**pack() 예제**

from tkinter import\*

#Tk 클래스 인스턴스 생성, 변수 win에 할당

win = Tk()

win.title('pack 예제')

#Label 위젯 생성

label = Label(win, width=50, height=5)

#pack 메서드 레이블에서 호출, 기본 창 내에서 표시

label.pack()

#버튼 위젯 생성

btn1 = Button(label, text="top")

btn1.pack(side = "top")

btn2 = Button(label, text="left")

btn2.pack(side = "left")

btn3 = Button(label, text="right")

btn3.pack(side = "right")

btn4 = Button(label, text="bottom")

btn4.pack(side = "bottom")

#창 크기 지정

win.geometry("300x300")

win.mainloop()

**레이블, 엔트리, 버튼 위젯으로 윈도 생성 예제**

from tkinter import \*

#tk 클래스 인스턴스 생성, 변수 win에 할당

win = Tk()

win.title('여러 위젯 구성')

#label 위젯 생성

lb1 = Label(win, text="레이블(Lable)")

#pack 메소드 호출

lb1.pack()

txt = Entry(win)

#insert 메소드로 초기값을 Entry로 설정

txt.insert(0, '엔트리(Entry)')

txt.pack()

#버튼 위젯 생성

btn = Button(win, text = "OK")

btn.pack()

win.mainloop()

**체크박스 선택 후 버튼 클릭 시 문구 출력 이벤트 예제**

import tkinter as tk

from tkinter import \*

win = tk.Tk()

win.title("체크박스 예제")

win.geometry("600x400")

#버튼 클릭하면 선택한 체크버튼을 선택하였다는 문구 출력하는 함수 설정

def sub\_chk():

if ChkVar1.get() == 1:

lb2.configure(text = "파이썬을 선택하였습니다.")

elif ChkVar2.get() == 1:

lb2.configure(text = "자바를 선택하였습니다.")

elif ChkVar3.get() == 1:

lb2.configure(text = "알고리즘을 선택하였습니다.")

#"과목을 선택하세요" 라벨 생성

lb1 = tk.Label(win, text="과목을 선택하세요.", bg = "yellow")

lb1.grid(row = 0, column = 0, columnspan = 3)

#"파이썬", "자바", "알고리즘" 체크버튼 생성

ChkVar1 = tk.IntVar()

chk1 = tk.Checkbutton(win, text = "파이썬", variable = ChkVar1)

chk1.grid(row = 1, column = 0, sticky = "w")

ChkVar2 = tk.IntVar()

chk2 = tk.Checkbutton(win, text = "자바", variable = ChkVar2)

chk2.grid(row = 1, column = 1, sticky = "w")

ChkVar3 = tk.IntVar()

chk3 = tk.Checkbutton(win, text = "알고리즘", variable = ChkVar3)

chk3.grid(row = 1, column = 2, sticky = "w")

#결과물 출력되는 라벨 생성

lb2 = tk.Label(win)

lb2.grid(row = 2, column = 0, sticky = "w", columnspan = 3)

#체크버튼을 선택 후 클릭 시 함수 실행

btn1 = tk.Button(win, text = "선택", command = sub\_chk)

btn1.grid(row = 1, column = 3, sticky = "w")

win.mainloop()

**입력 받아 저장하는 막대그래프 생성**

# tkinter 모듈을 tk로 불러옴

import tkinter as tk

# tkinter.messagebox 모듈을 mb로 불러옴

import tkinter.messagebox as mb

# matplotlib.pyplot 모듈을 plt로 불러옴

import matplotlib.pyplot as plt

# BarChartApp 클래스 정의

class BarChartApp(tk.Frame):

# 생성자 함수

def \_\_init\_\_(self, master=None):

super().\_\_init\_\_(master)

self.master = master

self.master.title("Bar Chart App")

self.create\_widgets()

# 위젯을 생성하는 함수

def create\_widgets(self):

# 데이터 입력을 위한 라벨과 엔트리 위젯 생성

tk.Label(self.master, text="apple").grid(row=0, column=0, padx=10, pady=10)

self.apple\_entry = tk.Entry(self.master)

self.apple\_entry.grid(row=0, column=1)

tk.Label(self.master, text="banana").grid(row=1, column=0, padx=10, pady=10)

self.banana\_entry = tk.Entry(self.master)

self.banana\_entry.grid(row=1, column=1)

# 막대그래프 출력 함수

def show\_bar\_chart(self):

# 엔트리 위젯에서 입력받은 값을 변수에 저장

try:

apple\_value = float(self.apple\_entry.get())

banana\_value = float(self.banana\_entry.get())

pear\_value = float(self.pear\_entry.get())

# 숫자가 아닌 값을 입력한 경우 오류창을 띄움

except ValueError:

mb.showerror("Error", "숫자를 입력해주세요.")

return

# 막대그래프 생성

labels = ["apple", "banana", "pear"]

values = [apple\_value, banana\_value, pear\_value]

plt.bar(labels, values)

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

# tkinter 루트 윈도우 생성

root = tk.Tk()

# BarChartApp 클래스의 인스턴스 생성

app = BarChartApp(master=root)

# 이벤트 루프 실행

app.mainloop()

**Entry값 입력 받아 Label에 출력**

from tkinter import \*

from tkinter import ttk

# Tkinter 인스턴스 생성

win = Tk()

# Window 사이즈 지정

win.geometry("300x200+100+100")

win.resizable(False, False)

# get() : entry에 텍스트를 문자열로 반환

# Cal 함수 정의, 버튼 클릭 시 호출되며 Entry 위젯에 입력된 값을 기준으로 결과

def Cal():

label.configure(text="결과 값 = " + str(eval(entry.get())))

#Entry 클래스를 이용한 Entry 위젯 생성. 변수 entry에 할당

entry = Entry(win)

#entry 위젯에서 pack 메소드가 호출되어 자동으로 창에 배치

entry.pack()

button = Button(win, text="계산", command=Cal)

button.pack()

#label 클래스를 이용한 Label 위젯 생성, label 변수에 할당

label = Label(win)

#label 위젯에서 pack 메소드 호출

label.pack()

win.mainloop()

**간단한 더하기 계산 프로그램(+예외처리)**

import tkinter as tk

#calculate 함수 정의

def calculate():

try:

#data1\_entry 및 data2\_entry에 수 입력, 더하기

result = int(data1\_entry.get()) + int(data2\_entry.get())

#계산결과

result\_label.config(text="결과: " + str(result))

#숫자가 아닌 걸 입력 시 Value Error 발생

except ValueError:

result\_label.config(text="숫자를 입력해주세요.")

#Tk 클래스 인스턴스 생성

root = tk.Tk()

root.title("간단한 계산기")

#위젯 생성

data1\_label = tk.Label(root, text="Data1:")

data1\_label.pack()

data1\_entry = tk.Entry(root)

data1\_entry.pack()

data2\_label = tk.Label(root, text="Data2:")

data2\_label.pack()

data2\_entry = tk.Entry(root)

data2\_entry.pack()

calculate\_button = tk.Button(root, text="계산", command=calculate)

calculate\_button.pack()

result\_label = tk.Label(root)

result\_label.pack()

root.mainloop()

**스터디 플래너 구현**

import tkinter as tk

from tkinter import messagebox

import matplotlib.pyplot as plt

# 전역 변수

event\_counter = 1

# 이벤트 추가 함수

def add\_event():

global event\_counter

event = event\_entry.get()

start\_time = start\_time\_entry.get()

end\_time = end\_time\_entry.get()

if event and start\_time and end\_time:

# 시작 시각과 종료 시각이 모두 입력된 경우

if start\_am\_pm.get() == "PM":

start\_time = add\_hours(start\_time, 12)

if end\_am\_pm.get() == "PM":

end\_time = add\_hours(end\_time, 12)

study\_time = calculate\_study\_time(start\_time, end\_time)

event\_info = f"{event} (시작시각: {start\_time}, 종료시각: {end\_time}, 공부시간: {study\_time})"

event\_listbox.insert(tk.END, f"{event\_counter}. {event\_info}")

event\_counter += 1

clear\_entries()

else:

messagebox.showwarning("경고", "이벤트, 시작시각, 종료시각을 모두 입력해주세요.")

# 시간에 시간 추가 함수

def add\_hours(time\_str, hours):

hour, minute = map(int, time\_str.split(":"))

total\_minutes = hour \* 60 + minute + (hours \* 60)

new\_hour = total\_minutes // 60

new\_minute = total\_minutes % 60

return f"{new\_hour:02d}:{new\_minute:02d}"

# 공부 시간 계산 함수

def calculate\_study\_time(start\_time, end\_time):

start\_hour, start\_minute = map(int, start\_time.split(":"))

end\_hour, end\_minute = map(int, end\_time.split(":"))

total\_minutes = (end\_hour \* 60 + end\_minute) - (start\_hour \* 60 + start\_minute)

hours = total\_minutes // 60

minutes = total\_minutes % 60

study\_time = f"{hours}.{minutes // 10} 시간"

return study\_time

# 엔트리 필드 초기화 함수

def clear\_entries():

event\_entry.delete(0, tk.END)

start\_time\_entry.delete(0, tk.END)

end\_time\_entry.delete(0, tk.END)

# 이벤트 삭제 함수

def delete\_event():

selected\_index = event\_listbox.curselection()

if selected\_index:

event\_listbox.delete(selected\_index)

global event\_counter

event\_counter = 1

events = event\_listbox.get(0, tk.END)

event\_listbox.delete(0, tk.END)

for event in events:

event\_listbox.insert(tk.END, f"{event\_counter}. {event}")

event\_counter += 1

# 모든 이벤트 삭제 함수

def clear\_all():

event\_listbox.delete(0, tk.END)

global event\_counter

event\_counter = 1

# 막대 그래프 표시 함수

def show\_bar\_chart():

study\_times = []

events = event\_listbox.get(0, tk.END)

for event in events:

study\_time = float(event.split("공부시간: ")[-1].split(" 시간")[0])

study\_times.append(study\_time)

labels = [event.split(". ")[1].split(" (")[0] for event in events]

plt.bar(labels, study\_times)

plt.xlabel('이벤트')

plt.ylabel('공부시간 (시간)')

plt.title('이벤트별 공부시간')

plt.show()

# Tkinter 윈도우 생성

root = tk.Tk()

# 이벤트 프레임 생성

event\_frame = tk.Frame(root)

event\_frame.pack(pady=10)

# 이벤트 레이블 및 엔트리 필드 생성

event\_label = tk.Label(event\_frame, text="이벤트:")

event\_label.grid(row=0, column=0)

event\_entry = tk.Entry(event\_frame)

event\_entry.grid(row=0, column=1)

# 시작시각 레이블 및 엔트리 필드 생성

start\_time\_label = tk.Label(event\_frame, text="시작시각:")

start\_time\_label.grid(row=1, column=0)

start\_time\_entry = tk.Entry(event\_frame)

start\_time\_entry.grid(row=1, column=1)

# 시작시각 AM/PM 라디오 버튼 생성

start\_am\_pm = tk.StringVar(value="AM")

start\_am\_radio = tk.Radiobutton(event\_frame, text="AM", variable=start\_am\_pm, value="AM")

start\_am\_radio.grid(row=1, column=2)

start\_pm\_radio = tk.Radiobutton(event\_frame, text="PM", variable=start\_am\_pm, value="PM")

start\_pm\_radio.grid(row=1, column=3)

# 종료시각 레이블 및 엔트리 필드 생성

end\_time\_label = tk.Label(event\_frame, text="종료시각:")

end\_time\_label.grid(row=2, column=0)

end\_time\_entry = tk.Entry(event\_frame)

end\_time\_entry.grid(row=2, column=1)

# 종료시각 AM/PM 라디오 버튼 생성

end\_am\_pm = tk.StringVar(value="AM")

end\_am\_radio = tk.Radiobutton(event\_frame, text="AM", variable=end\_am\_pm, value="AM")

end\_am\_radio.grid(row=2, column=2)

end\_pm\_radio = tk.Radiobutton(event\_frame, text="PM", variable=end\_am\_pm, value="PM")

end\_pm\_radio.grid(row=2, column=3)

# 이벤트 추가 버튼 생성

add\_button = tk.Button(event\_frame, text="추가", command=add\_event)

add\_button.grid(row=0, column=4, padx=10)

# 이벤트 리스트박스 생성

event\_listbox = tk.Listbox(root, width=50)

event\_listbox.pack(pady=10)

# 이벤트 삭제 버튼 생성

delete\_button = tk.Button(root, text="삭제", command=delete\_event)

delete\_button.pack()

# 모든 이벤트 삭제 버튼 생성

clear\_button = tk.Button(root, text="전체 삭제", command=clear\_all)

clear\_button.pack(pady=10)

# 막대 그래프 보기 버튼 생성

show\_chart\_button = tk.Button(root, text="그래프로 보기", command=show\_bar\_chart)

show\_chart\_button.pack(pady=10)

# Tkinter 이벤트 루프 실행

root.mainloop()