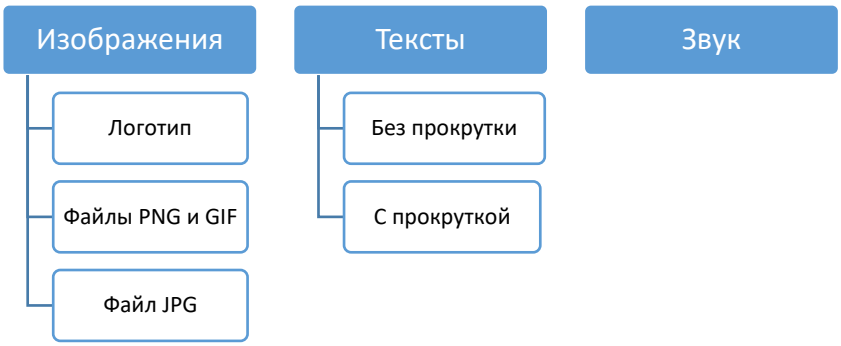
Работа с вкладками. Многоуровневое меню

В таком проекте пользователь сам решает, на какие вкладки ему переходить и в каком порядке. Самым разумным для него будет переходить от первой к последней. Но при такой навигации он не видит структуры информации, не имеет некого подобия оглавления.

Вы можете немного помочь пользователю разобраться в структуре, добавив в окно многоуровневое меню, отражающее структуру вашей информации. После того, как пользователь выберет соответствующий пункт меню, вы сделаете активной нужную вкладку.

При создании проекта вы вольны поступить по-разному: можно все изображения, тексты, настройки шрифта и т.п. загрузить в начале программы, а потом при создании фреймов использовать нужные; а можно создавать эти объекты по мере надобности – этот способ позволит не запутать в том, какой объект в каком фрейме используется (естественно, общеупотребительные элементы удобнее создавать в начале программы).

Попробуем использовать для иерархического меню виджет *Menu*. Мы будем создавать меню следующей структуры:



Создадим экземпляр меню окна:

```
mn=Menu()
```

Создадим два экземпляра подменю – для изображений и текстов:

```
mn_img = Menu()
mn_txt = Menu()
```

Для добавления пунктов основного меню вызывается метод *add_cascade()* – он позволяет пункту либо выполнять команду, либо содержать подменю. Текст пункта меню задается параметром *label*. Если пункт вызывает подменю, то должен быть указан параметр *menu=подменю*. Ели же пункт должен выполнить какую-то команду, то используется параметр *command=имя_функции* (функция предварительно должна быть описана). В нашем примере это будет выглядеть так:

```
def go_fr6():
    nb.select(fr6)

mn.add_cascade(label="Изображения", menu=mn_img)
mn.add_cascade(label="Тексты", menu=mn_txt)
mn.add_cascade(label="Звук", command=go_fr6)
```

Для добавления пунктов в подменю вызывается метод *add_command()*. Обратите внимание, что предыдущий пример мог выглядеть и так:

```
def go_fr6():
```

```
    nb.select(fr6)

mn.add_cascade(label="Изображения", menu=mn_img)
mn.add_cascade(label="Тексты", menu=mn_txt)
mn.add_command(label="Звук", command=go_fr6)
```

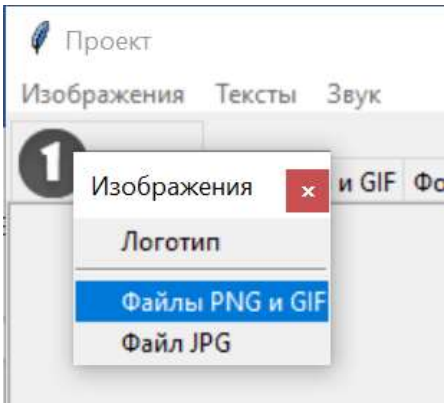
Но лучше, если все пункты верхнего уровня (основного меню) будут добавлены методом *add_cascade()*

В нашем примере добавим пункты в подменю:

```
def go_fr1():
    nb.select(fr1)

def go_fr2():
    nb.select(fr2)

def go_fr3():
    nb.select(fr3)

def go_fr4():
    nb.select(fr4)

def go_fr5():
    nb.select(fr5)

mn_img.add_command(label="Логотип",command=go_fr1)
mn_img.add_separator()
mn_img.add_command(label="Файлы PNG и GIF",command=go_fr2)
mn_img.add_command(label="Файл JPG",command=go_fr3)

mn_txt.add_command(label="Без прокрутки",command=go_fr4)
mn_txt.add_command(label="С прокруткой",command=go_fr5)
```

Обратите внимание на метод *add_separator()* – он добавляет линию-разграничитель.

По умолчанию все пункты, включающие подменю, будут содержать пунктирную линию – щелчок по ней позволяет отсоединить подменю от графического окна. Если же задать параметр метода *tearoff=0*, то подменю не сможет быть отсоединено.



Внимание!!! Недостаточно просто создать меню – его надо установить для текущего окна с помощью параметра *menu* в методе *config()*:

```
tk.config(menu=mn)
```

Вы можете добавить в полную версию предыдущей программы приведенный ниже код. Добавить можно как в начало программы – даже до создания *Notebook*, – так и в конец программы – после создания всех фреймов.

```
def go_fr1():
    nb.select(fr1)

def go_fr2():
    nb.select(fr2)

def go_fr3():
    nb.select(fr3)

def go_fr4():
    nb.select(fr4)

def go_fr5():
    nb.select(fr5)

def go_fr6():
    nb.select(fr6)

mn=Menu()

mn_img = Menu()
mn_txt = Menu()

mn.add_cascade(label="Изображения",menu=mn_img)
mn.add_cascade(label="Тексты",menu=mn_txt)
mn.add_cascade(label="Звук",command=go_fr6)

mn_img.add_command(label="Логотип",command=go_fr1)
mn_img.add_separator()
mn_img.add_command(label="Файлы PNG и GIF",command=go_fr2)
mn_img.add_command(label="Файл JPG",command=go_fr3)

mn_txt.add_command(label="Без прокрутки",command=go_fr4)
mn_txt.add_command(label="С прокруткой",command=go_fr5)

tk.config(menu=mn)
```