프로젝트 설계 보고서

20315179

김다은

>>모듈의 구현

기능을 다루는 모듈과 UI를 구현하는 모듈을 별도로 분리하였다.

>>각 모듈의 기능적 구분

MiniProject.py : UI부분을 구현한 파일이다.

--------------------------------------------------------------

from tkinter import \*

window = Tk()

window.title("MiniProject made by DaEun\_20135179")

#top frame

top\_frame = Frame(window)

top\_frame.grid(row=0, column=0, columnspan=3, sticky =N)

#name label

Label(top\_frame, text="name: ").grid(row=0,column=0,sticky=W)

entry\_name = Entry(top\_frame, width=20, bg = "green")

entry\_name.grid(row=0, column=1, sticky=W)

#grade label

Label(top\_frame, text="grade: ").grid(row=0,column=2,sticky=W)

entry\_grade = Entry(top\_frame, width=15, bg ="light green")

entry\_grade.grid(row=0, column=3, sticky=E)

#number label

Label(top\_frame, text="number: ").grid(row=1, column=2,sticky=W)

entry\_num = Entry(top\_frame, width=15, bg = "light green")

entry\_num.grid(row=1, column=3, sticky=E)

#file name label\_1

Label(top\_frame, text="file name: ").grid(row=2, column=2,sticky=W)

entry\_output = Entry(top\_frame, width=15, bg = "yellow")

entry\_output.grid(row=2, column=3, sticky=E)

#file name label\_2

Label(top\_frame, text="file name: ").grid(row=3, column=2,sticky=W)

entry\_input = Entry(top\_frame, width=15, bg = "yellow")

entry\_input.grid(row=3, column=3, sticky=E)

#middle frame

middle\_frame = Frame(window)

middle\_frame.grid(row=5, column=0, sticky=S)

#result frame

result\_frame = Frame(window)

result\_frame.grid(row=6, column=0, sticky = S)

output\_data = Text(result\_frame, width = 75, height = 10, wrap = WORD, background = "pink")

output\_data.grid(row = 5, column = 0, sticky = W)

#bottom

output\_state = Text(result\_frame, width = 75, height=1, wrap = WORD, background = "red")

output\_state.grid(row = 6, column =0, sticky = W)

--------------------------------------------------------------

MiniProjectFunc.py : 해당 기능을 구현한 파일이다.

--------------------------------------------------------------

from MiniProject import \*

import pickle

book=[ ]

# check state frame

def check\_state(msg):

output\_state.delete(0.0, END)

output\_state.insert(END, msg)

# check result frame

current\_state = 'number order'

def check(x):

global book

global current\_state

output\_data.delete(0.0, END)

if x == 'number order':

book.sort(key = lambda i:i[0])

check\_state('Sorted by number')

elif x == 'name order':

book.sort(key = lambda i:i[1])

check\_state('Sorted by name')

elif x == 'descending':

book.sort(key = lambda i:i[2])

check\_state('Sorted by score in descending sort order')

elif x == 'ascending':

book.sort(key = lambda i:i[2], reverse=True)

check\_state('Sorted by score in ascending sort order')

for data in book:

output\_data.insert(END, data)

output\_data.insert(END, '\n')

current\_state = x

def click\_func(x):

global book

if x == 'add':

try:

book.append([int(entry\_num.get()), entry\_name.get(), float(entry\_grade.get())])

check(current\_state)

check\_state('Data is added')

except ValueError:

refresh\_state('Invalid')

elif x == 'delete':

t = 0

try:

for i in book:

if i[0] == int(entry\_num.get()):

del book[t]

check(current\_state)

check\_state('Data is deleted')

break

t += 1

except ValueError:

check\_state('Invalid')

elif x == 'save':

try:

fout = open(entry\_output.get(), 'wb')

pickle.dump(book, fout)

fout.close()

check(current\_state)

check\_state('Data is saved')

except FileNotFoundError:

check\_state('Invalid')

elif x == 'open':

try:

fin = open(entry\_input.get(), 'rb')

book = pickle.load(fin)

check(current\_state)

check\_state('Data is loaded')

except FileNotFoundError:

check\_state('Invalid')

pad\_list=['add','delete','save','open']

c=0

for func in pad\_list:

def click(x=func):

click\_func(x)

Button(top\_frame, text=func, width=5, command=click).grid(row=c, column=5, sticky=E)

c+=1

# Sorting buttons

def click\_sort(x):

check(x)

num\_pad\_list = ['number order','name order','descending','ascending']

r=0

for btn in num\_pad\_list:

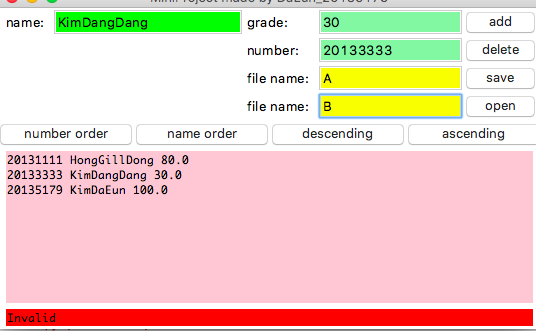
Button(middle\_frame, text=btn, width=12, command =click).grid(row=4, column=r)

r+=1

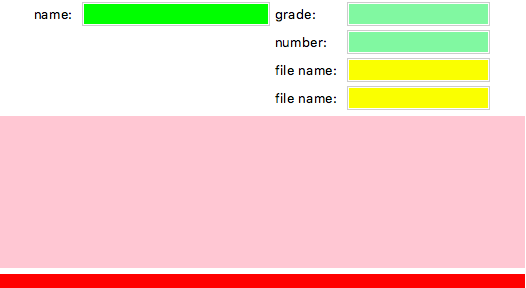
window.mainloop()

--------------------------------------------------------------

>>테스트 케이스



>>UI 설계



frame을 top, middle result 이렇게 세 부분으로 나누어 설계함.

top\_frame의 경우, columnspan을 3개로 나누어 정보를 입력할 수 있는 Label과 버튼을 배치함.

middle\_frame의 경우, 정렬을 해주는 버튼을 배치함.

result\_frame의 경우, 중간에 입력한 정보가 보일 수 있는 text상자와 입력 상태가 올바른지 알려주는 text 상자를 배치함.

>> 데이터 모델

* 각 클래스에 포함될 속성 : 버튼을 누르면 버튼 마다 해당되는 기능을 실행.
* 리스트를 이용하여 버튼을 생성하고 함수를 통해 기능을 입힘.