

Графические Системы. Часть II

Лекция № 10 (ПЗ № 7)

Программирование графического
пользовательского интерфейса
средствами X-WINDOW.
ИПВУ. Tcl/TK

Программирование в TCL/TK.

Пример программирования GUI с использованием
widget'ов listbox и связанного с ним scrollbar

Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

Рассмотрим пример программирования элемента **GUI** с использованием управляющих элементов типа **listbox** и ассоциированного с ним **scrollbar'a**, с помощью которых возможен выбор одного или более элементов из списка, с прокруткой (горизонтальной) множества строк, загруженных в окно управляющего элемента **listbox**.

Будут рассмотрены и изучены следующие определения и понятия, частично введенные ранее:

Make widget command
Widget command
Standard options supported by widgets
Widget specific options

Эти типы команд и командных опций используются в формальном описании языка (Tcl/Tk), которым мы будем пользоваться, и освоить который – первоочередная задача обучающихся:

Manual Pages zu Tcl/Tk:

<http://velociraptor.mni.fh-giessen.de/TclTk/index.html>

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.

Make widget command [listbox](#)

На этой странице перечислены **Widget command, Standard options supported by widgets** и **Widget specific options** для **widget'a listbox**

The screenshot shows the 'ACTIVE TCL USER GUIDE' for the 'listbox' widget. The page title is 'ACTIVE TCL USER GUIDE' with the 'ActiveState' logo. The breadcrumb trail is 'Tcl/Tk Documentation > TkCmd > listbox'. Navigation links include 'Tcl/Tk Applications', 'Tcl Commands', 'Tk Commands', 'Tcl Library', and 'Tk Library'. The main content is organized into sections: NAME, SYNOPSIS, STANDARD OPTIONS, WIDGET-SPECIFIC OPTIONS, DESCRIPTION, INDICES, and WIDGET COMMAND. The 'listbox' widget is described as a command to create and manipulate listbox widgets. The synopsis is 'listbox pathName ?options?'. The 'STANDARD OPTIONS' section lists various options like -background, -borderwidth, -cursor, etc. The 'WIDGET-SPECIFIC OPTIONS' section lists options like -activestyle, -height, -listvariable, etc. The 'DESCRIPTION' section lists indices like number, active, anchor, end, @x,y. The 'WIDGET COMMAND' section is partially visible at the bottom.

The screenshot shows the 'ActiveTcl 8.5.7.1 Help' window. The title bar includes 'ActiveTcl 8.5.7.1 Help' and standard window controls. The menu bar has 'Скрыть', 'Назад', 'Печать', and 'Параметры'. The main content area is titled 'WIDGET COMMAND' and lists various widget commands and their options, including 'activate index', 'bbox index', 'cget option', 'configure ?option? ?value option value ...?', 'curselection', 'delete first ?last?', 'get first ?last?', 'index index', 'insert index ?element element ...?', 'itemcget index option', 'itemconfigure index ?option? ?value? ?option value ...?' with sub-options like -background color, -foreground color, -selectbackground color, -selectforeground color, 'nearest y', 'scan option args', 'scan mark x y', 'scan dragto x y.', 'see index', 'selection option arg', 'selection anchor index', 'selection clear first ?last?', 'selection includes index', 'selection set first ?last?', 'size', 'xview args', 'xview index', 'xview moveto fraction', 'xview scroll number what', 'yview ?args?', 'yview index', 'yview moveto fraction', and 'yview scroll number what'. At the bottom, there are sections for 'DEFAULT BINDINGS', 'SEE ALSO', and 'KEYWORDS'.

Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

listbox – Create and manipulate listbox widgets.

SYNOPSIS

listbox pathName ?options?

Команда **listbox** создает новое окно и делает его **widget listbox**.

Дополнительные опции команды могут быть заданы в командной строке или запрошены в базе данных конфигурационных ресурсов, таких как **colors, font, text** и **relief**.

Возвращаемый параметр – **pathName**.

Здесь вводим понятие

Standard options supported by widgets.

В **Tk** существует понятие – стандартные опции, поддерживаемые **widget'ом**. Так обозначаются в **Tk** общие **конфигурационные опции**. Не каждый **widget** поддерживает весь список этих опций, однако, если **widget** поддерживает какую-либо из перечисленных в этом списке опций (см. описание каждого конкретного **widget'a**), то **widget** может быть подвержен **реконфигурированию** по этой опции в соответствии с правилами, описанными далее:

NAME

options - Standard options supported by widgets

DESCRIPTION

[-activebackground, activeBackground, Foreground](#)
[-activeborderwidth, activeBorderWidth, BorderWidth](#)
[-activeforeground, activeForeground, Background](#)
[-anchor, anchor, Anchor](#)
[-background or -bg, background, Background](#)
[-bitmap, bitmap, Bitmap](#)
[-borderwidth or -bd, borderWidth, BorderWidth](#)
[-cursor, cursor, Cursor](#)
[-disabledforeground, disabledForeground, DisabledForeground](#)
[-exportselection, exportSelection, ExportSelection](#)
[-font, font, Font](#)
[-foreground or -fg, foreground, Foreground](#)
[-highlightbackground, highlightBackground, HighlightBackground](#)
[-highlightcolor, highlightColor, HighlightColor](#)
[-highlightthickness, highlightThickness, HighlightThickness](#)
[-image, image, Image](#)
[-insertbackground, insertBackground, Foreground](#)
[-insertborderwidth, insertBorderWidth, BorderWidth](#)
[-insertofftime, insertOffTime, OffTime](#)
[-insertontime, insertOnTime, OnTime](#)
[-insertwidth, insertWidth, InsertWidth](#)
[-jump, jump, Jump](#)
[-justify, justify, Justify](#)
[-orient, orient, Orient](#)
[-padx, padX, Pad](#)
[-pady, padY, Pad](#)
[-relief, relief, Relief](#)
[-repeatdelay, repeatDelay, RepeatDelay](#)
[-repeatinterval, repeatInterval, RepeatInterval](#)
[-selectbackground, selectBackground, Foreground](#)
[-selectborderwidth, selectBorderWidth, BorderWidth](#)
[-selectforeground, selectForeground, Background](#)
[-setgrid, setGrid, SetGrid](#)
[-takefocus, takeFocus, TakeFocus](#)
[-text, text, Text](#)
[-textvariable, textVariable, Variable](#)
[-troughcolor, troughColor, Background](#)
[-underline, underline, Underline](#)
[-wraplength, wrapLength, WrapLength](#)
[-xscrollcommand, xScrollCommand, ScrollCommand](#)
[-yscrollcommand, yScrollCommand, ScrollCommand](#)

KEYWORDS

Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

Во-первых, сравним общие [Standard options supported by widgets](#), со

[Standard options supported by listbox](#):

В приведенном в [Standard options](#) описании, "**Command-Line Name**" для каждой опции является переключателем для семейства команд и реконфигурирует команды **wiget'a** для передачи им этого значения. Например, если нами рассматривается командный переключатель **-foreground**:

Command-Line Name: **-foreground** or **-fg**

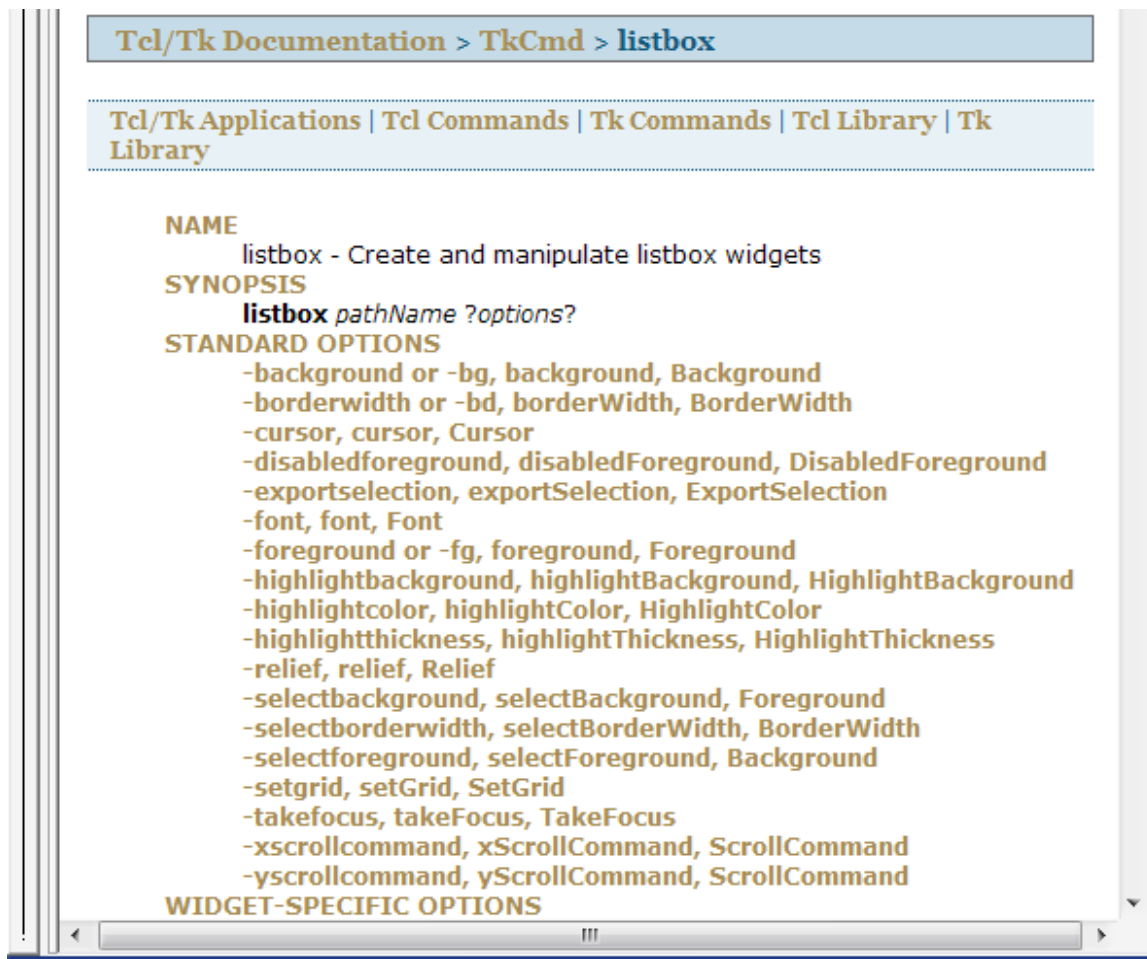
Database Name: foreground

Database Class: Foreground

Специфицирует **foreground** цвет, который применяется при визуализации **wiget'a**; и существует некий **wiget .a.b.c**, то команда

```
.a.b.c configure -foreground black
```

может быть использована для задания или изменения фонового цвета на чёрный для **wiget'a .a.b.c**. Командные переключатели могут использоваться и в форме разрешённых сокращений. "**Database Name**" указывает на имя из базы данных дополнений (например, в файле **.Xdefaults**). "**Database Class**" указывает на значение, присваиваемое элементу базы данных дополнений.



Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

Итак, **listbox** – это **widget**, который отображает список строк – одну строку в одной линии. В момент создания **listbox** пуст. Элементы **listbox** добавляются или удаляются из него с помощью специальных **widget command** для **listbox**:

insert

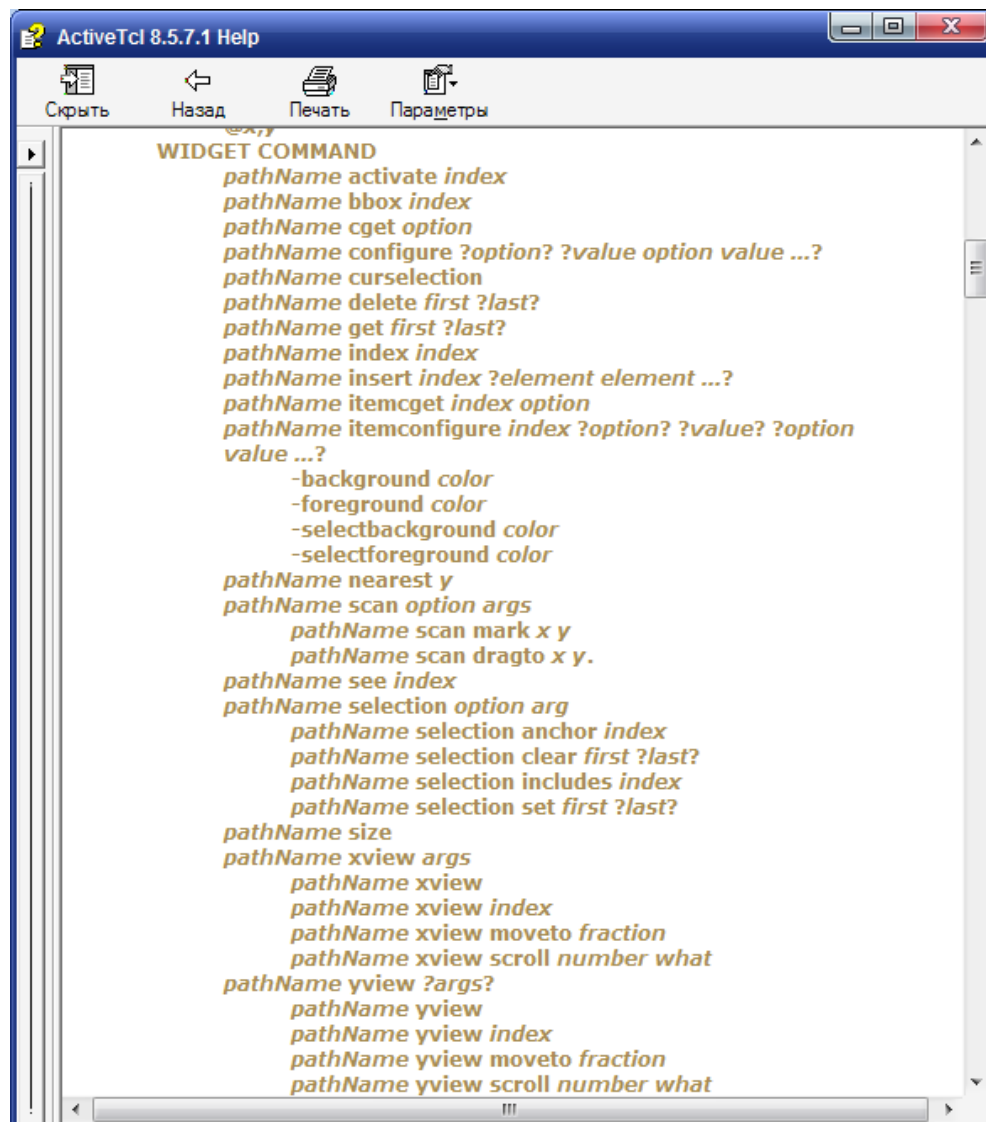
pathName insert index ?element element ...?

помещает «0» или более элементов в список сразу перед элементом, заданным **index'ом**. Если **index** специфицирован как **4** конец, новые элементы добавляются в конец. Возвращает пустую строку

delete

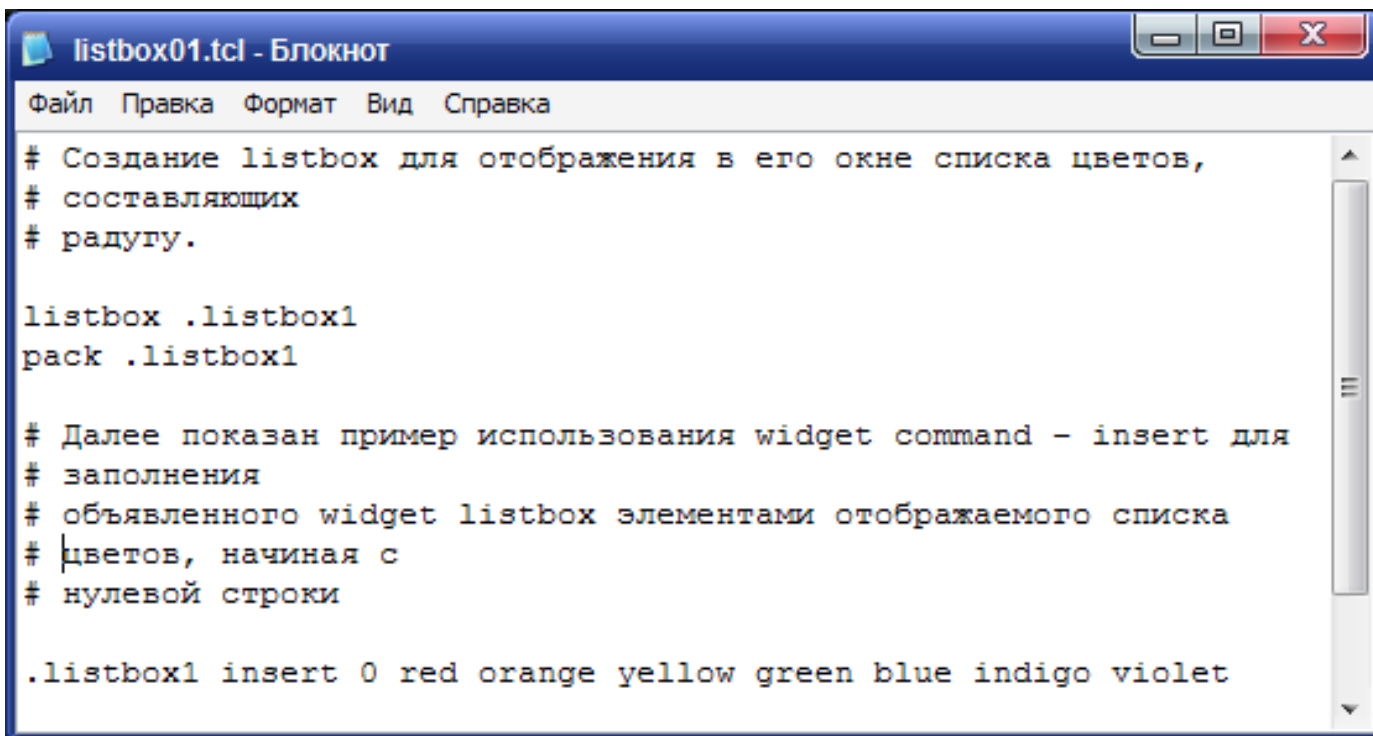
pathName delete first ?last?

Удаляет один или более элементов из **listbox'a**. **First** и **last** – индикаторы, которые специфицируют первый и последний элементы в интервале удаления. Если **last** не задан, он по умолчанию совмещается с **first**, т.е. удаляется единственный элемент



Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

Рассмотрим простейший пример - создание **listbox** для отображения в его окне списка цветов, составляющих радугу.



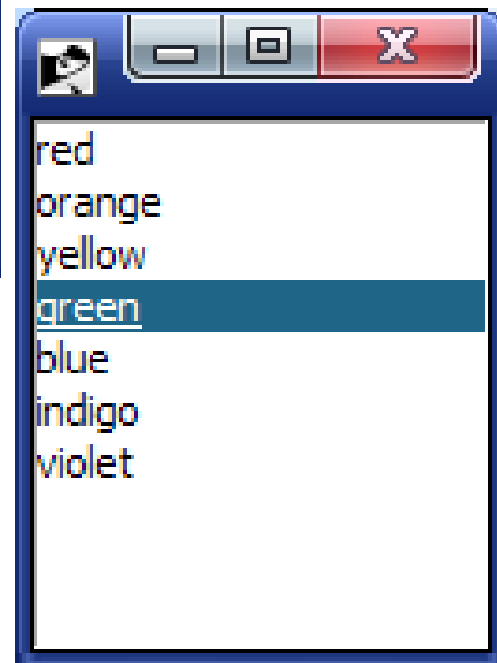
```
listbox01.tcl - Блокнот
Файл  Правка  Формат  Вид  Справка

# Создание listbox для отображения в его окне списка цветов,
# составляющих
# радугу.

listbox .listbox1
pack .listbox1

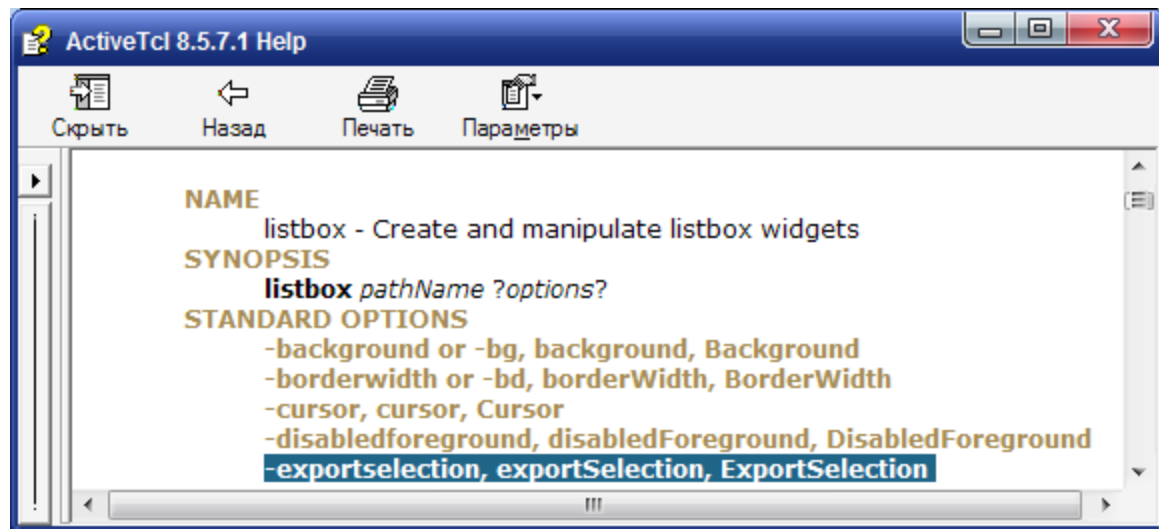
# Далее показан пример использования widget command - insert для
# заполнения
# объявленного widget listbox элементами отображаемого списка
# цветов, начиная с
# нулевой строки

.listbox1 insert 0 red orange yellow green blue indigo violet
```



Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

Здесь рассмотрим такую важную стандартную опцию, как [-exportselection](#)



Эта опция связывает
возможный выбор
(selection) в **widget
listbox** с **X-протоколом
(протоколом стандарта
X11)**. Иными словами,
опция **-exportselection**
специфицирует, является ли
выбор в **listbox**
X-выбором.

Возможные значения этой опции - **true, false, 0, 1, yes**, или **no**. По умолчанию опция принимает значение **1 (true, yes)**, что означает:

Если выбор в **widget listbox** экспортируется, то этот выбор деселектирует текущий **X-выбор**, выбор за пределами **widget (outside widget)** деселектирует любой выбор внутри **widget (inside widget)**, а сам **widget** отвечает за возврат запрошенного выбора (**selection**), если в нем – в **widget listbox** – был осуществлен выбор. Иными словами, если выбор в **listbox** экспортируется, то соблюдается протокол стандарта **X11** для обработки событий.

listbox-selection доступен как переменная типа **STRING**.

Значение **listbox-selection** - это текст выбранного элемента + элемент окончания строки.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.

Усложним **первый Пример.**

Добавим к этому примеру **Tcl-процедуру**, которая позволит визуально контролировать выбранный цвет из **.listbox1** и его название. Для этого воспользуемся командой **Tk bind**, которая позволяет связывать **Tcl-скрипты с X-событиями (X-events)**.

Переопределим **.listbox1**, добавив стандартную опцию **-exportselection**.

```
#Создаем listbox для просмотра цветов радуги

listbox .listbox1 -exportselection 1
pack .listbox1
.listbox1 insert 0 red orange yellow green blue indigo violet

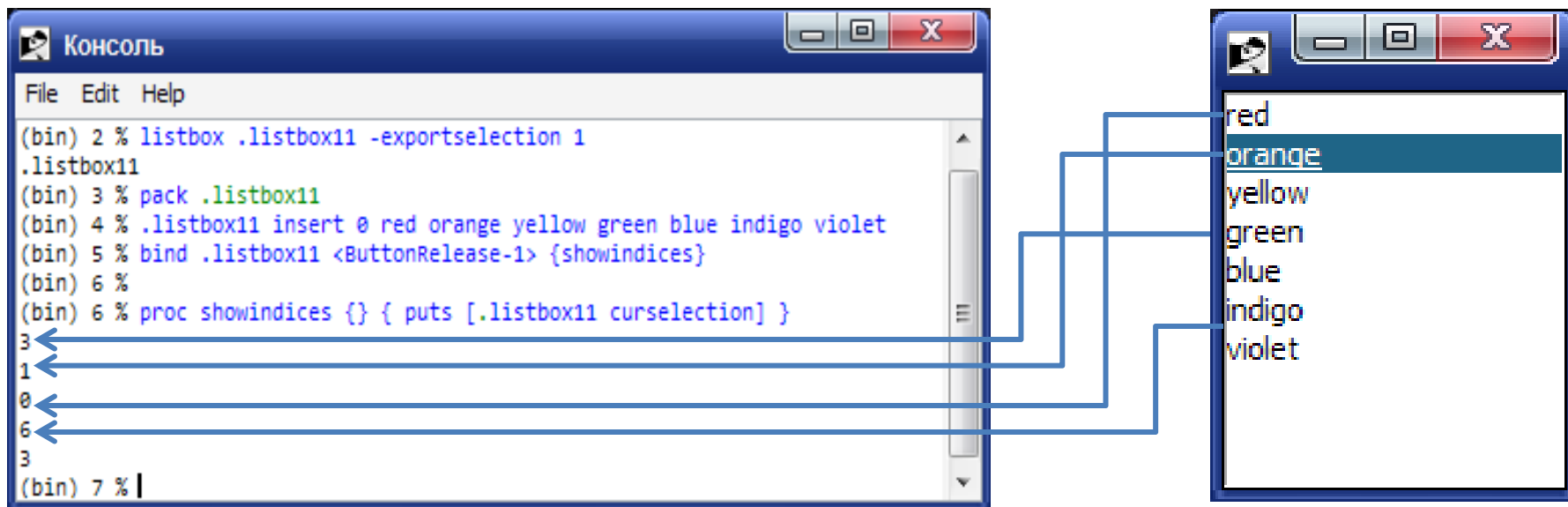
#Пока в данном listbox нет заранее объявленных связей для извлечения
строки и распознавания ее содержания. Теперь свяжем событие «отжать
клавишу» с левой клавишей мыши. Это никак не противоречит и не
препятствует процессу выбора из listbox.

bind .listbox1 <ButtonRelease-1> {showindices}

#Определяем процедуру для просмотра выбранного цвета

proc showindices {} { puts [.listbox1 curselection] }
```

Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.



После выбора цвета из **listbox'a**, отжатие левой клавиши мыши приводит к выводу на консоли номера выбранной строки, с учетом того, что нумерация строк в **listbox'e** начинается с «0».

Здесь появляется еще одна **widget command** для **listbox** - **curselection**. Команда возвращает список, содержащий номера выбранных строк из списка **listbox**, в соответствии с объявленными режимами выбора. См. [DEFAULT BINDINGS](#) и специфическую для **widget listbox** опцию [-selectmode](#).

Как видно из описания, эта опция позволяет по-разному определять поведение **listbox** с точки зрения выбора его строк – выбор может осуществляться по одной строке, по несколько строк, идущих подряд, и по несколько строк, расположенных в **listbox** не подряд. Возможные значения этой опции - **single**, **browse**, **multiple**, или **extended**. В приложениях, написанных с использованием **Tcl/Tk** чаще всего используется значение по умолчанию **browse**.

Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

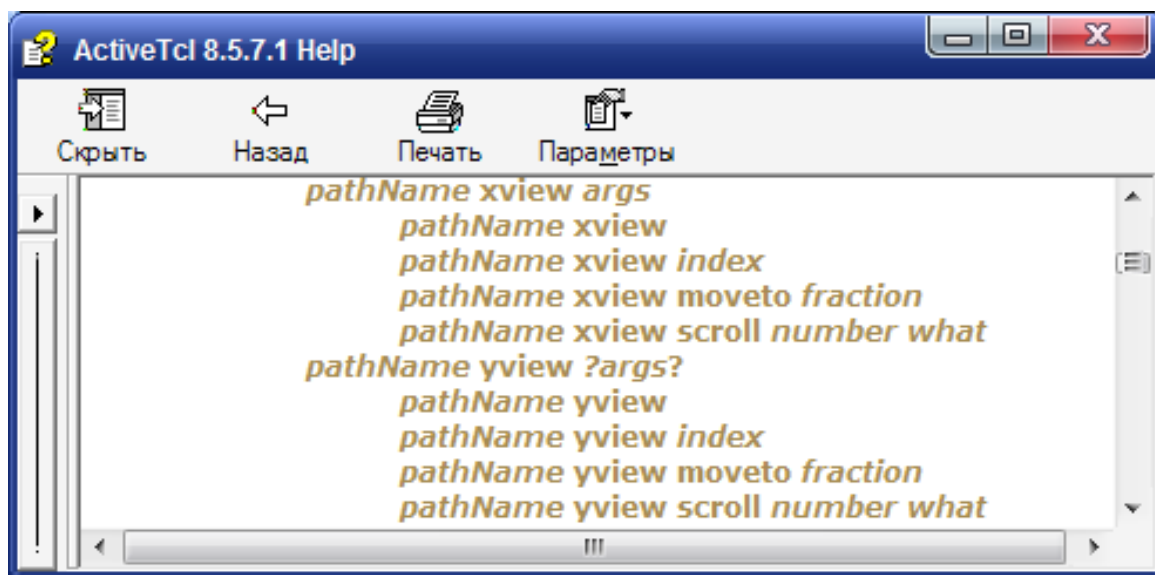
Прокрутка содержимого окон **widget'ов**.

Существуют такие **widget'ы**, у которых содержимое (список строк, строка символов или картинка) не полностью отображается в его видимом окне. Кроме [listbox](#) в список таких **widget'ов** входят:

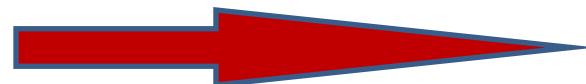
[label](#)
[entry](#)
[text](#)
[canvas](#)

Эти **widget'ы** могут быть управляемы таким образом, что относящиеся к ним **text**, **list** или **image** могут быть «перемещены» или «прокручены» по горизонтали или вертикали в окне управляющего элемента – **widget'a**. Этот процесс называется **scrolling'ом**, и он может быть реализован путем посылки **widget'y** специальной **widget command**.

Возвращаемся к **widget'y** [listbox](#). Его специальные команды (**widget command**) для прокрутки: **xview** и **yview**:

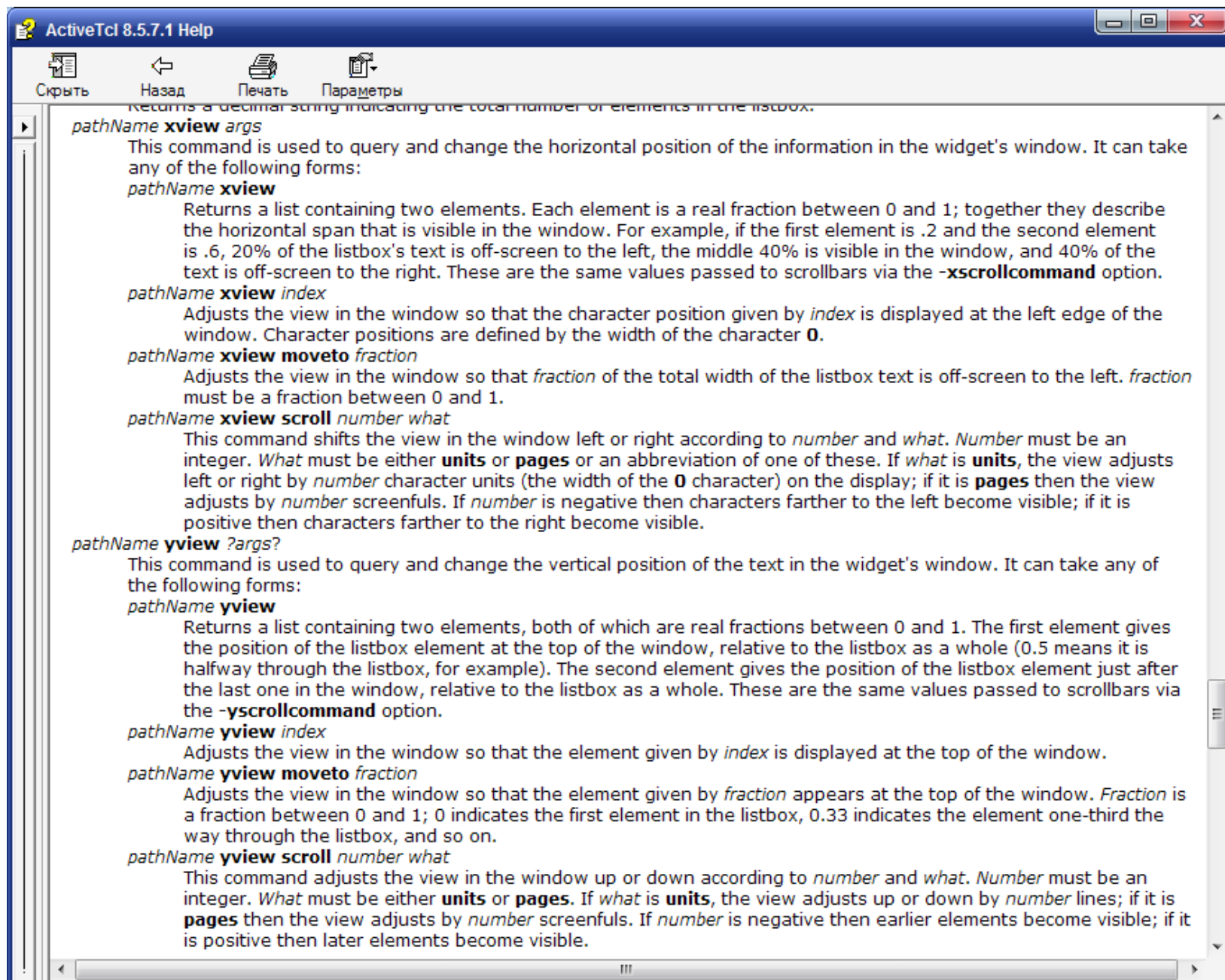


pathName xview args
Эта команда используется, чтобы опросить и изменить горизонтальную позицию информации в окне **widget 'a**.



Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.



Средства разработки графических интерфейсов в системе X Window System. Программирование в TCL/TK.

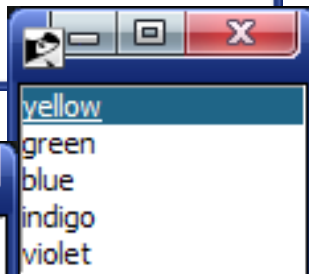
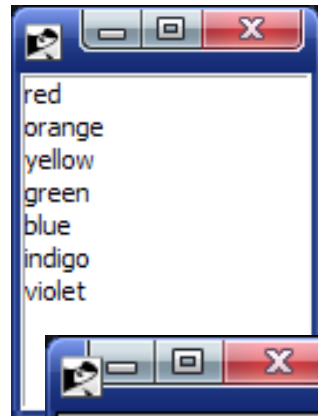
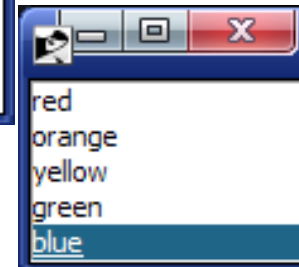
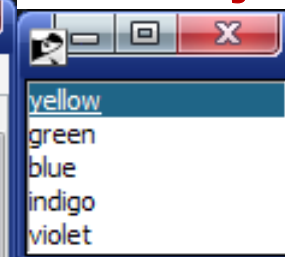
Аргументы этих **widget command** очевидны. Важно отметить, что эта команда возвращает список, состоящий из двух элементов, лежащих в интервале **0 и 1**. Первый элемент списка указывает на позицию **listbox**, находящегося наверху (слева) окна **listbox**, относительно всего списка (строки символов, если речь идет о горизонтальной прокрутке). Второй элемент списка указывает на позицию в **listbox**, лежащую сразу за последним видимым в окне элементом, относительно всего списка. Например (для **widget'a command yview**) если первый элемент списка равен **0.2**, а второй – **0.6**, это означает, что **20%** списка (сверху) находится выше верхней границы окна **widget'a listbox**, средние **40%** списка в данный момент видны в окне **widget'a listbox** и последние **40%** списка находятся ниже границы видимого окна **widget'a listbox**.

Ниже будет показано, как эти же самые значения используются для прокрутки содержимого **widget'a** (в частности, **widget'a listbox**) с применением стандартных опций, предназначенных для этого: **-yscrollcommand (-xscrollcommand)**.

```
Консоль
File Edit Help
(bin) 1 %
(bin) 1 % listbox .listbox1
.listbox1
(bin) 2 % pack .listbox1
(bin) 3 % .listbox1 insert 0 red orange yellow green blue indigo violet
(bin) 4 %
```

```
Консоль
File Edit Help
(bin) 2 % pack .listbox1
(bin) 3 % .listbox1 insert 0 red orange yellow green blue indigo violet
(bin) 4 %
(bin) 4 % .listbox1 yview 3
(bin) 5 % .listbox1 yview 10
(bin) 6 % .listbox1 yview 7
(bin) 7 % .listbox1 yview 10
(bin) 8 % .listbox1 yview 0
(bin) 9 % .listbox1 yview end
(bin) 10 %
```

Уже созданный **widget .listbox1**, содержащий список цветов радуги, может быть управляем за счет посылки ему **widget command yview**.



Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.

Рассмотрим еще один пример, в котором **listbox**, содержащий список европейских городов, управляется двумя кнопками, с помощью которых содержимое **listbox** прокручивается вертикально.

```
listbox02.tcl - Блокнот
Файл  Правка  Формат  Вид  Справка

# Создаем listbox, содержащий список европейских городов.

listbox .listbox2

# Далее следует заполнение listbox с помощью widget command insert.
# Заполнение происходит начиная с нулевой строки.

.listbox2 insert 0 Aberdeen Alicante Athens Basel Berne Bilbao Cannes\
Edinburg Glasgow Geneva Lisboa London Madrid Munich Paris\
Rome Rouen Venice Zurich

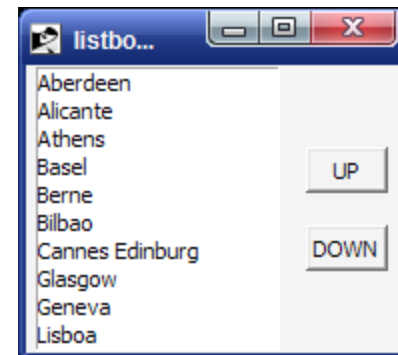
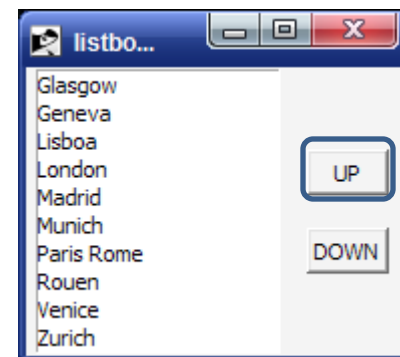
# Для прокрутки содержимого listbox организуем две кнопки «UP» и «DOWN»

# Для кнопок организуем отдельный frame

frame .fr -relief raised
button .fr.button1 -text "UP" -command {.listbox2 yview 0}
button .fr.button2 -text "DOWN" -command {.listbox2 yview 10}

# Теперь упакуем все объявленные widget

pack .listbox2 .fr -side left -padx 1m -fill x
pack .fr.button1 .fr.button2 -padx 1m -pady 2m -fill x
```



Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.



Ассоциирование прокручиваемых управляющих элементов с **widget'ом scrollbar** (на примере **widget'a listbox**).

Все **widget'ы**, допускающие горизонтальную и вертикальную прокрутку своего содержимого, используют в качестве стандартных опции **-xscrollcommand** и **-yscrollcommand**.

Именно эти опции позволяют прокручивать содержимое **widget'a** в обоих направлениях. Эти две опции (часто используются в сокращенной виде: **-xscroll** и **-yscroll**) специфицируют префикс для команды, используемой для связи с горизонтальным и вертикальным **widget'ом scrollbar**.

При любом изменении, которое происходит в окне **widget'a**, (с контентом **widget'a**), или любом другом изменении, приводящем к изменению экрана или перемещения ползунка **scrollbar**, **widget** будет выполнять **Tcl-команду** путем объединения **scroll-команды** и двух чисел. Каждое из этих чисел – это положительная десятичная дробь в диапазоне от 0 до 1; служит для индикации позиции в документе. 0 – означает начало документа, 1 – конец документа. 0.33 отмечает позицию на границе первой трети документа и т.д. Первая дробь указывает на позицию первой строки в видимом окне **widget'a**, а вторая – на позицию, сразу после последней видимой строки в окне. Команда затем передается интерпретатору **Tcl** на выполнение.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.

Ассоциирование прокручиваемых управляющих элементов с **widget'ом scrollbar** (на примере **widget'a listbox**).

Как правило, опция **-yscroll (-xscroll)** имеет значение **".tt set"**, где **.tt** - это имя (**path name**) **widget'a scrollbar**, с которым ассоциируется прокручиваемый **widget**.

В результате исполнения команды (**widget command**) **set, widget**, ассоциированный со **scrollbar'ом**, получает информацию о его состоянии и изменяет свое состояние – в видимом окне за счет исполнения опции **-yscroll (-xscroll)**.

Рассмотрим **Пример**, реализующий прокрутку **widget listbox** с использованием ассоциированного с ним **scrollbar'a**.

The image shows a screenshot of a Tcl/Tk application window titled "listbox03.tcl - Блокнот". The window contains a menu bar with "Файл", "Правка", "Формат", "Вид", and "Справка". The main text area displays the following code:

```
listbox .lb1 -exportselection 1 -selectmode single -yscroll ".sb1 set"  
pack .lb1  
.lb1 insert 0 red orange yellow green blue indigo violet weiss black blue red red red red  
bind .lb1 <ButtonRelease-1> {show}  
#  
scrollbar .sb1 -relief sunken -command ".lb1 yview"  
#  
pack .sb1 .lb1 -side right -fill y -expand yes  
proc show {} {  
puts [ .lb1 curselection ]  
}
```

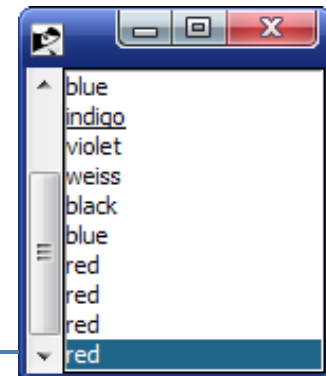
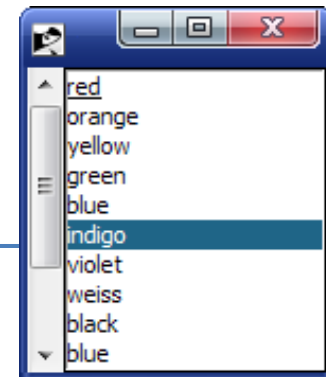
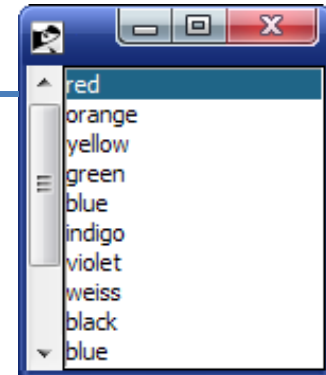
A yellow arrow labeled "Прокрутка" points from the code to two separate window instances. The left window shows a listbox with a scrollbar, displaying the list of colors: red, orange, yellow, green, blue, indigo, violet, weiss, black, blue, red, red, red, red. The right window shows a similar listbox with a scrollbar, displaying the list of colors: blue, indigo, violet, weiss, black, blue, red, red, red, red.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.

Ассоциирование прокручиваемых управляющих элементов с **widget'ом scrollbar** (на примере **widget'a listbox**).

```
Консоль
File Edit Help
(bin) 1 % listbox .lb1 -exportselection 1 -selectmode single -yscroll ".sb1 set"
.lb1
(bin) 2 % pack .lb1
(bin) 3 % .lb1 insert 0 red orange yellow green blue indigo violet weiss black blue red red red red
(bin) 4 % bind .lb1 <ButtonRelease-1> {show}
(bin) 5 % #
(bin) 6 % scrollbar .sb1 -relief sunken -command ".lb1 yview"
.sb1
(bin) 7 % #
(bin) 8 % pack .sb1 .lb1 -side right -fill y -expand yes
(bin) 9 % proc show {} {
> puts [.lb1 curselection]
> }
0 ←
5 ←
13 ←
(bin) 10 % |
```

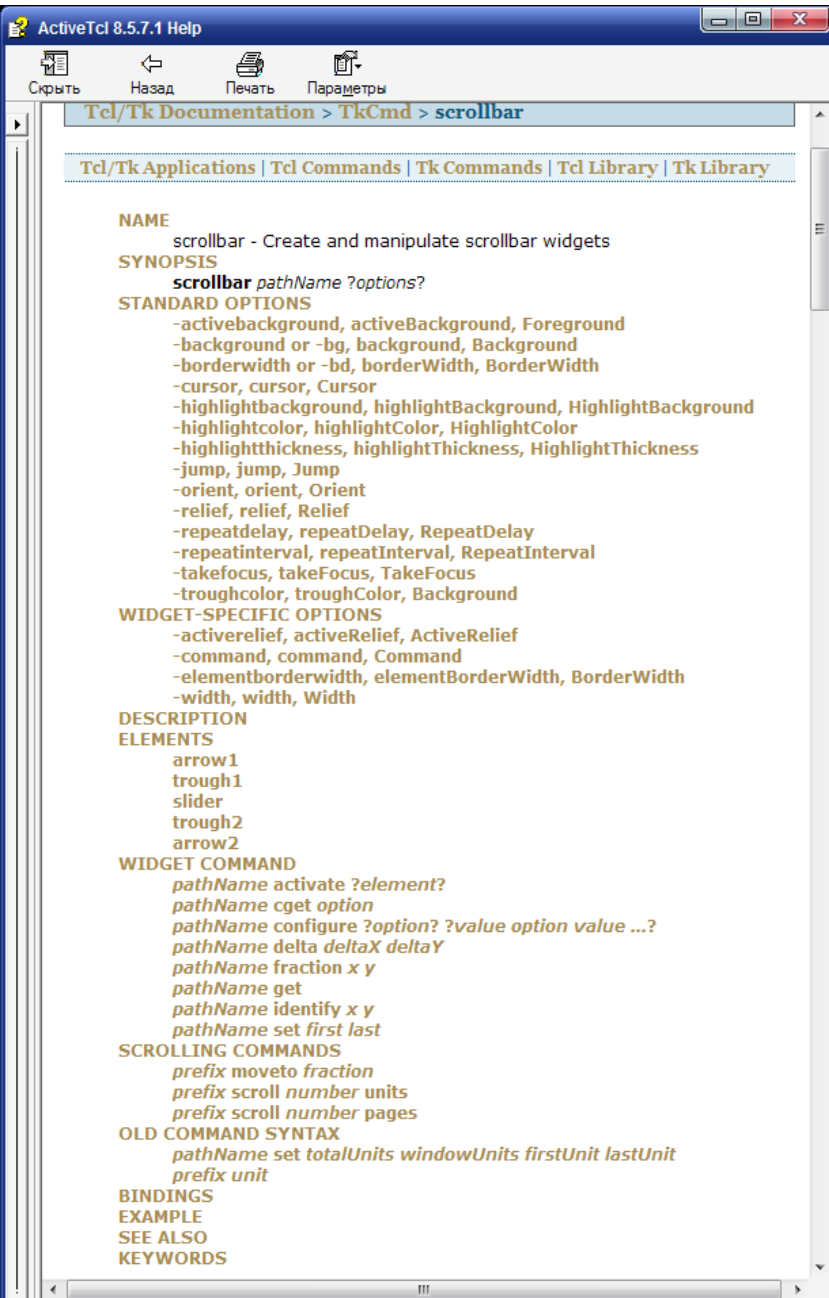


В данном Примере впервые появился **widget scrollbar**.
Рассмотрим структуру **make widget command scrollbar** подробнее.



Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.



scrollbar - Create and manipulate scrollbar widgets

scrollbar pathName ?options?

Команда создает и позволяет манипулировать **widget scrollbar**.

Для нас наибольший интерес представляет специфическая для данного **widget'a** опция [-command](#). Именно эта опция позволяет ассоциировать прокручиваемые управляющие элементы с **widget'ом scrollbar**.

Эта опция специфицирует префикс для вызова **Tcl**-команды, которая изменяет вид (**view**) в **widget'e**, ассоциируемом с **widget'ом scrollbar**. Когда пользователь манипулирует с ползунком **scrollbar**, запрашивая, таким образом, изменение вида в окне основного **widget'a (listbox, entry, label, text или canvas)**, вызывается **Tcl**-команда. Активная команда состоит из этой опции и может быть дополнена, как описано ниже. Опция **-command** почти всегда имеет значение:

".t xview" или ".t yview",

где **.t** – имя прокручиваемого **widget'a**, ассоциируемого со **scrollbar'ом**. С другой стороны, все прокручиваемые **widget'ы (listbox, entry, label, text или canvas)** имеют **widget command xview** и **yview**, которые собственно и управляют процессом скроллинга в таких **widget'ax**. Таким образом осуществляется ассоциирование скроллируемых **widget'ов** с **widget'ом scrollbar**.

Средства разработки графических интерфейсов в системе X Window System.

Программирование в TCL/TK.

```
listbox04.tcl - Блокнот
Файл  Правка  Формат  Вид  Справка

# Tcl/Tk - script для binding, listing, scrolloing, filling и select #
# Объявляем listbox .lb1, ассоциируем его со scrollbar'ом .sb1 благодаря
# использованию стандартной опции -yscrollcommand
listbox .lb1 -exportselection 1 -selectmode single -yscroll ".sb1 set"
# Объявляем scrollbar .sb1, ассоциируем его с объявленным ранее listbox .lb1 в
# специфической для данного widget опции -command
scrollbar .sb1 -relief sunken -command ".lb1 yview"
# Упаковываем объявленные widget'ы.
pack .sb1 .lb1 -side right -fill y -expand yes
# Связываем с помощью команды bind X-событие «отжатие левой клавиши мыши» с
# Tcl-процедурой show, которая позволяет нам контролировать сделанный из listbox
# выбор цвета - собственно цветом и по названию.
bind .lb1 <ButtonRelease-1> {show}
# Объявляем метку .l1 для отображения выбранного цвета и его названия.
label .l1 -relief sunken -text "String von Listbox"
# Упаковываем метку, объявляем и упаковываем кнопку «ВЫХОД»
pack .l1
button .b1 -text "EXIT" -command {exit}
pack .b1
# Подготовка widget listbox .lb1
# Заполнение widget listbox .lb1 значениями из отсортированного файла rgb.txt
exec cat -f 3- rgb.txt > rgb1.txt
# Используемые команды Tcl в цикле заполнения listbox .lb1, представленном ниже,
# будут описаны в конце листинга данного скрипта.
set rgb [open rgb1.txt r]
while {[eof $rgb] == 0} {
set colorstr [gets $rgb]
if {$colorstr != ""} {
.lb1 insert 0 $colorstr
}
}
.lb1 delete 0
close $rgb
# Процедура отображения выбранного в listbox .lb1 цвета и его названия
# на метке label .l1
proc show {} {
global w
global text
set w [.lb1 curselection]
set text [.lb1 get $w]
.l1 configure -text $text -bg $text
}
#Конец листинга программы
```

Комплексный пример работы с **widget'ом listbox**:

Tcl/Tk - script для binding, listing, scrolling, filling и select.

Короткое описание Tcl-команд, использованных в примерах данного раздела можно найти по ссылке: <http://a0601.narod.ru/ProgrTclTk.htm#aaa>