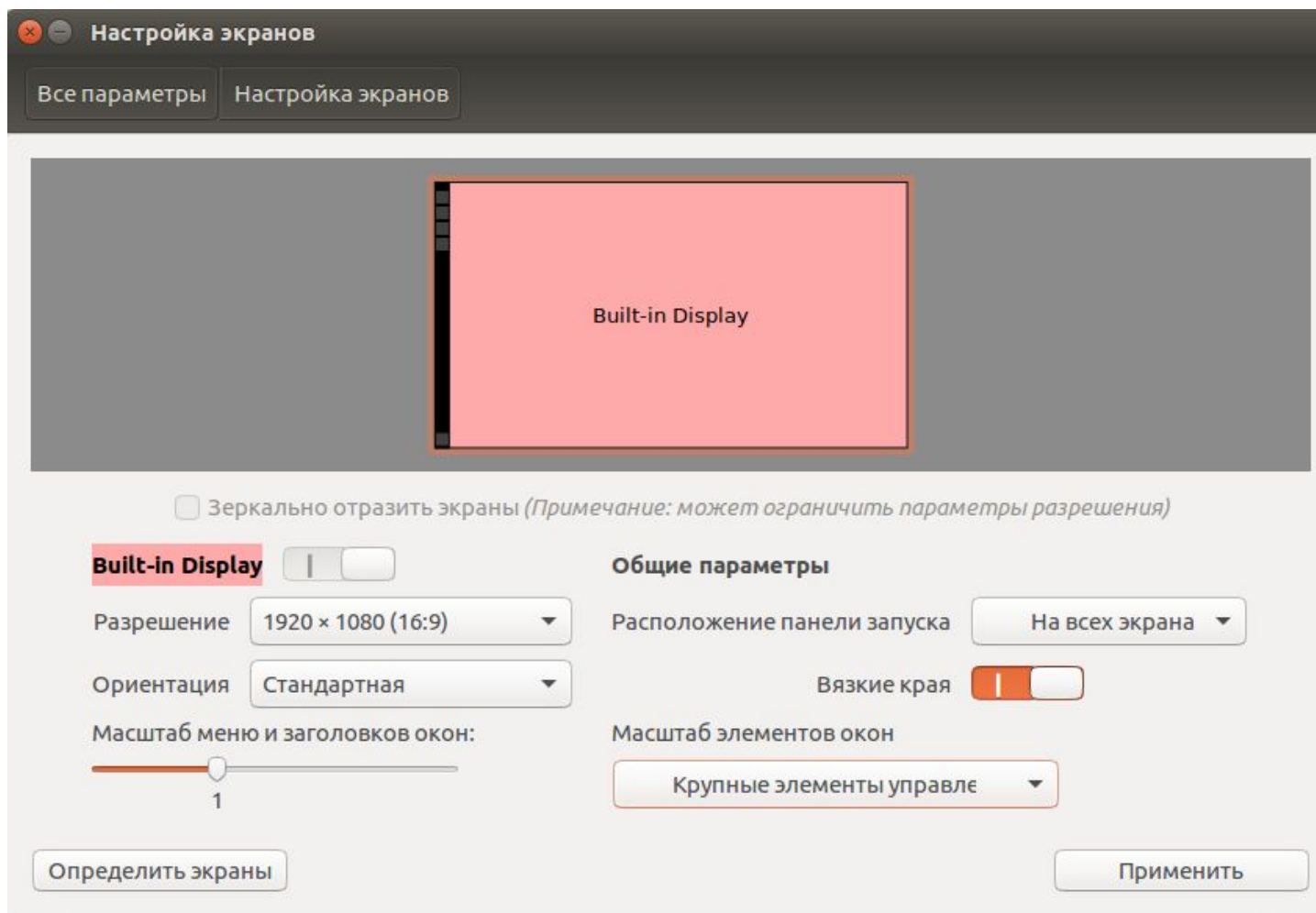


Graphical user Interface (GUI)

Элементы интерфейса представлены в виде графических изображений.



Элементы графического интерфейса (виджеты)

- надпись,
- кнопка,
- переключатель: один из,
- переключатель: несколько из списка,
- флаг,
- однострочное текстовое поле,
- многострочное текстовое поле,
- etc.

Основы ООП.

Класс -- сложный тип данных, в котором есть набор данных и функций для работы с ними.

Класс:

- свойства,
- методы.

Классы могут наследоваться друг от друга, так что дочернему классу станут доступны данные родительского класса.

Объект -- экземпляр класса.

Создание объекта:

```
obj1 = class1()
```

```
lst = list()
```

Базовый класс приложения Tk

```
from Tkinter import Tk # подключение
```

```
window = Tk()          # создание объекта
```

```
window.mainloop()     # цикл обработки событий
```

Основные методы:

- `title(str)` -- заголовок окна,
- `bind(event_name, callback)` -- вешаем обработчик на событие,
- `geometry(size)` -- устанавливает геометрию окна в формате ШИРИНАxВЫСОТА+X+Y,
- `mainloop()` -- отрисовывает форму и запускает цикл обработки событий.

В Tkinter есть три встроенных менеджера разметки: **pack**, **grid** и **place**.

- **Place** – это менеджер геометрии, который размещает виджеты, используя абсолютное позиционирование.
- **Pack** – это менеджер геометрии, который размещает виджеты по горизонтали и вертикали.
- **Grid** – это менеджер геометрии, который размещает виджеты в двухмерной сетке.

Метка, label

-- надпись, без возможности редактирования.

```
class Label(**kwargs)
```

kwargs:

- bg - str, цвет фона,
- fg - str, цвет текста,
- height - int, высота виджета,
- image - изображение,
- text - str, текст надписи,
- textvariable - StringVar, переменная связанная с виджетом,
- width - int, ширина виджета,

```
lbl1 = Label(window, text='The first example')
```

```
lbl1.pack(side=LEFT)
```

Кнопка, button

`class tkinter.Button(**kwargs)`

- background - str, цвет фона,
- bg - str, цвет фона,
- command - ссылка на функцию обработчика,
- compound,
- default,
- fg - str, цвет текста,
- height - int, высота виджета, количество строк - для обычной кнопки, пиксели - для кнопки картинки,
- image - кнопка-картинка,
- text - str, текст надписи на кнопке,
- width - str, ширина виджета, количество символов - для обычной кнопки, пиксели - для кнопки картинки.

Пример

```
btn_end = Button(window, text = 'Выйти', command=exit)
```

```
btn_end.pack(padx = 150, pady = 20)
```

```
def tog():
```

```
    if window.cget('bg') == 'green':
```

```
        window.configure(bg = 'yellow')
```

```
    else:
```

```
        window.configure(bg = 'green')
```

```
btn_tog = Button(window, text = 'Изменить', command=tog)
```

```
btn_tog.pack(padx = 150, pady = 20)
```

Метод **bind()** привязывает событие к какому-либо действию (нажатие кнопки мыши, нажатие клавиши на клавиатуре и т.д.).

Аргументы:

- название события;
- функцию, которая будет вызвана при наступлении события;
- третий аргумент (необязательный) – строка “+” – означает, что эта привязка добавляется к уже существующим.

Синтаксис связывания:

виджет.bind(событие, функция)

К событиям мыши относятся следующие события (часто используемые):

- **<Button-1>** – щелчок левой кнопки мыши.
- **<Button-2>** – щелчок средней кнопки мыши.
- **<Button-3>** – щелчок правой кнопки мыши.
- **<Double-Button-1>** – двойной клик левой кнопки мыши.
- **<Motion>** – движение мыши.

Сообщения

```
import tkinter.messagebox as box
```

Информационные:

`box.showinfo(str1, str2)` -- информационное сообщение,

`box.showwarning(str1, str2)` -- предупреждение,

`box.showerror(str1, str2)` -- сообщение об ошибке.

Сообщения с диалогом:

возвращают номер нажатой кнопки

`box.askyesno(str1, str2)`

`box.askokcancel(str1, str2)`

`box.askretrycancel(str1, str2)`

Однострочное поле ввода, Entry

class Entry(**kwargs)

- bd - ширина границы,
- insertbackground ,
- selectbackground - строка, цвет фона выделенного текста,
- selectborderwidth - строка, ширина границы выделенного текста,
- selectforeground - строка, цвет выделенного текста,
- show - строка, которая заменяет вводимые символы, например для пароля,
- state,
- validate,
- validatecommand,
- vcmd,
- width - ширина элемента,
- xscrollcommand.

```
entry = Entry(parent)
```

Методы:

`get()` -- возвращает текст виджета,

`delete(first, last=None)` -- очищает поле ввода от и до указанной позиции,

`icursor(index)` -- перемещает курсор на указанную позицию,

`index(index)` -- возвращает позицию курсора,

`insert(index, string)` -- вставляет текст в поле, по указанной позиции

`select_clear()` -- убирает выделение,

`selection_from(index)` -- устанавливает начальную позицию для выбора,

`selection_present()` -- возвращает булево, имеется ли выделение в виджете,

`select_range(start, end)`, `selection_range(start, end)` -- выделяет указанный диапазон,

`select_to(index)`, `selection_to(index)` -- выделяет до указанной позиции,

Пример

```
from tkinter import *
window = Tk()
window.title('Example')
window.geometry('600x400+100+200')
import tkinter.messagebox as box

def showdata(event):
    box.showinfo('Введенные данные', ent1.get())

lbl1 = Label(window, text='The first example')
lbl1.grid(row = 0, column = 0)
ent1 = Entry()
ent1.grid(row = 0, column = 1)
btn1 = Button(window, text = 'Ввести данные', command=exit)
btn1.grid(row = 1, column = 1)
btn1.bind('<Button-1>',showdata)
window.mainloop()
```