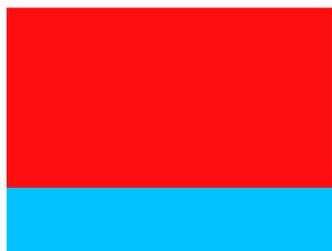


Практическая работа «Разработка программы с графическим интерфейсом»

Цель: научиться создавать программы с графическим интерфейсом при помощи библиотеки Tkinter.

Ход работы

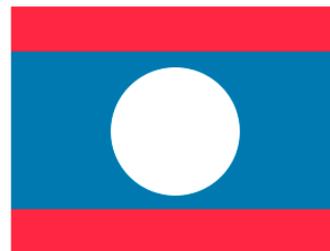
1. С использованием библиотеки Tkinter создать один из следующих флагов (варианты распределяет учитель).



А

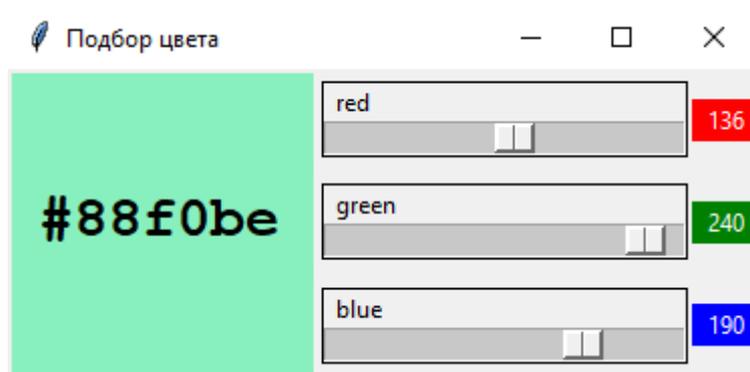


Б



В

2. Создать приложение с оконным интерфейсом, используя элемент управления Scale (Шкала). Внешний вид приложения показан на рисунке.



Окно приложения содержит холст, три шкалы с диапазоном изменения от 0 до 255 и три метки, которые красиво отображают значения, установленные на шкалах. Шкала имеет много параметров. Основные из них:

`from_` - нижняя граница диапазона (символ подчеркивания нужен, так как ключевое слово `from` используется в директиве подключения модулей);

`to` – верхняя граница диапазона;

`orient` — ориентация шкалы:

`HORIZONTAL` или `'horizontal'` — горизонтальное расположение;

`VERTICAL` или `'vertical'` — вертикальное расположение.

`length` – длина шкалы, по умолчанию равна 100 пикселей;
`sliderlength` — длина ползунка (слайдера), по умолчанию 30 пикселей;
`width` – ширина шкалы, по умолчанию равна 15 пикселей;
`relief` – тип рамки вокруг шкалы;
`label` – надпись на шкале;
`variable` – переменная, в которой содержится значение, установленное на шкале;
`showvalue` – признак вывода установленного на шкале значения (при `showvalue=0` значение не выводится);
`command` – функция-обработчик перемещения ползунка.
`tickinterval` — разность значений двух соседних делений шкалы.

По умолчанию `tickinterval=0`, в этом случае шкала не содержит делений. `resolution` — разрешение шкалы. Фактически задаёт дискретный набор значений, которые можно установить на шкале. В случае целочисленных значений разрешение шкалы равно единице. Для «грубой» настройки значения шкалы используется перемещение ползунка при движении мыши с нажатой левой кнопкой. Но в этом случае трудно точно выставить определённое значение. Для «тонкой» настройки необходимо щёлкнуть левой кнопкой мыши левее/выше или правее/ниже ползунка. В этом случае значение шкалы изменяется на минимально возможное значение.

Напишем программу:

```
# импорт графической библиотеки Питона
from tkinter import *

# создание окна приложения с "холстом" для рисования
root=Tk()
root.title('Подбор цвета')
r=IntVar(); r.set(90)
g=IntVar(); g.set(240)
b=IntVar(); b.set(190)
def changecolor(event):
    txt='#{:02x} {:02x} {:02x}'.format(
        r.get(),g.get(),b.get())
    canvas['bg']=txt
    canvas.itemconfig(color,text=txt)
```

```
canvas=Canvas(root,width=150,height=150,bg='white')
color=canvas.create_text(75,75,font=('Courier New',22,'bold'),
text=#{0:02x}{1:02x}{2:02x}'.format(
r.get(),g.get(),b.get()))
rscale=Scale(root,from_=0,to=255,orient='horizontal',
length=180,sliderlength=20,showvalue=0,
label='red',relief='solid',
command=change_color,
variable=r)
gscale=Scale(root,from_=0,to=255,orient='horizontal',
length=180,sliderlength=20,showvalue=0,
label='green',relief='solid',
command=change_color,
variable=g)
bscale=Scale(root,from_=0,to=255,orient='horizontal',
length=180,sliderlength=20,showvalue=0,
label='blue',relief='solid',
command=change_color,
variable=b)
rvalue=Label(root,width=4,bg='red', fg='white',
textvariable=r)
gvalue=Label(root,width=4,bg='green',fg='white',
textvariable=g)
bvalue=Label(root,width=4,bg='blue', fg='white',
textvariable=b)
canvas.grid(row=0,column=0,rowspan=3)
rscale.grid(row=0,column=1)
gscale.grid(row=1,column=1)
bscale.grid(row=2,column=1)
```

```
rvalue.grid(row=0,column=2)
```

```
gvalue.grid(row=1,column=2)
```

```
bvalue.grid(row=2,column=2)
```

Запустите код и проверьте правильность его выполнения.

Задание для самостоятельной работы

Измените расположение и параметры шкал так, чтобы получить следующий вид приложения:

