

# Ohjelmoinnin peruskurssi Y1

CS-A1111

# Graafinen käyttöliittymä: toiminnan lisääminen komponenttiin

## Oppimistavoitteet: tämän luennon jälkeen

- ▶ Tiedät, miten saat graafista käyttöliittymää käyttävään ohjelmaan lisättyä komponentin, joka reagoi käyttäjän toimenpiteisiin.
- ▶ Osaat kirjoittaa ohjelman, jossa on graafisessa käyttöliittymässä tekstikenttä niin, että ohjelma pystyy lukemaan käyttäjän tekstikenttään kirjoittaman arvon ja käyttämään sitä.

## Painike ikkunaan

- ▶ Ikkunaan voidaan luoda painikkeita luokan `Button` avulla.
- ▶ Pitää myös kertoa, mitä ohjelman halutaan tekevän, kun painiketta on painettu.
- ▶ Ikkunan määrittelevään luokkaan kirjoitetaan oma tapahtumankäsittelijämetodi, joka kertoo, mitä ohjelman halutaan tekevän painiketta painettaessa.
- ▶ Painiketta luodessa metodin nimi annetaan luokan `Button` `__init__`-metodille `command`-nimisenä parametrina.
- ▶ Kun painiketta on painettu, Python-tulkki kutsuu automaattisesti sille määriteltyä tapahtumankäsittelijämetodia.

# Dialogin käyttäminen

- ▶ Esimerkkiohjelmassa halutaan, että painikkeen painaminen saa aikaan sen, että avataan uusi dialogi-ikkuna.
- ▶ Moduulissa `messagebox` on valmiita funktioita, joiden avulla voi luoda dialogi-ikkunoita, joissa on haluttu otsikko ja teksti.
- ▶ Esimerkkiohjelmassa on käytetty moduulin metodia `showinfo`, joka luo tiedoteikkunan.

## Painike-esimerkki, koodi

```
from tkinter import *
import tkinter.messagebox

class Painikeikkuna:
    def __init__(self):
        self.__paaikkuna = Tk()
        self.__paaikkuna.title("Esimerkki 3")
        self.__nappi = Button(self.__paaikkuna, \
                               text = "Paina minua!", \
                               command = self.anna_ilmoitus)

        self.__nappi.pack()
        mainloop()
```

## Painike-esimerkki, koodi jatkuu

```
def anna_ilmoitus(self):  
    tkinter.messagebox.showinfo("Vastaus", \  
                                "Hienoa, osasit toimia oikein")
```

```
oma_ikkuna = Painikeikkuna()
```

## Syötteen lukeminen käyttäjältä

- ▶ Kirjoitetaan ohjelma, joka muuntaa käyttäjän fahrenheitina antaman lämpötilan celsius-asteiksi.
- ▶ Käyttäjä voi lämpötilan tekstikenttään, joka luodaan luokan `Entry` avulla.
- ▶ Kenttään kirjoitettu teksti voidaan lukea metodin `get` avulla.
- ▶ Metodi `get` palauttaa tekstin merkkijonona. Se pitää muuntaa desimaaliluvuksi laskemista varten.
- ▶ Käyttäjä ilmoittaa painiketta painamalla siitä, että hän on jo antanut luvun. Muunnoksen tekevä ohjelman osa kirjoitetaan painikkeen tapahtumankäsittelijämetodin sisään.
- ▶ Muunnoksen tulos ilmoitetaan `Label`-komponentissa, jossa olevaa tekstiä muutetaan metodin `configure` avulla.



## Lämpötilamuunnos, koodi

```
from tkinter import *

class Muunnin1:
    def __init__(self):
        self.__paaikkuna = Tk()
        self.__paaikkuna.title("Lämpötilamuunnin")
        self.__teksti = Label(self.__paaikkuna, \
                               text = "Fahrenheit-lämpötila:")
        self.__ruutu = Entry(self.__paaikkuna, \
                              width = 10)
        self.__nappi = Button(self.__paaikkuna, \
                               text = "Muunna", \
                               command = self.muunna_asteet)
        self.__vastaus = Label(self.__paaikkuna, \
                                text = "Celsius-lämpötila:")
```

## Lämpötilamuunnos, koodi jatkuu

```
self.__teksti.pack()  
self.__ruutu.pack()  
self.__vastaus.pack()  
self.__nappi.pack()  
mainloop()
```

```
def muunna_asteet(self):  
    syote = self.__ruutu.get()  
    fahrenheit = float(syote)  
    celsius = 5.0 / 9.0 * (fahrenheit - 32.0)  
    vastausteksti = "Celsius-lämpötila: {:.2f} C".\n                    format(celsius)  
    self.__vastaus.configure(text = vastausteksti)
```

```
oma_ikkuna = Muunnin1()
```

# Virheenkäsittely lämpötilamuunnokseen

- ▶ Muutetaan lämpötilamuunnoksen tekevää ohjelmaa niin, että se käsittelee mahdolliset virheelliset syötteen.
- ▶ Virheestä ilmoitetaan virhedialogi-ikkunalla. Sellainen voidaan tehdä moduulin `messagebox` funktion `showerror` avulla.

## Virheen käsittely lämpötilamuunnokseen, koodi

```
from tkinter import *
import tkinter.messagebox

class Muunnin2:
    def __init__(self):
        self.__paaikkuna = Tk()
        self.__paaikkuna.title("Lämpötilamuunnin")
        self.__teksti = Label(self.__paaikkuna, \
                               text = "Fahrenheit-lämpötila:")
        self.__ruutu = Entry(self.__paaikkuna, \
                              width = 10)
        self.__nappi = Button(self.__paaikkuna, \
                               text = "Muunna", \
                               command = self.muunna_asteet)
        self.__vastaus = Label(self.__paaikkuna, \
                                text = "Celsius-lämpötila:")
```

## Virheen käsittely lämpötilamuunnokseen, koodi jatkuu

```
self.__teksti.pack()  
self.__ruutu.pack()  
self.__vastaus.pack()  
self.__nappi.pack()  
mainloop()
```

## Virheen käsittely lämpötilamuunnokseen, koodi jatkuu

```
def muunna_asteet(self):
    syote = self.__ruutu.get()
    try:
        fahrenheit = float(syote)
        celsius = 5.0 / 9.0 * (fahrenheit - 32.0)
        vastausteksti = "Celsius-lämpötila: {:.2f} C".\
            format(celsius)
        self.__vastaus.configure(text = vastausteksti)
    except ValueError:
        virheteksti = "Anna lämpötila lukuna!"
        tkinter.messagebox.showerror("Virhe", \
            virheteksti)
```

```
oma_ikkuna = Muunnin2()
```