

Programowanie w Ruby

Wykład 9

Marcin Młotkowski

5 grudnia 2012

Plan wykładu

Proste środowisko graficzne

Menu

Obrazki

Wątki w aplikacjach graficznych Tkinter

Środowisko okienkowe Tcl/Tk

- ▶ Tcl: Tool Command Language
- ▶ Tk: Tk Toolkit, przenośna biblioteka graficzna

Przykład

Kalkulator graficzny

- ▶ Okienko do wprowadzania danych
- ▶ Przycisk <Oblicz>
- ▶ Miejsce na wynik
- ▶ Przycisk <Koniec>

Implementacja

```
require 'tk'
```

```
class Okienko
```

```
  def run
```

```
    @win = TkRoot.new { title 'Kalkulator' }
```

Implementacja

```
require 'tk'
```

```
class Okienko
```

```
  def run
```

```
    @win = TkRoot.new { title 'Kalkulator' }
```

```
    @entry = TkEntry.new(@win) { pack }
```

```
    @butt = TkButton.new(@win) { text 'Oblicz'; pack }
```

```
    @butt.command { self.oblicz }
```

```
    @label = TkLabel.new(@win) { text '0'; pack }
```

```
    TkButton.new(@win) { text 'KONIEC';  
                        command { exit }; pack }
```

Implementacja

require 'tk'

class Okienko

def run

```
@win = TkRoot.new { title 'Kalkulator' }
```

```
@entry = TkEntry.new(@win) { pack }
```

```
@butt = TkButton.new(@win) { text 'Oblicz'; pack }
```

```
@butt.command { self.oblicz }
```

```
@label = TkLabel.new(@win) { text '0'; pack }
```

```
TkButton.new(@win) { text 'KONIEC';  
                    command { exit }; pack }
```

```
Tk.mainloop
```

end

end

Implementacja obsługi zdarzeń

```
butt.command { oblicz }  
def oblicz  
  @label.text = eval(@entry.value)  
  @entry.value = "  
end
```

Wynik

Oblicz

0

KONIEC

Opis obrazka



Układanie kontroltek

Model grid

- ▶ Okno jest dzielone na komórki ułożone w wiersze i kolumny
- ▶ Kontrolki są wstawiane w komórki
- ▶ kontrolki mogą zajmować więcej niż jedną komórkę

Układanie kontrolek w komórkach siatki

```
@entry = TkEntry.new(@win)
        { grid(:row => 1, :column => 1) }

butt = TkButton.new(@root)
        { grid(:row => 1, :column => 2) }

@label = TkLabel.new(@root)
        { grid('row' => 1, 'column' => 3) }

TkButton.new(@root) { grid(:row => 2, :column => 2) }
```

Rezultat

	Oblicz	8388608
	KONIEC	

Pierwsze menu

```
menu = TkMenu.new()
one = TkMenu.new(menu)
menu.add('cascade', :menu => one, :label => 'Plik')
one.add('command', :label => 'Zapisz',
        :command => proc { puts "Zapisane" })
one.add('command', :label => 'KONIEC',
        :command => proc { root.destroy })
```

Drugie menu

```
two = TkMenu.new(menu)
menu.add('cascade', :menu => two, :label => 'Edycja')
two.add('command', :label => 'Skopiuj', :command => proc { })
root.menu(menu)
```

Wynik

Plik

Edycja

Kontrolka TkCanvas

```
cv = TkCanvas.new(root)
```

Kontrolka TkCanvas

```
cv = TkCanvas.new(root)
```

Ustalenie rozmiarów

```
cv.place(:height => 200, :width => 200)
```

Rysowanie na płótnie

Na płótnie umieszcza się elementy Tk^{c*}, na przykład TkArc, TkLine czy TkBitmap.

Przykład

```
cv.create(TkcOval, 50, 50, 150, 150, 'width' => 1)
cv.create(TkcLine, 100, 80, 100, 110)
cv.create(TkcArc, 80, 110, 120, 130,
          :start => 180, :extent => 180)
cv.create(TkcOval, 70, 70, 90, 100, :fill => 'black')
cv.create(TkcOval, 110, 70, 130, 100, :fill => 'black')
TkText.new(cv, 100, 170, :font => 'Arial 12 bold',
           :text => 'Hello, world!', :anchor => 'center')
```

Wynik

Wynik

Plik Edycja



Hello, world!

Pliki graficzne

Kontrolki wyświetlające plik

```
img = TkPhotoImage(:file => 'plik.gif')
```

Uwagi

- ▶ Obrazków nie można umieszczać jako samodzielnych kontrolek
- ▶ Obrazy mogą być elementem przycisków lub etykiet

Uwagi

- ▶ Obrazków nie można umieszczać jako samodzielnych kontrolerek
- ▶ Obrazy mogą być elementem przycisków lub etykiet

Przykłady

```
TkLabel.new(win) { image img; pack }
```

```
TkButton.new(win) { image img; pack }
```

```
TkclImage.new(cv, 100, 100, 'image' => img)
```

Plan wykładu

Proste środowisko graficzne

Menu

Obrazki

Wątki w aplikacjach graficznych Tkinter

Wątki

- ▶ Zadania permanentnie wykonywane podczas działania programu
- ▶ Zadania wykonywane na żądanie w niezależnym wątku

Przykład

Zegar

```
@clock = TkLabel.new(@win) { text " "; pack }  
@timer = TkAfter.new(1, -1,  
    proc { @clock.text = Time.now.to_s })  
@timer.start
```

Zadania w wątkach

```
menu.add('command',  
  :label => 'Wykonaj',  
  :command => proc { Thread.new { self.run } })
```