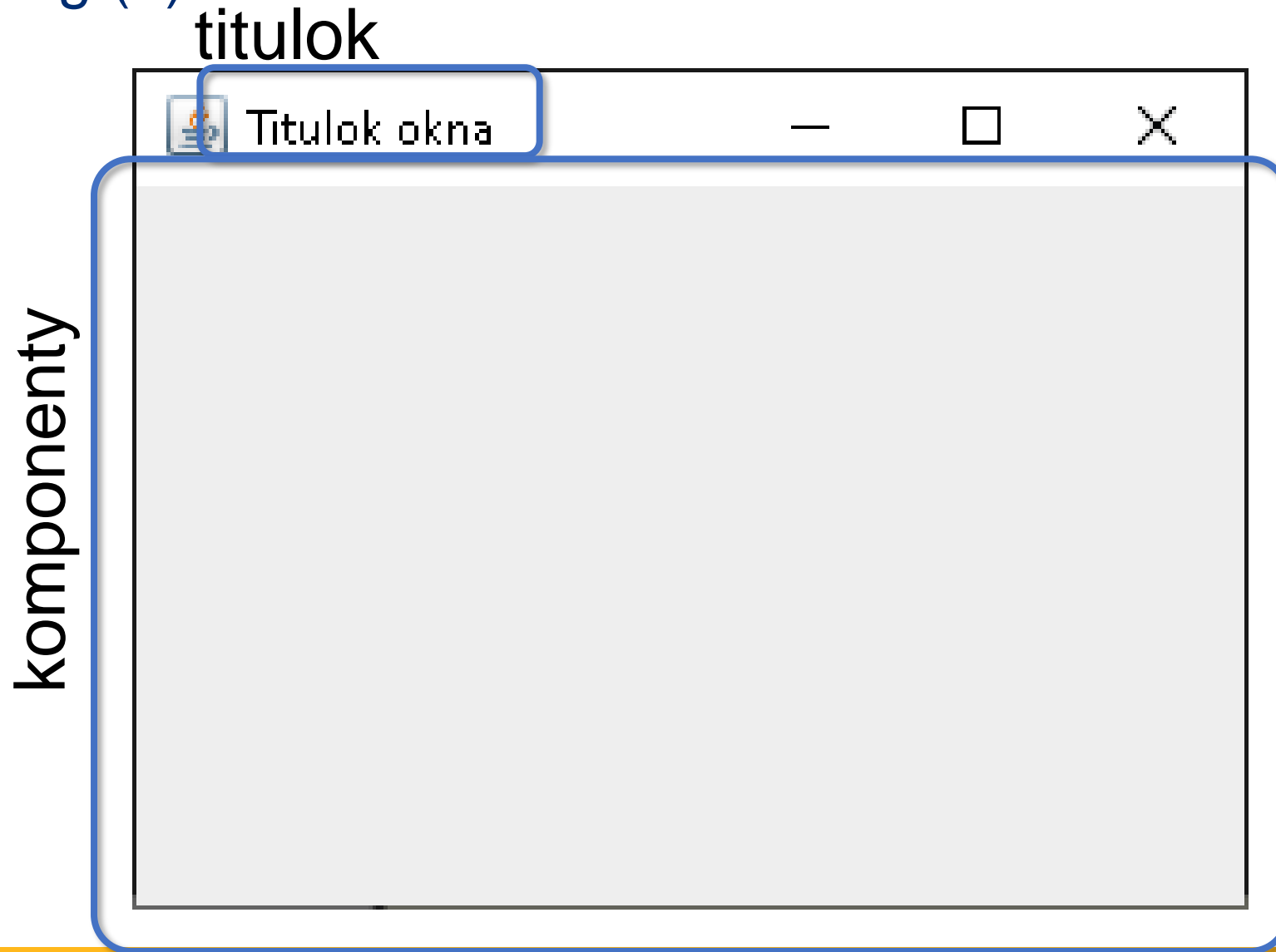
Okno v Swing (1)

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

JFrame

- + new(): JFrame
- + new(titulok: String): JFrame
- + add(komponent: Component): void
- + setVisible(viditelnost: boolean): void
- + pack(): void

Okno v Swing (2)

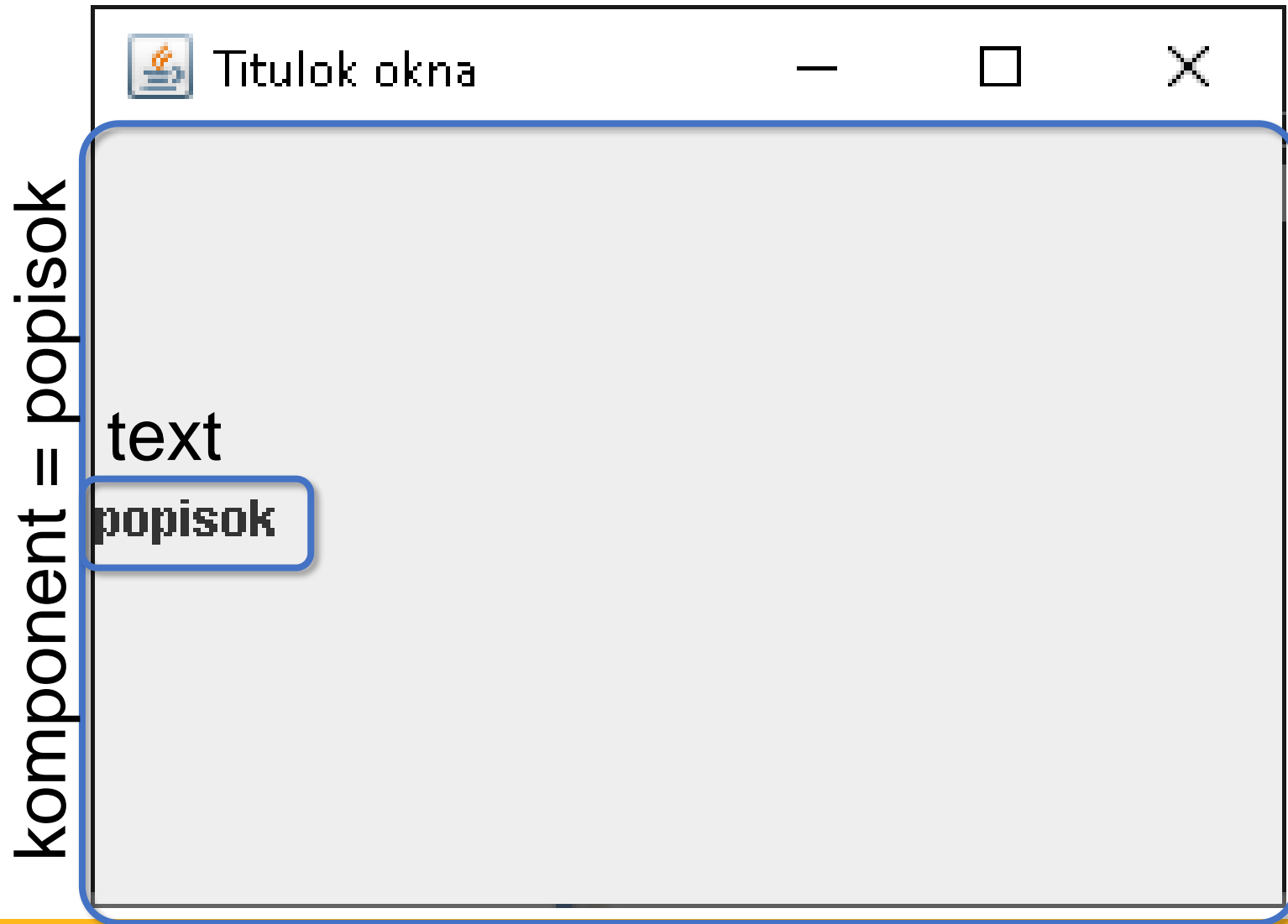


Popisok (1)

- javax.swing.JLabel = popisok v okne

JLabel
+ <u>new(popisok: String): JLabel</u> + getText(): String + setText(popisok: String): void

Popisok (2)



Tlačidlo (1)

- javax.swing.JButton = tlačidlo v okne

JButton

- + new(popisok: String): JButton
- + getText(): String
- + setText(popisok: String): void

Tlačidlo (2)



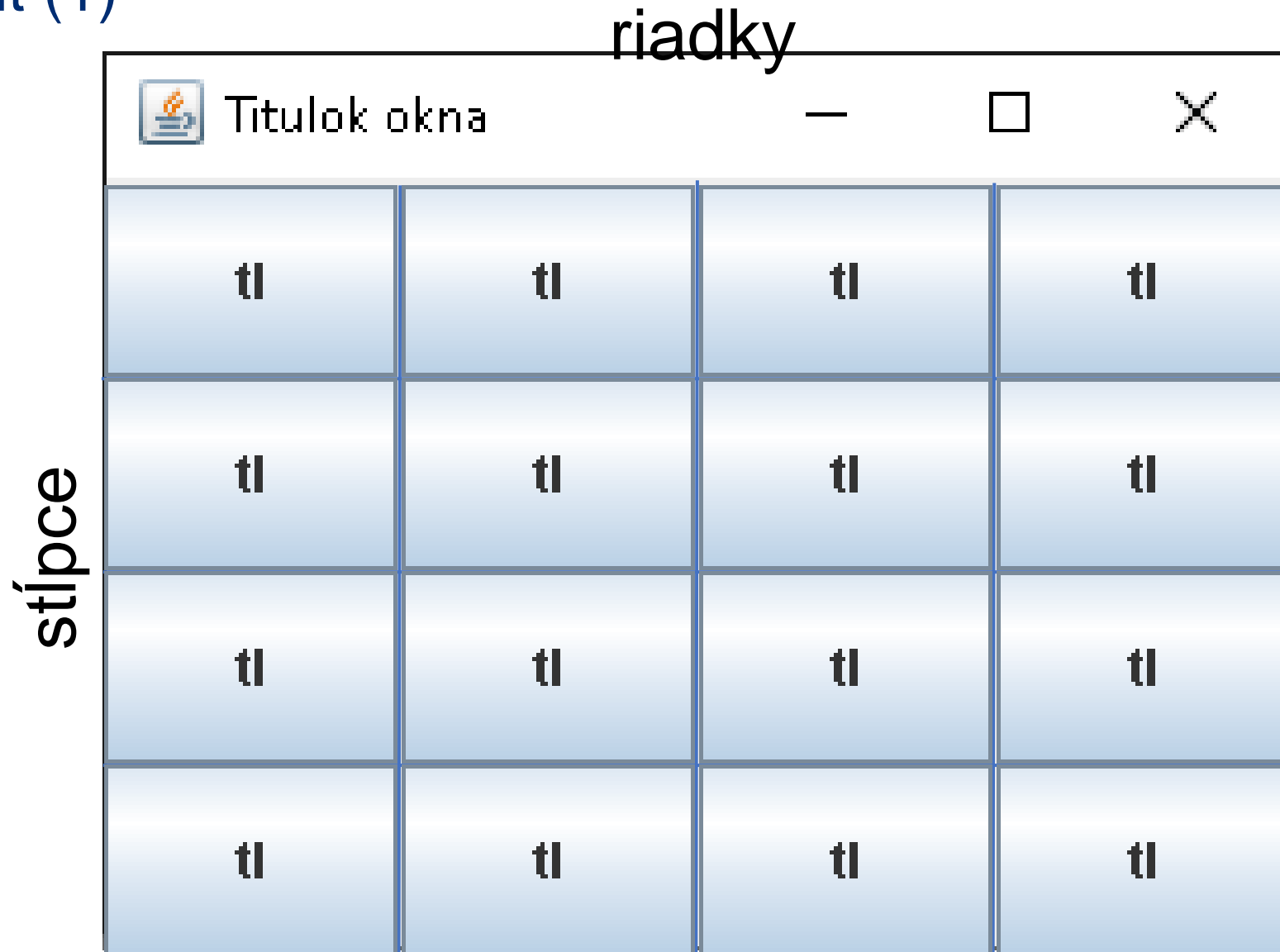
Kalkulačka

- 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)

```
var okno = new JFrame("Titulok okna");  
okno.setLayout(new GridLayout(4, 4));  
  
okno.add(new JButton("t1"));  
okno.add(new JButton("t1"));  
okno.add(new JButton("t1"));  
okno.add(new JButton("t1"));  
okno.add(new JButton("t1"));  
  
...
```

Kalkulačka pomocou GridLayout

 Titulok okna — □ ×			
displej	???	???	???
7	8	9	+
4	5	6	-
1	2	3	*
.	0	=	/

Implementácia kalkulačky (1)

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

Implementácia kalkulačky (2)

```
var 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[] parametre) {  
        var 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
 - ...

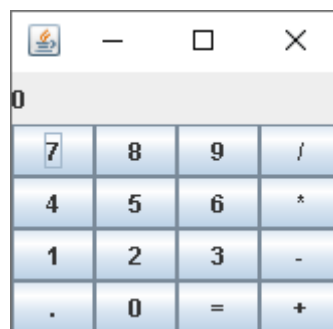
Implementácia HlavneOkno (1)

```
public class HlavneOkno {  
    private JFrame okno;  
  
    . . .  
}
```

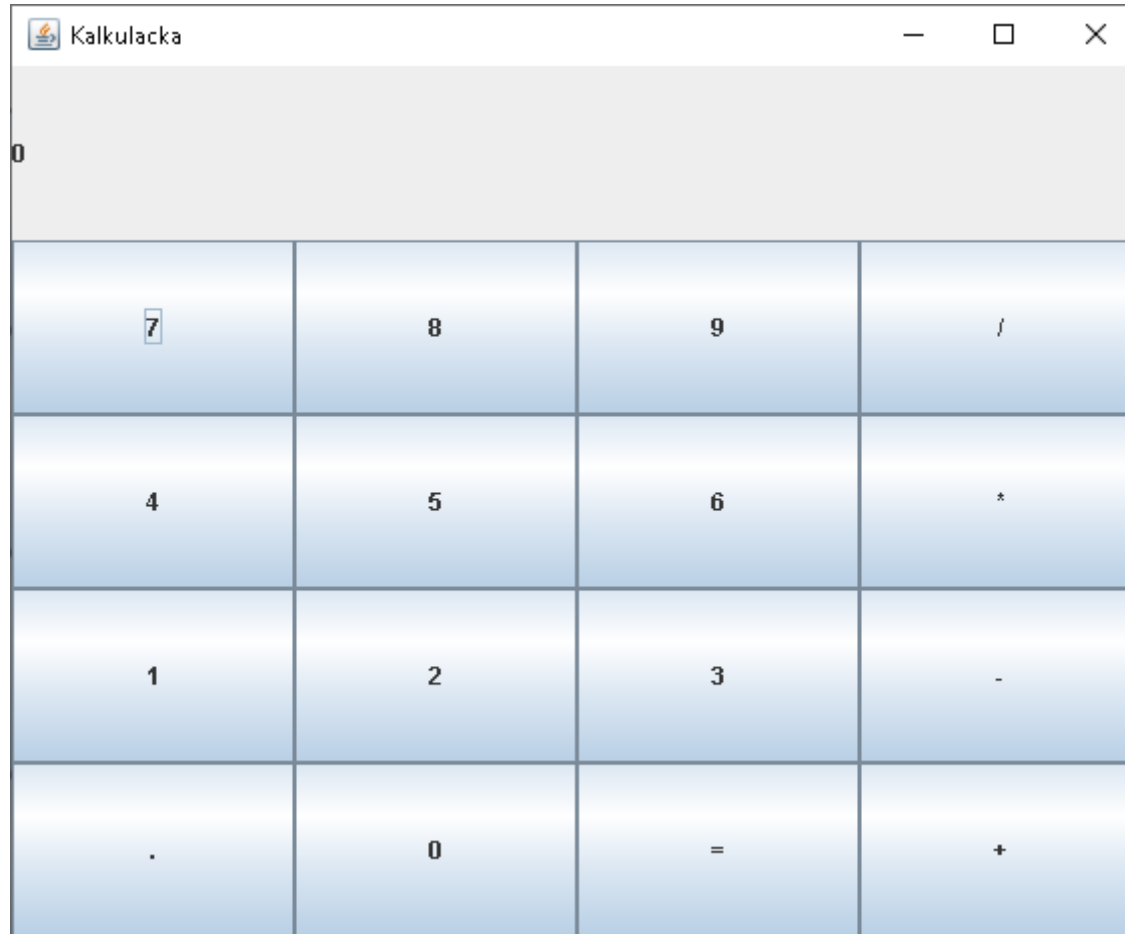
Implementácia HlavneOkno (2)

```
public HlavneOkno() {  
    this.okno = new JFrame("Kalkulacka");  
    ...  
}  
  
public void zobraz() {  
    this.okno.setVisible(true);  
}
```

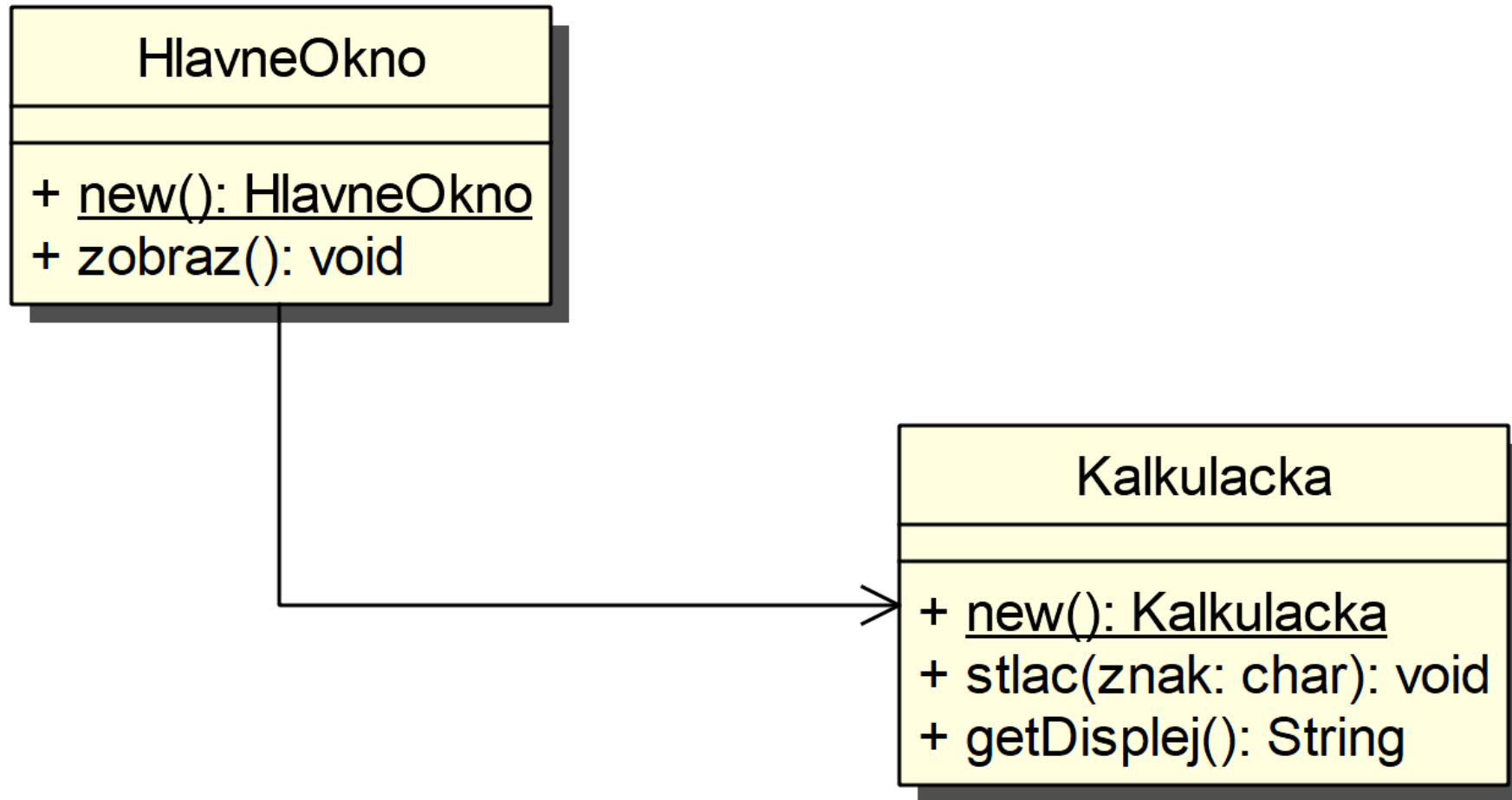
Výsledok



Zväčšovanie okna kalkulačky



Spolupráca okna s kalkulačkou



Udalosti GUI

- 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, ...
 - ...

Poslucháč tlačidiel

- nastane po kliknutí???
 - `actionPerformed`
- akcia:
 - pošli kalkulačke správu `stlac(operacia)`
 - aktualizuj hodnotu na displeji
- vstupy:
 - objekt kalkulačka
 - komponent displeja
 - operácia, ktorá sa má vykonať

PosluchacTlacidiel (1)

```
public class PosluchacTlacidiel implements ActionListener {  
    private Kalkulacka kalkulacka;  
    private JLabel displej;  
    private char operacia;  
  
    ...  
}
```

PosluchacTlacidiel (2)

```
public PosluchacTlacidiel(char operacia,  
                           JLabel displej, Kalkulacka kalkulacka) {  
    this.operacia = operacia;  
    this.displej = displej;  
    this.kalkulacka = kalkulacka;  
}
```

PosluchacTlacidiel (3)

```
public void actionPerformed(ActionEvent e) {  
    this.kalkulacka.stlac(this.operacia);  
    this.displej.setText(this.kalkulacka.getDisplej());  
}
```

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 okno;  
    private Kalkulacka kalkulacka;  
    private JLabel displej;  
  
    ...  
}
```

Implementácia HlavneOkno (2)

```
public HlavneOkno() {  
    this.okno = new JFrame("Kalkulacka");  
    this.displej = new JLabel("0");  
    this.kalkulacka = new Kalkulacka();  
  
    ...  
}
```

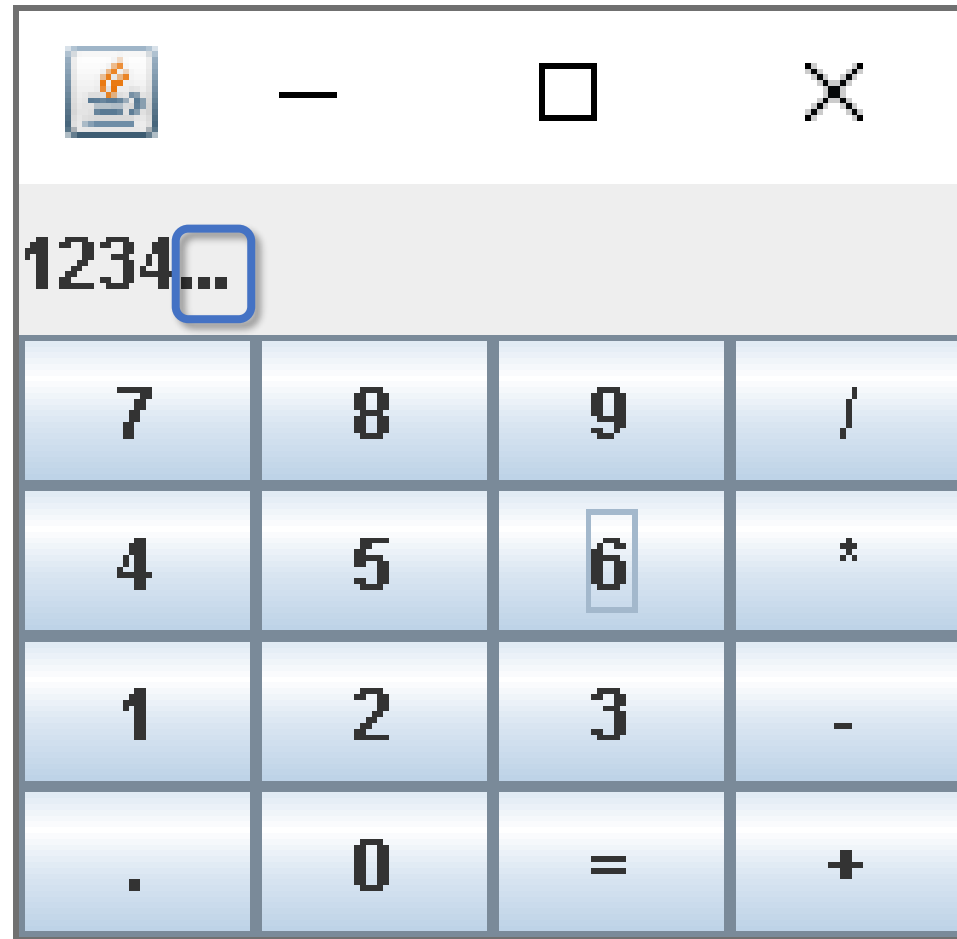
Implementácia HlavneOkno (3)

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

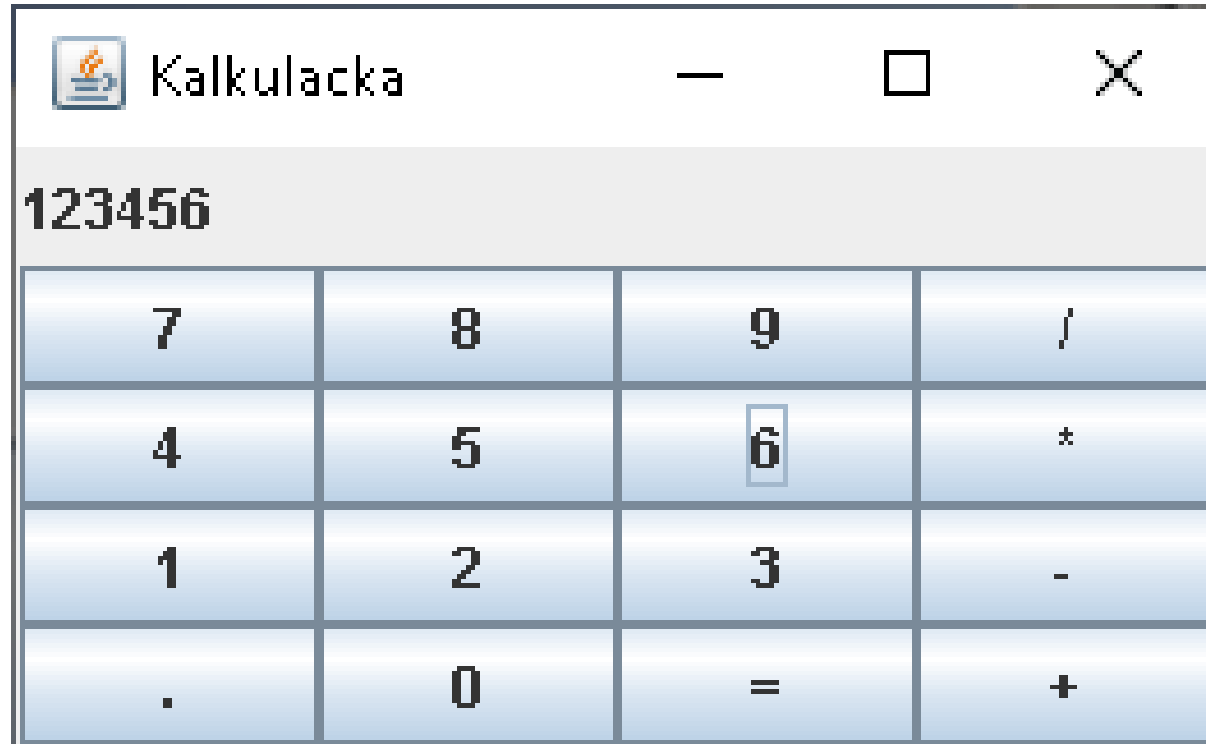
Implementácia HlavneOkno (4)

```
var tlacidlo = new JButton("7");  
tlacidlo.addActionListener(  
    new PosluchacTlacidiel(  
        '7', this.displej, this.kalkulacka  
    ));  
this.okno.add(tlacidlo);
```



Problém s displejom(1)



Problém s displejom (2)



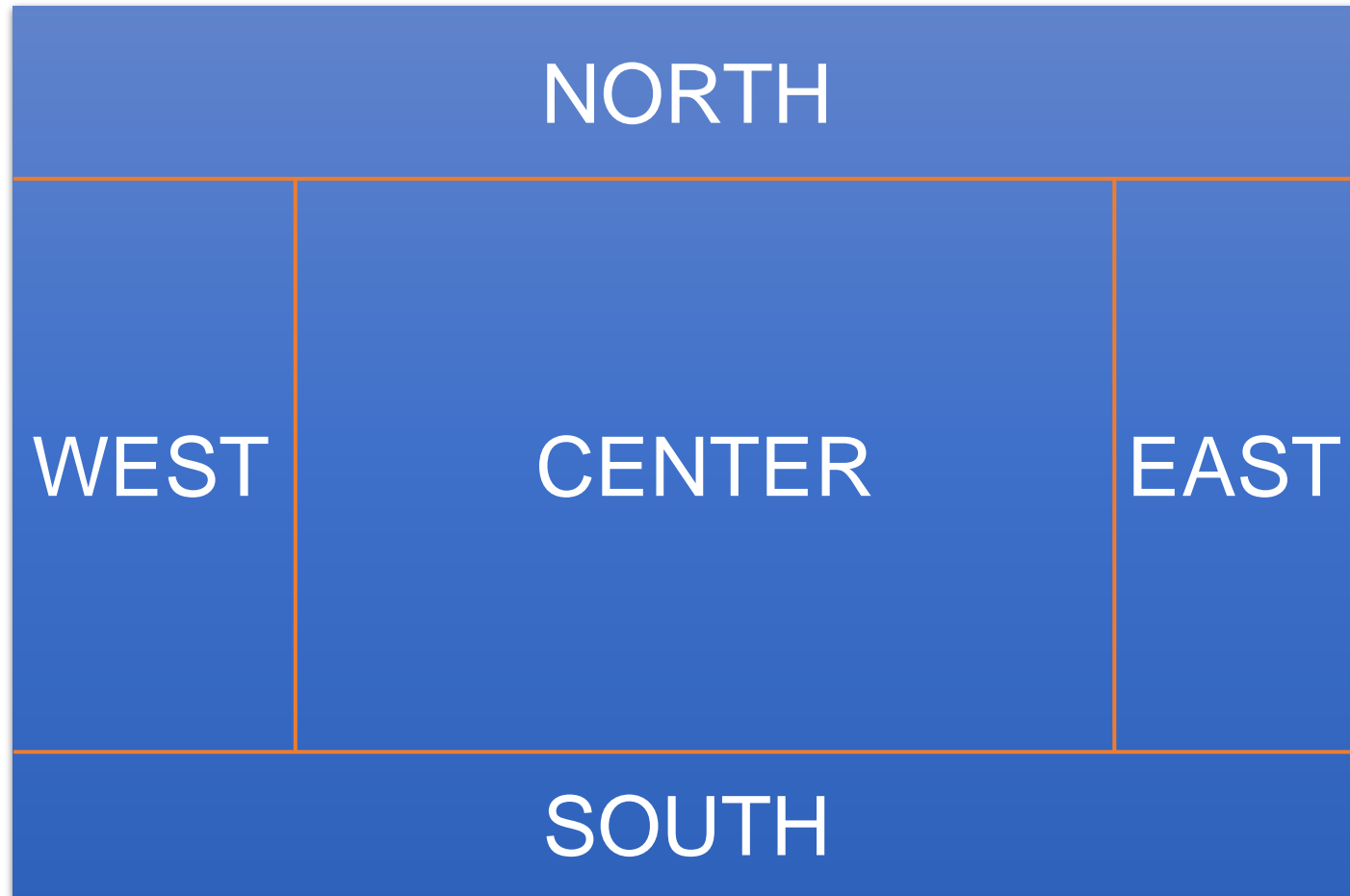
Problém objasnený

	-	\square	\times
1234...			
7	8	9	/
4	5	6	*
1	2	3	-
.	0	=	+

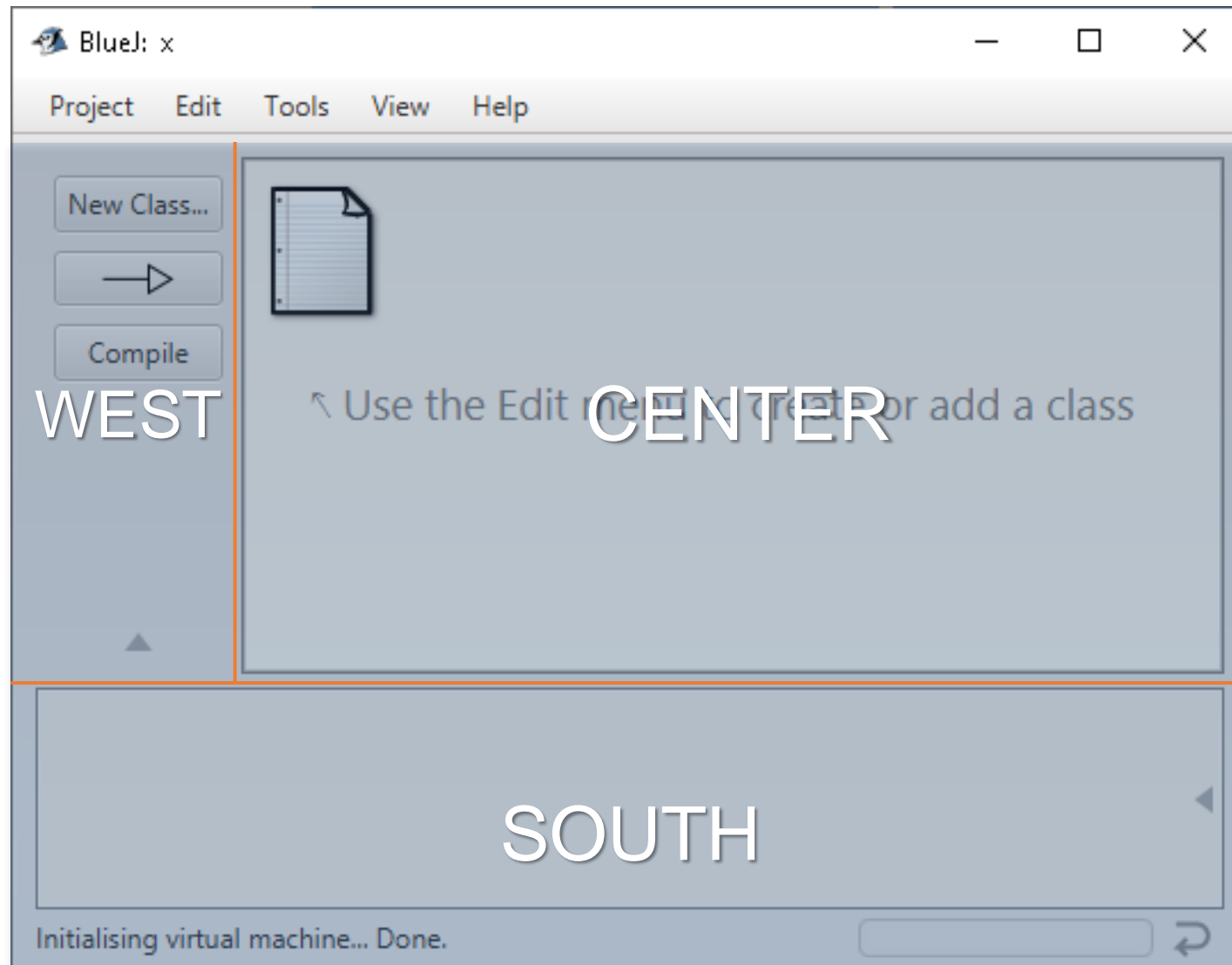
GUI komponenty – kontajnery

- 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

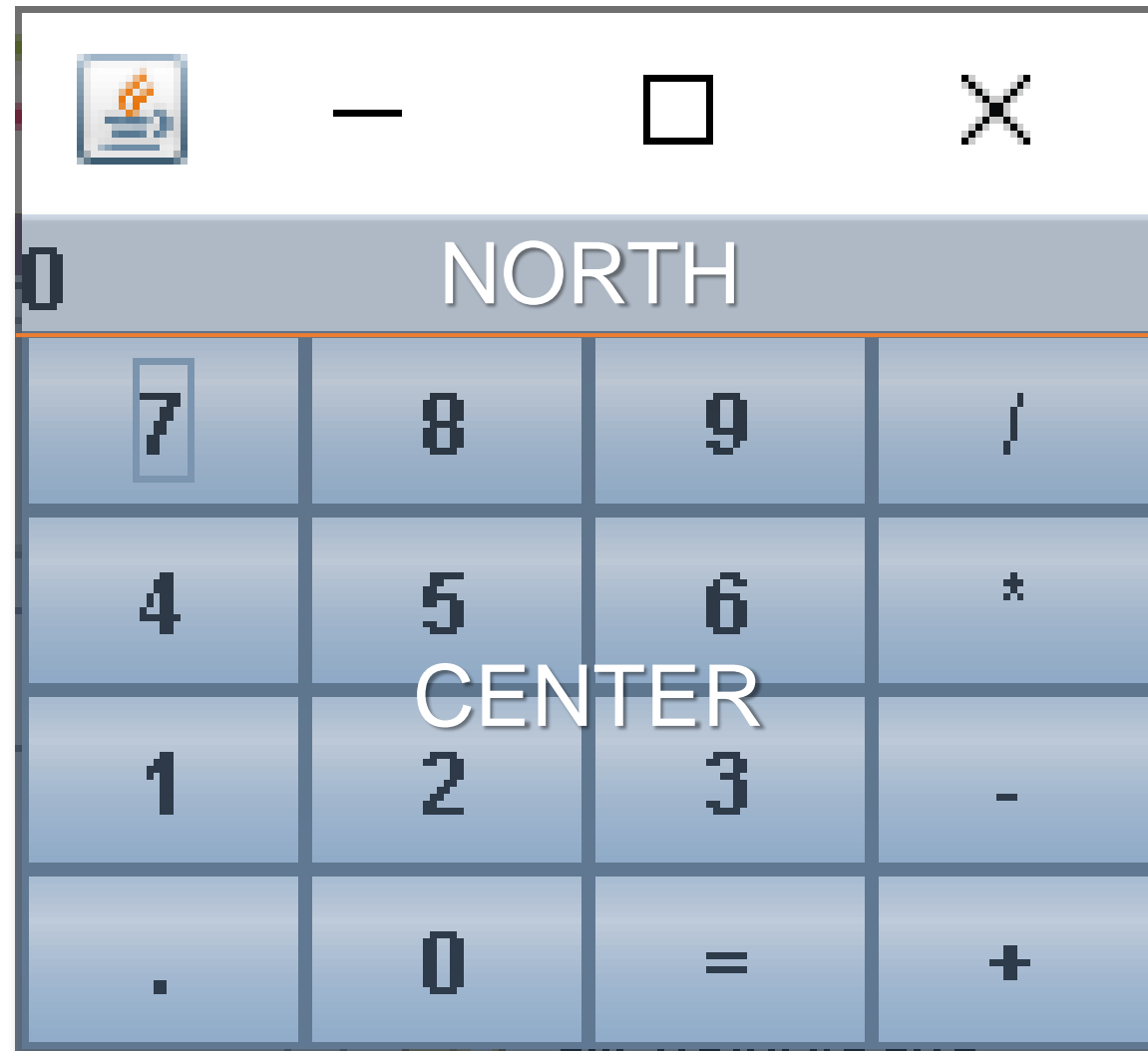
BorderLayout



BorderLayout v BlueJ



BorderLayout v Kalkulačke



Kombinácia viac Layout

```
this.okno.setLayout(new BorderLayout());  
  
this.displej = new JLabel("0");  
this.okno.add(this.displej, BorderLayout.NORTH);  
  
this.klavesnica = new JPanel();  
this.okno.add(this.klavesnica, BorderLayout.CENTER);  
  
this.klavesnica.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

«interface»
WindowListener

+ windowClosing(e: WindowEvent): void
+ windowIconified(e: WindowEvent): void
...

Adaptéry

- 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);  
    }  
}
```

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
 - priamy prístup ku atribútom vonkajšej triedy (NazovTriedy.this.atribut)
 - vytvoriť inštanciu je možné len z inštalácie vonkajšej alebo vnútornej triedy

PosluchacTlacidiel ako vnorená trieda (1)

```
public class HlavneOkno {  
    private JFrame okno;  
    private Kalkulacka kalkulacka;  
    private JLabel displej;  
  
    private class PosluchacTlacidiel implements ActionListener {  
        ...  
    }  
    ...  
}
```

PosluchacTlacidiel ako vnorená trieda (2)

```
private char operacia;

public PosluchacTlacidiel(char operacia) {
    this.operacia = operacia;
}

public void actionPerformed(ActionEvent e) {
    HlavneOkno.this.kalkulacka.stlac(this.operacia);
    HlavneOkno.this.displej.setText(
        HlavneOkno.this.kalkulacka.getDisplej());
}
```

Anonymná trieda

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

Anonymná trieda ako potomok (1)

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

Anonymná trieda ako potomok (2)

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

Anonymná trieda ako implementácia (1)

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

Anonymná trieda ako implementácia (2)

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

Príklad – ukončenie aplikácie

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

Dá sa aj jednoduchšie 😊

- default close operation
- operácia po zavretí okna
 - nič
 - ukončenie aplikácie
 - zrušenie okna
 - skrytie okna (implicitne)

```
this.setDefaultCloseOperation(  
                                WindowConstants.EXIT_ON_CLOSE) ;
```

Uzávery (closure)

- nutnosť dodatočných informácií
 - operácia kalkulačky
- možnosť použitia lokálnych premenných z nadradeného menného priestoru v anonymnej triede

Príklad – poslucháč tlačidiel (1)

```
private void pridajTlacidlo(String text) {  
    var tlacidlo = new JButton(text);  
  
    ...  
  
    this.okno.add(tlacidlo);  
}
```


Príklad – poslucháč tlačidiel (2)

```
var operacia = text.charAt(0);  
tlacidlo.addActionListener(  
    new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            HlavneOkno.this.kalkulacka.stlac(operacia);  
            HlavneOkno.this.displej.setText(  
                HlavneOkno.this.kalkulacka.getDisplej());  
        }  
    }  
);
```

Lambda funkcia

- zjednodušená syntax
- anonymná trieda implementujúca interface s jednou správou
- možnosť priameho použitia this

Príklad – poslucháč tlačidiel

```
private void pridajTlacidlo(String text) {  
    var tlacidlo = new JButton(text);  
    char operacia = text.charAt(0);  
    tlacidlo.addActionListener((ActionEvent e) -> {  
        this.kalkulacka.stlac(operacia);  
        this.displej.setText(this.kalkulacka.getDisplej());  
    });  
    this.okno.add(tlacidlo);  
}
```

Ešte zjednodušená lambda

```
private void pridajTlacidlo(String text) {  
    var tlacidlo = new JButton(text);  
    var operacia = text.charAt(0);  
    tlacidlo.addActionListener(e -> {  
        this.kalkulacka.stlac(operacia);  
        this.displej.setText(this.kalkulacka.getDisplej());  
    });  
    this.okno.add(tlacidlo);  
}
```

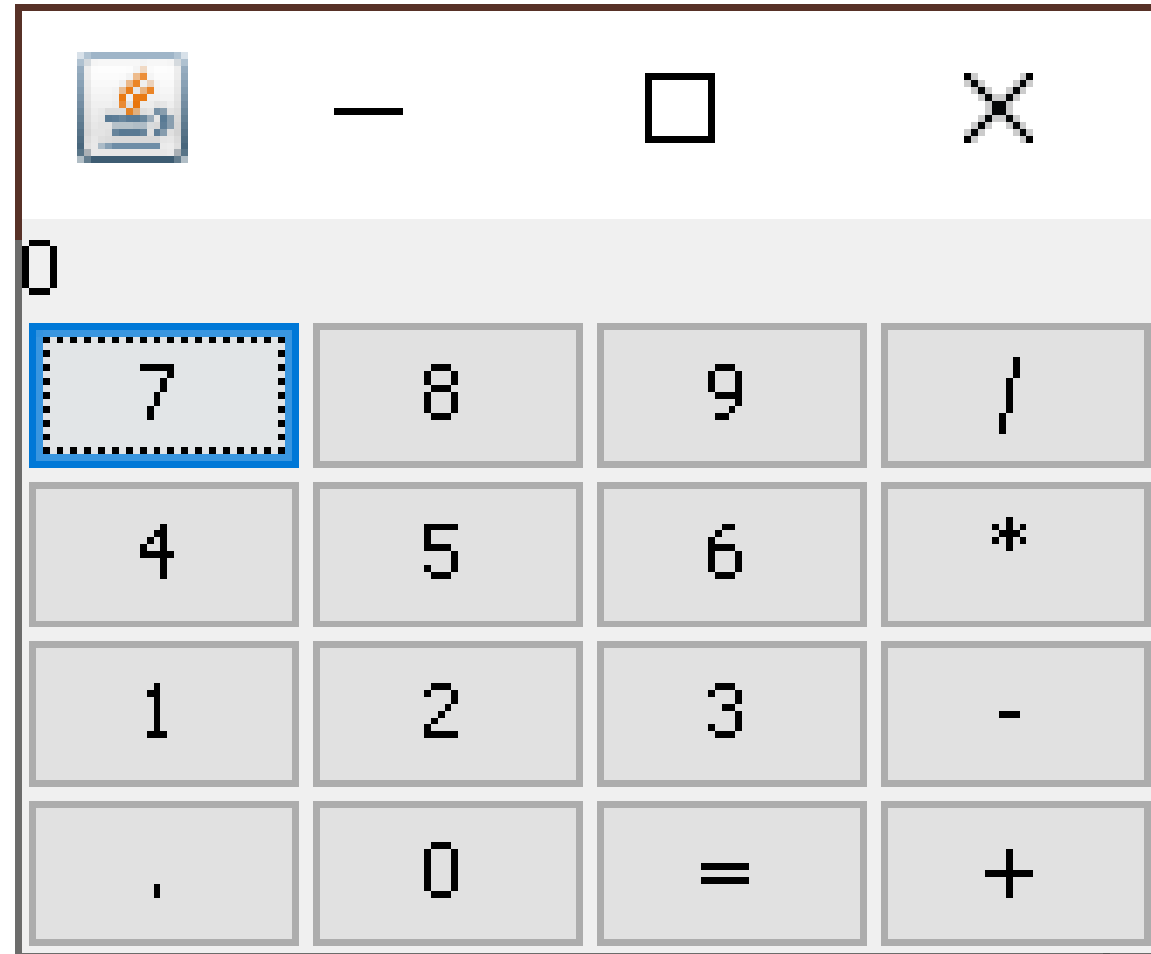
Vzhľad SWING aplikácií

- 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  
}
```

Kalkulačka s témou System (Windows)



Kalkulačka s témou System (Linux)

