



Fundusze Europejskie
dla Wielkopolski

Dofinansowane przez
Unię Europejską



SAMORZĄD
WOJEWÓDZTWA
WIELKOPOLSKIEGO



Programowanie w języku Python

**Materiały dydaktyczne dla
uczestników kursu**

Dodatek

**Programowanie graficzne w Pythonie –
biblioteka tkinter**

Paweł Sobczak

Konin 2025

Tytuł
Programowanie w języku Python
Materiały dydaktyczne dla uczestników kursu

Dodatek
Programowanie graficzne w Pythonie –
biblioteka tkinter

Autor
Paweł Sobczak

Projekt pn.
„Rozwój studiów o profilu praktycznym i form kształcenia ustawicznego
dostosowanych do potrzeb Wielkopolski Wschodniej”,
realizowany przez Akademię Nauk Stosowanych w Koninie,
jest współfinansowany przez Unię Europejską
ze środków Funduszu na rzecz Sprawiedliwej Transformacji
w ramach programu Fundusze Europejskie dla Wielkopolski 2021-2027



Fundusze Europejskie
dla Wielkopolski

Dofinansowane przez
Unię Europejską



SAMORZĄD
WOJEWÓDZTWA
WIELKOPOLSKIEGO



Wydawca
Akademia Nauk Stosowanych w Koninie
ul. Przyjaźni 1, 62-510 Konin



1. Czym jest tkinter?

Tkinter - to standardowa biblioteka Pythona służąca do tworzenia graficznych interfejsów użytkownika (GUI). Jest prosty w użyciu i działa na większości systemów operacyjnych (Windows, macOS, Linux).

2. Tworzenie głównego okna

```
import tkinter as tk

root = tk.Tk() # główne okno
root.title("Moja aplikacja")
root.geometry("300x200") # szerokość x wysokość

root.mainloop()
```

3. Dodanie przykładowych elementów na oknie (widżetów)

a. Dodanie etykiety (Label)

```
label = tk.Label(root, text="Witaj w tkinter!")
label.pack()
```

b. Przycisk (Button)

```
def klik():
    print("Kliknięto przycisk!")

button = tk.Button(root, text="Kliknij mnie", command=klik)
button.pack()
```

c. Pole tekstowe (Entry)

```
entry = tk.Entry(root)
entry.pack()

def pokaz_tekst():
    tekst = entry.get()
    print("Wpisano:", tekst)

button = tk.Button(root, text="Pokaż", command=pokaz_tekst)
button.pack()
```



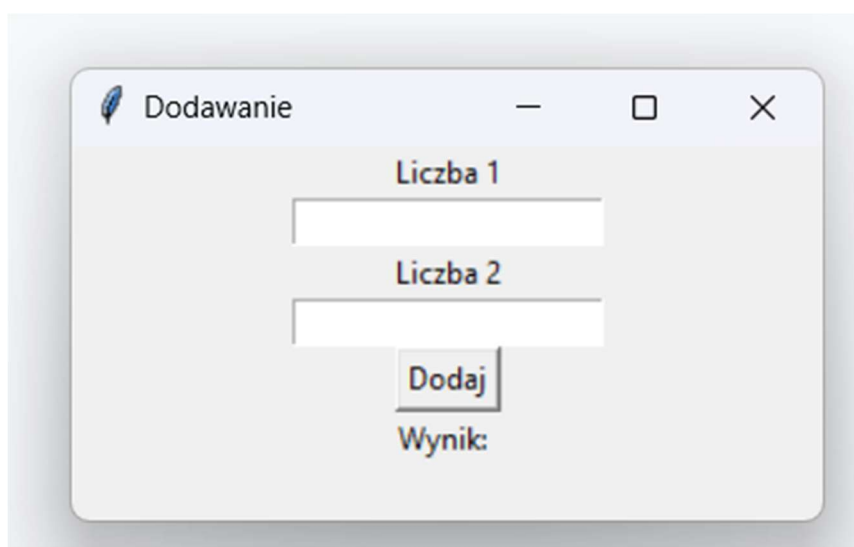
4. Układ elementów (layout)

Wyróżniamy zasadniczo 3 układy rozmieszczenia elementów na oknie:

- `pack()` – automatyczne układanie w pionie
- `grid(row=0, column=1)` – siatka (zalecane przy bardziej złożonych układach)
- `place(x=50, y=100)` – dokładne pozycjonowanie (rzadziej używane)

5. Przykład: Prosty kalkulator – dodawania dwóch liczb

a. Wygląd zewnętrzny okna



b. Kod programu

```
import tkinter as tk

def dodaj():
    try:
        a = float(entry1.get())
        b = float(entry2.get())
        wynik.config(text=f"Wynik: {a + b}")
    except ValueError:
        wynik.config(text="Błąd: podaj liczby")

root = tk.Tk()
root.title("Dodawanie")
root.geometry("300x150")

tk.Label(root, text="Liczba 1").pack()
entry1 = tk.Entry(root)
entry1.pack()
```



```
tk.Label(root, text="Liczba 2").pack()
entry2 = tk.Entry(root)
entry2.pack()

tk.Button(root, text="Dodaj", command=dodaj).pack()
wynik = tk.Label(root, text="Wynik: ")
wynik.pack()

root.mainloop()
```

6. Inne popularne widgety:

- **Checkbutton** - Przełącznik (checkbox)
- **Radiobutton** - Przycisk opcji (radio button)
- **Listbox** - Lista elementów
- **Text** - Wielowierszowe pole tekstowe
- **Canvas** - Do rysowania kształtów
- **Menu** - Pasek menu w aplikacji

7. Przykładowy projekt – książka telefoniczna

a. Wygląd okna

b. Założenia projektu:

Cel programu:

Zaimplementowanie prostego graficznego interfejsu użytkownika do:

- dodawania nowych kontaktów (imię, nazwisko, telefon, email),
- edytowania kontaktów,



- usuwania kontaktów,
- wyświetlania szczegółów kontaktu.

Funkcje:

- GUI podzielone na 3 części: lewa (lista), prawa (formularz), dolna (szegóły).
- Lista kontaktów z Listbox.
- Pola do edycji: Entry.
- Dynamiczne odświeżanie listy i szczegółów.

c. Kod programu wraz z komentarzem

```
8. from tkinter import * # Import wszystkich klas i funkcji
   z tkinter do tworzenia GUI

# Definicja klasy opisującej pojedynczy kontakt
class Kontakt:
    def __init__(self, imie, nazwisko, telefon, email):
# Konstruktor
        self.imie = imie
        self.nazwisko = nazwisko
        self.telefon = telefon
        self.email = email

listaKontaktow = [] # Lista do przechowywania obiektów
klasy Kontakt

# Funkcja dodająca nowy kontakt do listy
def dodajKontakt():
    imie = entry_Imie.get() # Pobiera tekst z
    pola imię
    nazwisko = entry_Nazwisko.get() # Pobiera nazwisko
    telefon = entry_Telefon.get() # Pobiera telefon
    email = entry_Email.get() # Pobiera email

    kontakt = Kontakt(imie, nazwisko, telefon, email) #
    Tworzy nowy obiekt kontaktu
    listaKontaktow.append(kontakt) # Dodaje go do
    listy kontaktów

    # Czyści pola po dodaniu
    entry_Imie.delete(0,END)
    entry_Nazwisko.delete(0,END)
    entry_Telefon.delete(0,END)
    entry_Email.delete(0,END)

    entry_Imie.focus() # Ustawia kursor z
    powrotem na pole imię
```



```
        pokazKontakty()                                # Odświeża listę
kontaktów w GUI

# Funkcja pokazująca listę kontaktów w listboxie
def pokazKontakty():
    listbox_ListaKontaktow.delete(0, END)  # Czyści cały
listbox
    for x, y in enumerate(listaKontaktow): # Iteruje po
wszystkich kontaktach
        listbox_ListaKontaktow.insert(x, f"{y.imie}
{y.nazwisko}") # Wstawia kontakt do listboxa

# Funkcja do usuwania wybranego kontaktu
def usunKontakt():
    index = listbox_ListaKontaktow.index(ACTIVE) #
Pobiera indeks zaznaczonego elementu
    listaKontaktow.pop(index)                  # Usuwa
z listy
    pokazKontakty()                            #
Odświeża listę

# Funkcja pokazująca szczegóły wybranego kontaktu
def pokazSzczegoly():
    index = listbox_ListaKontaktow.index(ACTIVE) #
Indeks zaznaczonego kontaktu
    # Ustawienie tekstu etykiet na dane kontaktu

label_SzczegolyKontaktu_ImieV.config(text=listaKontaktow[
index].imie)

label_SzczegolyKontaktu_NazwiskoV.config(text=listaKontak
tow[index].nazwisko)

label_SzczegolyKontaktu_TelefonV.config(text=listaKontakt
ow[index].telefon)

label_SzczegolyKontaktu_EmailV.config(text=listaKontaktow
[index].email)

# Funkcja przygotowująca formularz do edycji wybranego
kontaktu
def edytujKontakt():
    index = listbox_ListaKontaktow.index(ACTIVE) #
Indeks zaznaczonego kontaktu

    # Czyści pola formularza
    entry_Imie.delete(0, END)
    entry_Nazwisko.delete(0, END)
    entry_Telefon.delete(0, END)
```



```
entry_Email.delete(0, END)

# Wstawia do pól dane kontaktu
entry_Imie.insert(0, listaKontaktow[index].imie)
entry_Nazwisko.insert(0,
listaKontaktow[index].nazwisko)
entry_Telefon.insert(0,
listaKontaktow[index].telefon)
entry_Email.insert(0, listaKontaktow[index].email)

# Zmienia działanie przycisku dodaj, aby teraz
edytował
button_DodajKontakt.config(text="Zmień kontakt",
command=zmienKontakt)

# Funkcja kończąca edycję kontaktu
def zmienKontakt():
    index = listbox_ListaKontaktow.index(ACTIVE) #
    Indeks edytowanego kontaktu

    # Pobranie danych z pól
    imie = entry_Imie.get()
    nazwisko = entry_Nazwisko.get()
    telefon = entry_Telefon.get()
    email = entry_Email.get()

    # Nadpisanie danych w obiekcie
    listaKontaktow[index].imie = imie
    listaKontaktow[index].nazwisko = nazwisko
    listaKontaktow[index].telefon = telefon
    listaKontaktow[index].email = email

    # Czyści formularz
    entry_Imie.delete(0, END)
    entry_Nazwisko.delete(0, END)
    entry_Telefon.delete(0, END)
    entry_Email.delete(0, END)

    entry_Imie.focus() # Ustawia kursor na polu imię

    # Przywraca oryginalne ustawienie przycisku
    button_DodajKontakt.config(text="Dodaj kontakt",
command=dodajKontakt)

    pokazKontakty() # Odświeżenie listy

# == Interfejs graficzny ==

root = Tk()
```



Fundusze Europejskie
dla Wielkopolski

Dofinansowane przez
Unię Europejską



SAMORZĄD
WOJEWÓDZTWA
WIELKOPOLSKIEGO

```
root.geometry("700x300")           # Ustawienie rozmiaru
okna                                okna
root.title("Książka telefoniczna") # Tytuł okna

# Tworzenie ramek (lewa, prawa, dolna)
ramkaLewa = Frame(root)
ramkaPrawa = Frame(root)
ramkaDolna = Frame(root)

# Rozmieszczenie ramek w siatce
ramkaLewa.grid(row=0, column=0, pady=(20,20), padx=10)
ramkaPrawa.grid(row=0, column=1, sticky=N, pady=(20,20))
ramkaDolna.grid(row=1, column=0, columnspan=2, sticky=W,
padx=10)

# === LEWA RAMKA ===

label_ListaKontaktow = Label(ramkaLewa, text="Lista
kontaktów")
listbox_ListaKontaktow = Listbox(ramkaLewa, width=25,
height=7)
button_PokazSzczegoly = Button(ramkaLewa, text="Pokaż
szczegóły kontaktu", command=pokazSzczegoly)
button_UsunKontakt = Button(ramkaLewa, text="Usuń
kontakt", command=usunKontakt)
button_EdytujKontakt = Button(ramkaLewa, text="Edytuj
kontakt", command=edytujKontakt)

# Umieszczenie widgetów w ramce lewej
label_ListaKontaktow.grid(row=0, column=0, columnspan=3)
listbox_ListaKontaktow.grid(row=1, column=0,
columnspan=3)
button_PokazSzczegoly.grid(row=2, column=0)
button_UsunKontakt.grid(row=2, column=1)
button_EdytujKontakt.grid(row=2, column=2)

# === PRAWA RAMKA ===

# Formularz dodawania nowego kontaktu
label_NowyKontakt = Label(ramkaPrawa, text="Nowy
kontakt")
label_Imie = Label(ramkaPrawa, text="Imię:")
label_Nazwisko = Label(ramkaPrawa, text="Nazwisko:")
label_Telefon = Label(ramkaPrawa, text="Telefon:")
label_Email = Label(ramkaPrawa, text="Email:")

# Pola wejściowe
entry_Imie = Entry(ramkaPrawa)
entry_Nazwisko = Entry(ramkaPrawa, width=30)
```



```
entry_Telefon = Entry(ramkaPrawa)
entry_Email = Entry(ramkaPrawa)

# Przycisk dodawania kontaktu
button_DodajKontakt = Button(ramkaPrawa, text="Dodaj
kontakt", command=dodajKontakt)

# Rozmieszczenie formularza w ramce
label_NowyKontakt.grid(row=0, column=0, columnspan=2)
label_Imie.grid(row=1, column=0, sticky=W)
label_Nazwisko.grid(row=2, column=0, sticky=W)
label_Telefon.grid(row=3, column=0, sticky=W)
label_Email.grid(row=4, column=0, sticky=W)
entry_Imie.grid(row=1, column=1, sticky=W)
entry_Nazwisko.grid(row=2, column=1, sticky=W)
entry_Telefon.grid(row=3, column=1, sticky=W)
entry_Email.grid(row=4, column=1, sticky=W)
button_DodajKontakt.grid(row=5, column=0, columnspan=2)

# === DOLNA RAMKA ===

# Wyświetlanie szczegółów kontaktu
label_SzczegolyKontaktu = Label(ramkaDolna,
text="Szczegóły Kontaktu")
label_SzczegolyKontaktu_Imie = Label(ramkaDolna,
text="Imie")
label_SzczegolyKontaktu_ImieV = Label(ramkaDolna,
text="...", width=10)
label_SzczegolyKontaktu_Nazwisko = Label(ramkaDolna,
text="Nazwisko")
label_SzczegolyKontaktu_NazwiskoV = Label(ramkaDolna,
text="...", width=10)
label_SzczegolyKontaktu_Telefon = Label(ramkaDolna,
text="Telefon")
label_SzczegolyKontaktu_TelefonV = Label(ramkaDolna,
text="...", width=10)
label_SzczegolyKontaktu_Email = Label(ramkaDolna,
text="Email")
label_SzczegolyKontaktu_EmailV = Label(ramkaDolna,
text="...", width=10)

# Ułożenie elementów szczegółów
label_SzczegolyKontaktu.grid(row=0, column=0,
columnspan=8, sticky=W)
label_SzczegolyKontaktu_Imie.grid(row=1, column=0)
label_SzczegolyKontaktu_ImieV.grid(row=1, column=1)
label_SzczegolyKontaktu_Nazwisko.grid(row=1, column=2)
label_SzczegolyKontaktu_NazwiskoV.grid(row=1, column=3)
label_SzczegolyKontaktu_Telefon.grid(row=1, column=4)
```



Fundusze Europejskie
dla Wielkopolski

Dofinansowane przez
Unię Europejską



SAMORZĄD
WOJEWÓDZTWA
WIELKOPOLSKIEGO

```
label_SzczegolyKontaktu_TelefonV.grid(row=1, column=5)  
label_SzczegolyKontaktu_Email.grid(row=1, column=6)  
label_SzczegolyKontaktu_EmailV.grid(row=1, column=7)
```

```
# Uruchomienie głównej pętli aplikacji  
root.mainloop()
```