

## 5. 사용자 정의 위젯과 클래스: 객체를 염두에 두고 (10장)



순천향대학교 컴퓨터공학과  
이 상 정



## 학습 내용

- 지역 함수(local function)
  - 함수 안의 함수
- 위젯은 일종의 객체(object)
- 클래스(class)
  - 클래스는 객체를 생성하는 틀
  - 생성된 객체를 클래스의 인스턴스(instance)
  - 클래스는 객체의 행위를 정의하는 메서드(method)를 가지고 있음
  - 초기화 메서드(생성자)는 생성된 객체를 초기화
- 클래스를 사용하여 사용자 정의 위젯을 생성

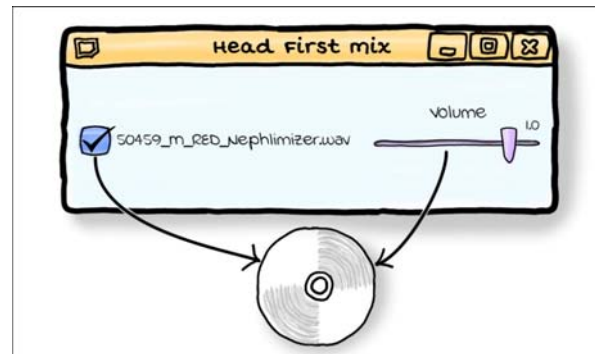
## 여러 음악을 연주하는 믹싱 프로그램

### □ 지금까지 한 노래를 연주

- 체크박스, 슬라이더 볼륨
- 이벤트 처리 코드

### □ 여러 노래를 동시에 연주하는 프로그램 작성 방법

1. 각 노래의 코드를 여러 번 복사
  - 코드 수정 시에 문제점 발생
2. 함수를 정의하여 코드를 재사용
  - 위젯을 생성하고 필요한 이벤트 처리기를 생성하는 함수
    - create\_gui()
  - 별도의 모듈 sound\_panel.py 작성



## 음악 믹싱 코드: 여러 음악 연주

```

from tkinter import *
from sound_panel import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")

mixer = pygame.mixer
mixer.init()

create_gui(app, mixer, "50459_M_RED_Nephlimizer.wav")
create_gui(app, mixer, "49119_M_RED_HardBouncer.wav")

def shutdown():
    mixer.stop()
    app.destroy()

app.protocol("WM_DELETE_WINDOW", shutdown)
app.mainloop()

```

## 음악 믹싱 코드: sound\_panel.py (1)

```

from tkinter import *
import pygame

def create_gui(app, mixer, sound_file):

    def track_toggle():          # 지역 함수
        if track_playing.get()==1:
            track.play(loops=-1)
        else:
            track.stop()

    def change_volume(v):       # 지역 함수
        # Scale 클래스 내에서 호출 시 인수 포함, 인수기술 필요
        track.set_volume(volume.get())

    track=mixer.Sound(sound_file)
    track_playing=IntVar()

```

## 음악 믹싱 코드: sound\_panel.py (2)

```

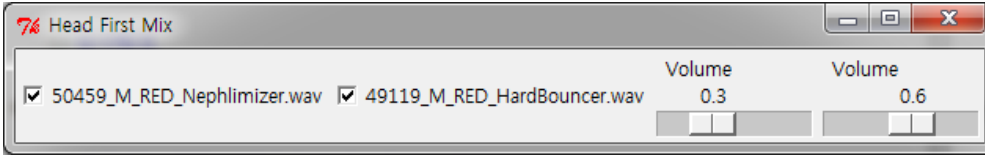
track_button=Checkbutton(app, variable=track_playing,
                        command=track_toggle, text=sound_file)
track_button.pack(side=LEFT)

volume=DoubleVar()
volume.set(track.get_volume())
volume_scale=Scale(variable=volume, from_=0.0, to=1.0,
                  resolution=0.1, command=change_volume,
                  label="Volume", orient=HORIZONTAL)
volume_scale.pack(side=RIGHT)

```



어떤 볼륨 슬라이더가  
어떤 노래에 해당하는지  
어떻게 알 수 있지?



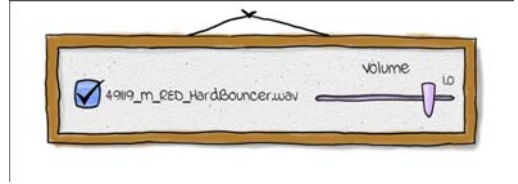
- 하나의 노래에 대해 체크 상자와 볼륨 슬라이더가 나란히 그룹화되어 배치
- 체크 상자와 슬라이더를 그룹화하는 새로운 위젯을 정의



# 프레임 위젯 (Frame Widget)

□ Tkinter는 프레임(frame)이라는 특별한 위젯을 제공하여 사용자 정의 위젯을 생성

- 프레임은 액자처럼 다른 위젯들을 포함



□ 새로운 형태의 프레임 사용 예

- 체크 상자와 슬라이더를 갖는 SoundPanel 프레임
- 새로운 형태의 객체

```
panel = SoundPanel(app, mixer, "50459_M_RED_Nephlimizer.wav")
panel.pack()

panel = SoundPanel(app, mixer, "49119_M_RED_HardBouncer.wav")
panel.pack()
```

# 클래스 (Class)

□ 클래스(class)는 객체를 생성하는 틀

- 클래스를 사용하여 사용자 정의 위젯을 생성
- 생성된 객체를 클래스의 인스턴스(instance)
  - 앞의 예에서 panel이 클래스 SoundPanel의 객체(인스턴스)
    - 레퍼런스 변수
- 클래스는 객체의 행위를 정의하는 메서드(method)를 가지고 있음

```
class SoundPanel(Frame):      # 클래스 이름은 SoundPanel
                              # 부모 클래스는 Frame
    .....
    def track_toggle(self):    # 메서드
        if self.track_playing.get() == 1: # self는 메서드를 자기 자신의 객체를 표시
            self.track.play(loops=-1)     # 즉 self는 메서드를 호출한 객체(위젯)을 표시
        else:
            self.track.stop()

    def change_volume(self, v): # 메서드
        self.track.set_volume(volume.get())
```

- 클래스는 생성된 객체를 초기화하는 초기화 메서드 필요
  - 생성자(constructor)라고도 함

```
class SoundPanel(Frame):    # 클래스 이름은 SoundPanel
                           # 부모 클래스는 Frame

    def __init__(self, app, mixer, sound_file): # 초기화 메서드
        Frame.__init__(self, app)             # 부모 클래스의 초기화 메서드 호출하여
                                                # Frame을 초기화
        .....
        .....

    def track_toggle(self):    # 메서드
        if self.track_playing.get()==1:      # self는 메서드를 자기 자신의 객체를 표시
            self.track.play(loops=-1)        # 즉 self는 메서드를 호출한 객체(위젯)을 표시
        else:
            self.track.stop()

    def change_volume(self, v):
        self.track.set_volume(volume.get())
```

순

```
from tkinter import *
from sound_panel import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")
mixer = pygame.mixer
mixer.init()

panel = SoundPanel(app, mixer, "50459_M_RED_Nephlimizer.wav") # 객체(인스턴스) 생성
panel.pack()
panel = SoundPanel(app, mixer, "49119_M_RED_HardBouncer.wav") # 객체(인스턴스) 생성
panel.pack()

def shutdown():
    mixer.stop()
    app.destroy()

app.protocol("WM_DELETE_WINDOW", shutdown)
app.mainloop()
```

## 음악 믹싱 코드: 사용자 정의 위젯 sound\_panel.py (1)

```

from tkinter import *
import pygame.mixer

class SoundPanel(Frame):
    def track_toggle(self):          # 메서드
        if self.track_playing.get()==1:
            self.track.play(loops=-1)
        else:
            self.track.stop()

    def change_volume(self, v):      # Scale 이벤트 처리 함수
                                    # 스케일 값 인수로 전달
        self.track.set_volume(self.volume.get())
        #self.track.set_volume(float(v))

```

## 음악 믹싱 코드: 사용자 정의 위젯 sound\_panel.py (2)

```

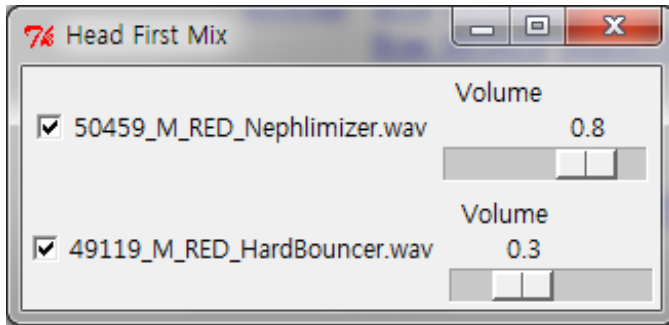
def __init__(self, app, mixer, sound_file):    # 초기화 메서드(생성자)
    Frame.__init__(self, app)                # 부모 메서드 초기화

    self.track=mixer.Sound(sound_file)

    self.track_playing=IntVar()
    track_button=Checkbutton(self, variable=self.track_playing,
                              command=self.track_toggle, text=sound_file)
    track_button.pack(side=LEFT)

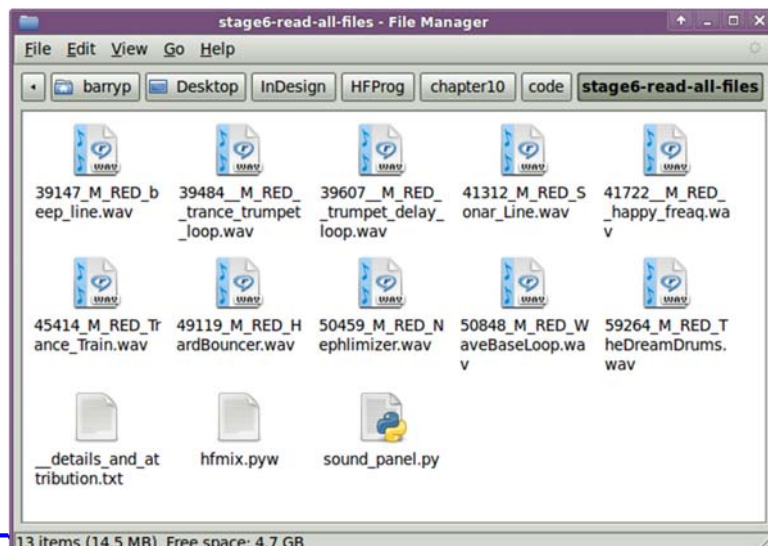
    self.volume=DoubleVar()
    self.volume.set(self.track.get_volume())
    volume_scale=Scale(self, variable=self.volume, from_=0.0, to=1.0,
                       resolution=0.1, command=self.change_volume,
                       label="Volume", orient=HORIZONTAL)
    volume_scale.pack(side=RIGHT)

```



## 디렉터리(폴더)의 음악을 믹싱

- 폴더(디렉터리)에 있는 음악을 동적으로 로드(적재)하여 믹싱
  - os 모듈 임포트하여 현재 폴더의 파일 이름 가져옴





## 음악 믹싱 코드: 폴더 파일 동적 적재 (1)

```

from tkinter import *
from sound_panel import *
import pygame.mixer
import os

app=Tk()
app.title("Head First Mix")
mixer = pygame.mixer
mixer.init()

dirList=os.listdir(".")
for fname in dirList:
    if fname.endswith(".wav"):
        panel = SoundPanel(app, mixer, fname)
        panel.pack()

```

# 현재 폴더의 모든 파일 이름을 가져옴  
# 각 파일의 이름을 가져옴  
# .wav로 끝나면  
# SoundPanel 위젯 생성  
# GUI에추가

## 음악 믹싱 코드: 폴더 파일 동적 적재 (2)

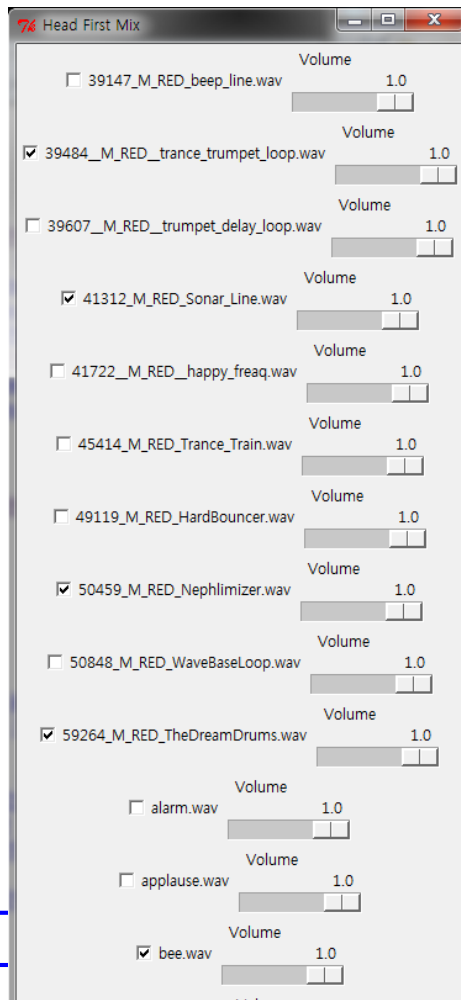
```

def shutdown():
    mixer.stop()
    app.destroy()

app.protocol("WM_DELETE_WINDOW", shutdown)

app.mainloop()

```



# 시험주행



# 프로그래밍 도구상자 (1)

## □ 프로그래밍 도구

- 지역함수는 다른 함수 안에서 정의
- 객체지향(object-oriented)은 소프트웨어 객체를 사용하여 복잡성을 다루는 방법
- 클래스는 객체를 생성하는 틀
- 생성된 객체를 클래스의 인스턴스(instance)라고 함
- 초기화 메서드는 생성된 객체에게 무엇을 해야 할 것인지를 알려줌
- 초기화 메서드는 생성자(constructor)라고도 함
- 모든 객체가 GUI 객체는 아님
- 위젯은 일종의 객체



## 프로그래밍 도구상자 (2)

### □ 파이썬 도구

- `Frame()` - tkinter의 프레임은 다른 위젯을 포함하는 위젯으로서 위젯들을 그룹화
- `class` - 클래스 정의가 시작되는 것을 알려주는 키워드
- `__init__()` - 객체가 생성될 때 자동으로 호출되는 초기화 메서드
- `self` - 메서드는 현재 객체를 가리키는 `self`라는 특별한 변수를 가짐
- `self`를 변수 이름 앞에 붙이면 이 변수는 현재 객체에 속함 것임

## 과제

1. 최종 음악 믹싱 코드를 작성하고 실행
2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
  - 프로그램 설명
  - 프로그램 소스
  - 실행 결과

