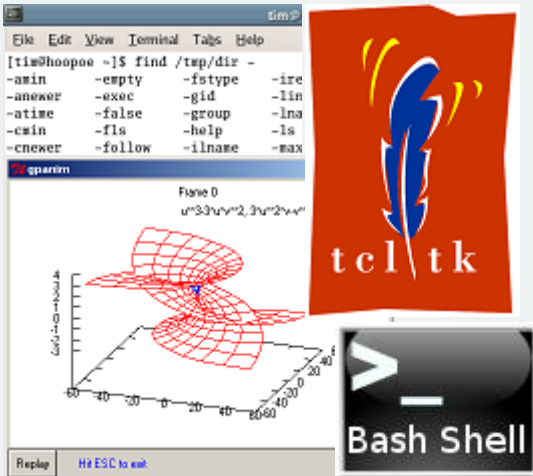




Интерпретируемые языки программирования

Лекция 4

Интерпретируемый язык Tcl.
Библиотека Tk.



Программирование с библиотекой Tk



Указать интерпретатор

`#!/usr/bin/wish`

Создать компоненты

`label .l1 -text "This is a text"`

`grid .l1 -row 0 -column 0`

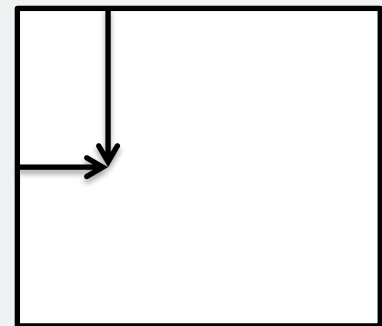
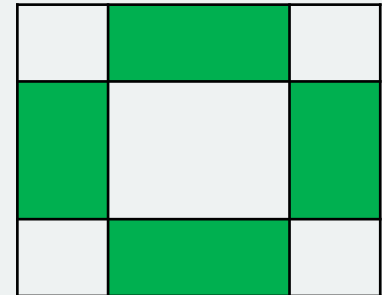
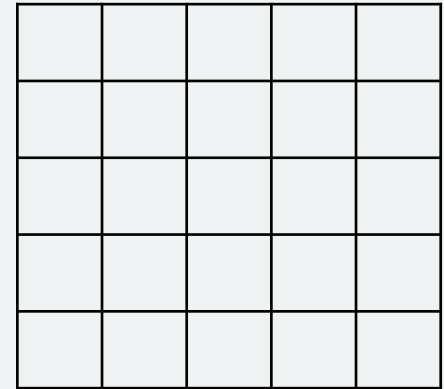
Расположить компоненты

+ Задать обработчики

Менеджеры компоновки

В Tcl/Tk доступны 3 вида менеджеров компоновки:

- `grid` – разделяет рабочую область главного окна на виртуальные ячейки, располагает все компоненты внутри ячеек;
- `pack` – позволяет расположить компоненты внутри окна относительно границ окна;
- `place` – позволяет задать абсолютное позиционирование компонентов на окне.

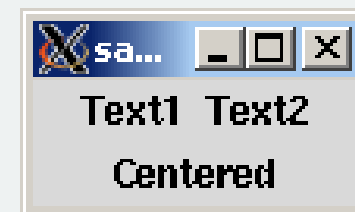
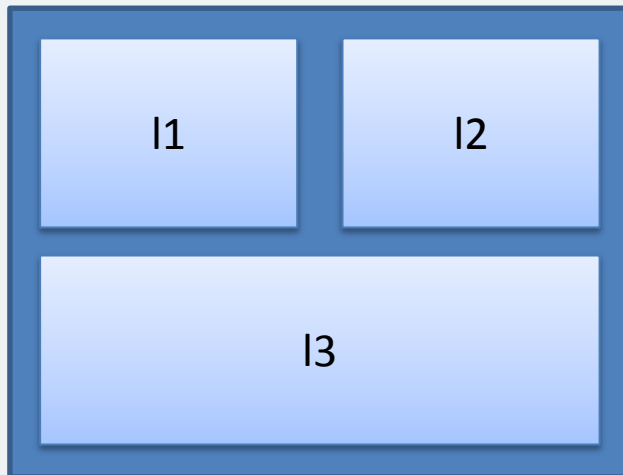
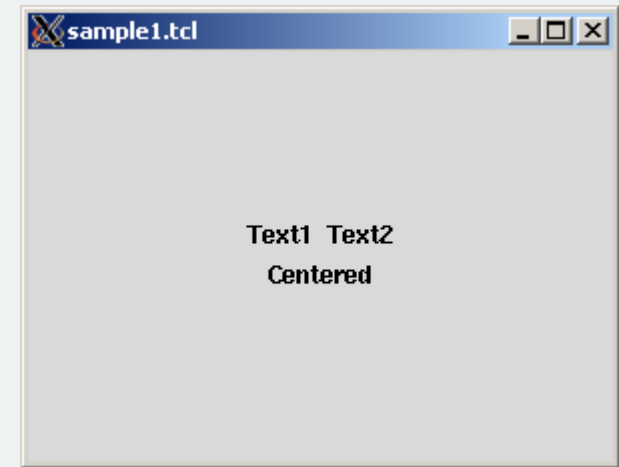


Менеджер компоновки grid

```
#!/usr/bin/wish

label .l1 -text "Text1"
label .l2 -text "Text2"
label .l3 -text "Centered"

grid .l1 -row 0 -column 0
grid .l2 -row 0 -column 1
grid .l3 -row 1 -column 0 -columnspan 2
```



Менеджер компоновки grid

Основные параметры расположения при использовании grid

- row - указание ряда, в котором располагается компонент
- column - указание столбца, в котором располагается компонент
- rowspan - указание числа строк для объединения
- columnspan - указание числа столбцов для объединения
- padx - указание отступа в пикселях от края или краёв по X
- pady - указание отступа в пикселях от края или краёв по Y

Управление главным окном

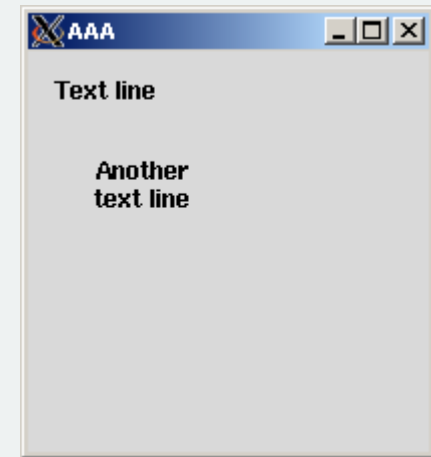
```
#!/usr/bin/wish
```

```
wm title . "AAA"
```

```
label .l1 -text "Text line"
```

```
label .l2 -text "Another\ntext line"
```

```
...
```



Управление главным окном – команда `wm`

Основные аргументы:

`title` – задание заголовка окна;

`resizable` – возможность менять размеры;

`geometry` – задание положения окна.

```
#!/usr/bin/wish
```

```
wm title . "AAA"
```

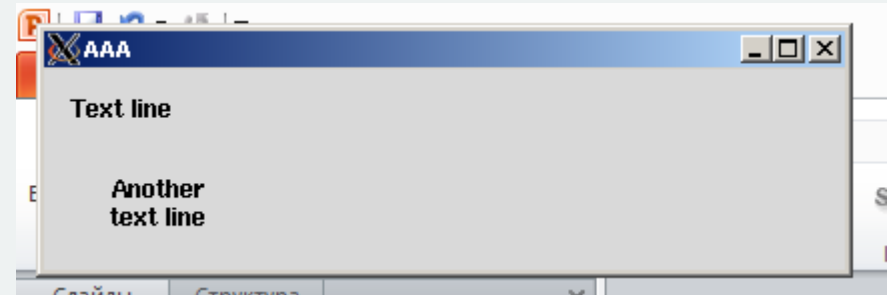
```
wm resizable . 0 0
```

```
wm geometry . =400x100+10+10
```

```
label .l1 -text "Text line"
```

```
label .l2 -text "Another\ntext line"
```

```
...
```



Основные компоненты библиотеки Tk

Метки (Label) – текстовые надписи

Кнопки (Button)

Переключатели (Checkbutton, Radiobutton)

Поля ввода (Entry)

Текстовые поля (Text)

Контекст графического устройства (Canvas)

Метки

```
#!/usr/bin/wish
```

```
label .l1 -text "First line"
```

```
label .l2 -text "Second line" -foreground Yellow \  
-background Blue
```

```
label .l3 -text "Third line" -font { -size 24 }
```

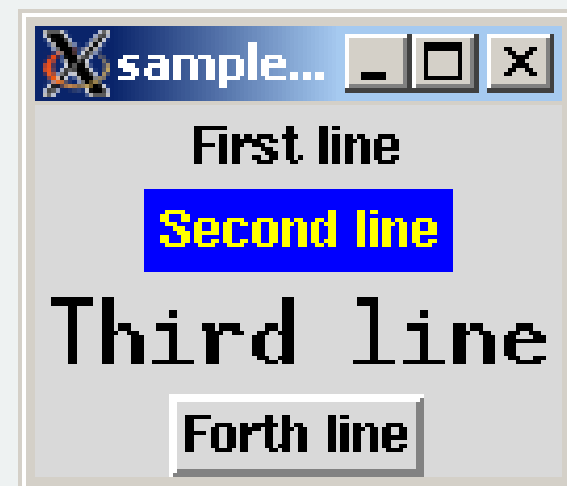
```
label .l4 -text "Forth line" -relief raised
```

```
grid .l1 -row 0
```

```
grid .l2 -row 1
```

```
grid .l3 -row 2
```

```
grid .l4 -row 3
```



Кнопки

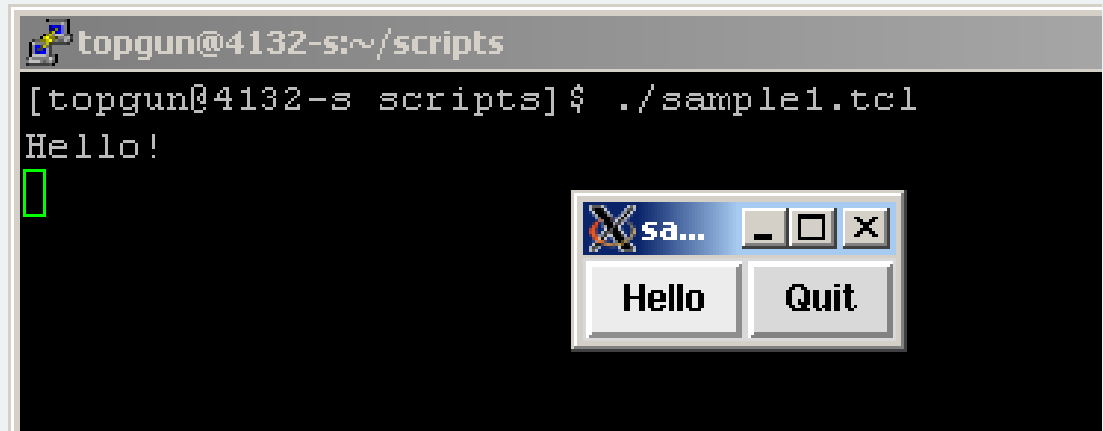


```
#!/usr/bin/wish
```

```
button .b1 -text "Hello" -command { puts stdout "Hello" }  
button .b2 -text "Quit" -command { destroy . }
```

```
grid .b1 -row 0 -column 0  
grid .b2 -row 0 -column 1
```

Кнопки: задание обработчика нажатия в виде процедуры



```
#!/usr/bin/wish
```

```
proc b1_proc {} {  
    puts stdout "Hello!"  
}
```

```
button .b1 -text "Hello" -command b1_proc  
button .b2 -text "Quit" -command { destroy . }
```

```
grid .b1 -row 0 -column 0  
grid .b2 -row 0 -column 1
```

Переключатели

```
#!/usr/bin/wish
```

```
proc print_vars {} {  
    global cValue1 cValue2 rValue
```

```
    puts stdout "Value 1 = $cValue1"  
    puts stdout "Value 2 = $cValue2"  
    puts stdout "Value   = $rValue"
```

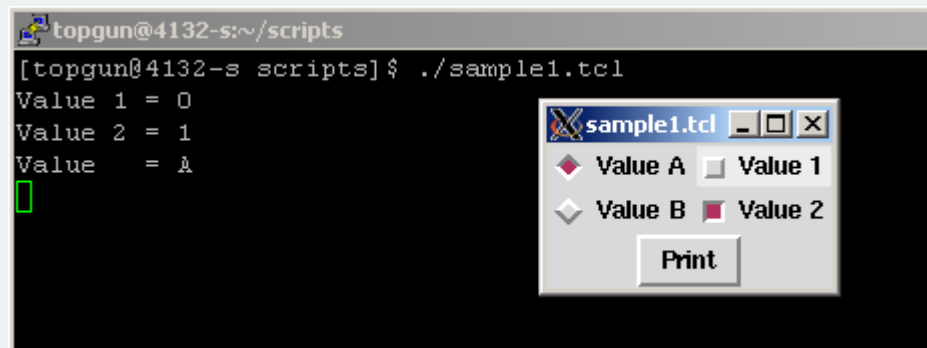
```
}
```

```
checkbox .c1 -text "Value 1" -variable cValue1  
checkbox .c2 -text "Value 2" -variable cValue2
```

```
radiobutton .r1 -text "Value A" -variable rValue -value "A"  
radiobutton .r2 -text "Value B" -variable rValue -value "B"
```

```
button .b -text "Print" -command print_vars
```

```
grid .c1 -row 0 -column 1  
grid .c2 -row 1 -column 1  
grid .r1 -row 0 -column 0  
grid .r2 -row 1 -column 0  
grid .b -row 2 -column 0 -columnspan 2
```



Поля ввода



```
#!/usr/bin/wish
```

```
label .l -text "Enter: "
```

```
entry .e -width 40 -textvariable var1
```

```
button .b -text "Clear" -command {set var1 ""}
```

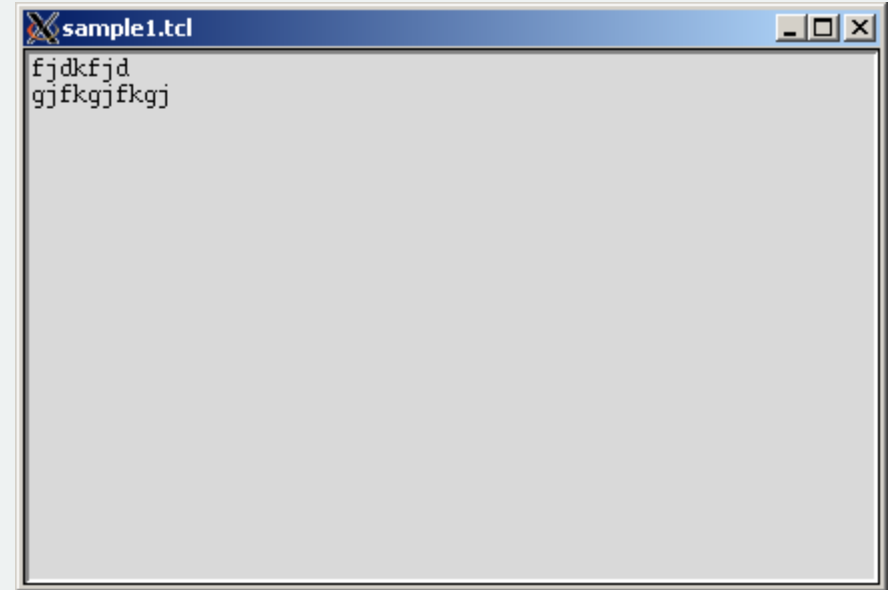
```
focus .e
```

```
grid .l -row 0 -column 0 -sticky e
```

```
grid .e -row 0 -column 1 -sticky w
```

```
grid .b -row 1 -column 0 -columnspan 2
```

Поля ввода



```
#!/usr/bin/wish
```

```
text .t -width 60 -height 20 -wrap word
```

```
grid .t -row 0 -column 0
```

Изменение свойств объектов



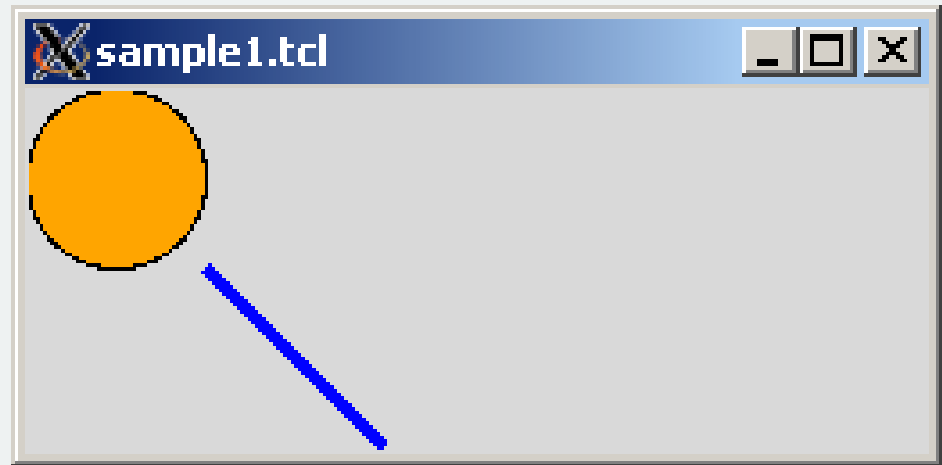
```
#!/usr/bin/wish
```

```
proc change_text { } {  
    global .b1  
    .b1 configure -text "AAA"  
}
```

```
button .b1 -text "Button" -command change_text
```

```
grid .b1 -row 0 -column 0
```

Контекст графического устройства

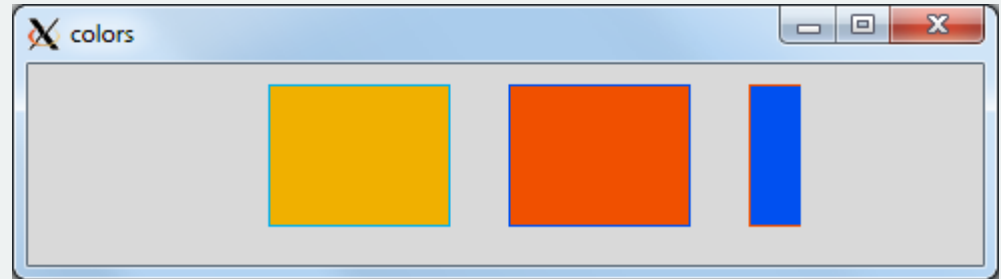


```
#!/usr/bin/wish
```

```
canvas .c -width 250 -height 100  
.c create oval 0 0 50 50 -fill orange  
.c create line 50 50 100 100 -fill blue -width 4
```

```
grid .c -row 0 -column 0
```


Контекст графического устройства



```
#!/usr/bin/wish
canvas .can
.can create rect 30 10 120 80 -outline #0bf -fill #fb0
.can create rect 150 10 240 80 -outline #05f -fill #f50
.can create rect 270 10 370 80 -outline #f50 -fill #05f

pack .can

wm title . "colors"
wm geometry . 400x100+300+300
```

Контекст графического устройства

```
#!/usr/bin/wish
```

```
canvas .can
```

```
.can create oval 10 10 80 80
```

```
.can create oval 110 10 210 80 \  
-outline #777 -fill #777
```

```
.can create rect 230 10 290 60 \  
-outline #777 -fill #777
```

```
.can create arc 30 200 90 100 -start 0 -extent 210 \  
-outline #777 -fill #777
```

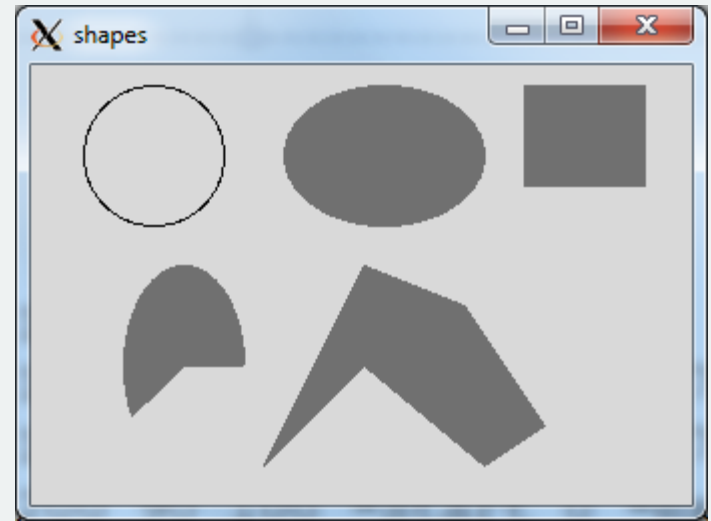
```
set points [ list 150 100 200 120 240 180 210 200 \  
150 150 100 200 ]
```

```
.can create polygon $points -outline #777 -fill #777
```

```
pack .can
```

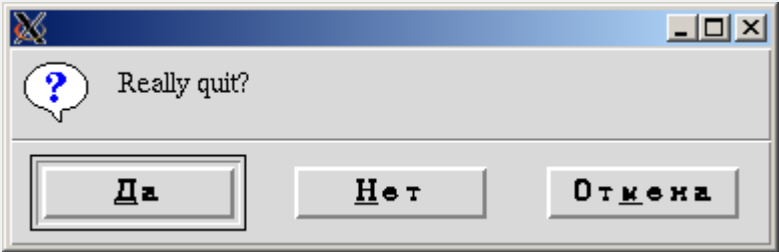
```
wm title . "shapes"
```

```
wm geometry . 330x220+300+300
```



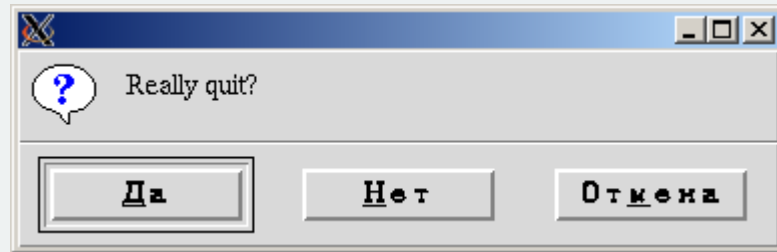
Информационные диалоги

```
tk_messageBox -message "Really quit?" \  
              -type yesno \  
              -icon question
```



- message - текст сообщения
- parent - окно, из которого вызван диалог
- title - заголовок
- type - набор кнопок
 - ok
 - okcancel
 - yesno
 - yesnocancel
 - retrycancel
 - abortretryignore
- icon - иконка сообщения
 - error
 - info
 - question
 - warning

Информационные диалоги



```
set answer [tk_messageBox -message "Really quit?" \
                          -type yesnocancel -icon error]

switch -- $answer {
    yes exit
    no { tk_messageBox -message \
                      "Quit canceled!" \
                      -type ok}
}
```

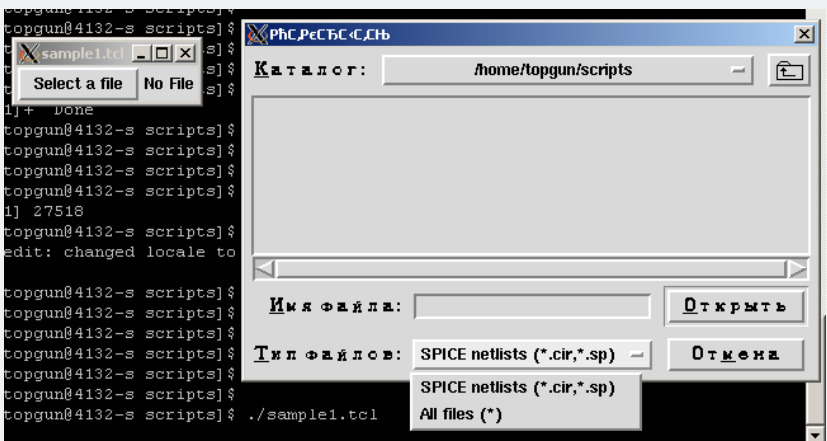
Информационные диалоги

```
set types {  
    {"SPICE netlists" {.cir .sp} }  
    {"All files" * }  
}
```

```
proc doIt {label} {  
    global types  
    set file [tk_getOpenFile -filetypes $types -parent .]  
    $label configure -text $file  
}
```

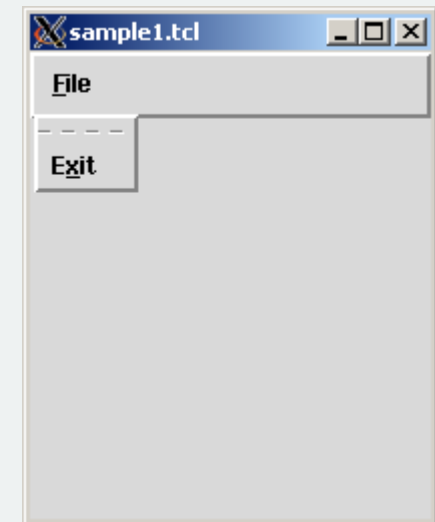
```
label .l -text "No File"  
button .b -text "Select a file" \  
    -command "doIt .l"
```

```
grid .b -row 0 -column 0  
grid .l -row 0 -column 1
```



Создание меню

```
menu .menubar  
. configure -menu .menubar  
  
.menubar add cascade -label File \  
                        -menu .menubar.file \  
                        -underline 0  
  
menu .menubar.file  
.menubar.file add command -label Exit \  
                        -underline 1 \  
                        -command exit
```



Задание на лабораторную работу

