

1. 선택 프로젝트명: 자동 판매기 프로그램(3강 계산 실습 6)
2. 프로젝트 설명
  - 자동 판매기는 사용자로부터 투입한 돈과 물건값을 입력받는다. (물건값: 100원 단위)
  - 프로그램: 잔돈을 계산하여 출력(동전: 500원, 100원짜리만 있음)
3. 프로젝트 업그레이드 내용

(추가한 코드량이 많기에, **기존의 코드**를 빨간색으로 표시하였습니다. 양해 부탁드립니다.)

step 0) tkinter 모듈 import & 전역변수 설정

```
from tkinter import *
# 0. 전역 변수 설정
bill = {1: 0, 2: 0, 3: 0, 4: 0} # 주문하는 메뉴(Key) & 주문수량(Value)
# -> 수량은 일단 0으로 초기화된 상태

total_money = 0 # 결제금액
change = 0 # 거스름돈
```

step 1) 각 단계별 창 구성하기

- 동일한 창을 여러 번 활용하는 경우들이 존재하므로, 각각의 창(window)을 만드는 함수를 정의한 다음, 필요 시 호출하여 사용한다.

① 메인화면: main\_window( )

→ 레이블로 안내 문구들을 표시하고, 지금까지의 주문내역을 토대로 결제예정금액을 표시해 준다. 버튼을 이용하여 원하는 업무(주문,취소,결제)를 선택할 수 있다.

# 이벤트 처리 함수

```
def ordering(): # 주문하기
    main.destroy() # 초기화면 파괴
    order_window() # 주문하기 창 생성

def canceling(): # 취소하기
    main.destroy() # 초기화면파괴
    cancel_window() # 취소하기 창 생성

def confirming(): # 결제하기(주문확정)
    main.destroy() # 초기화면 파괴
    confirm_window() # 주문확정 창 생성
```

# 화면 인터페이스 구성

```
main = Tk() # 메인 윈도우 생성
main.title("메인화면")
main.geometry("700x700") # 메인 화면 창 크기 조절하기
main.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정 가능

# 라벨 생성하기(info,total)
info = Label(main, text="Happy Drink:", font=("Consolas", 20))
info.grid(row=0, column=0, columnspan=4)

total_label = Label(main, text="결제예정금액: " + str(total_money) + "원",
                    font=("Consolas", 20))
total_label.grid(row=1, column=0, columnspan=4)

work = Label(main, text="원하는 업무를 선택해 주세요", font=("Consolas", 20))
work.grid(row=2, column=0, columnspan=4)

# 버튼 생성하기
order_button = Button(main, text="1. 주문하기",
                     command=ordering, font=("Consolas", 20))
order_button.grid(row=3, column=0, columnspan=4)
```

```
cancel_button = Button(main, text="2. 취소하기",
                        command=canceling, font=("Consolas", 20))
cancel_button.grid(row=4, column=0, columnspan=4)
confirm_button = Button(main, text="3. 결제하기",
                        command=confirming, font=("Consolas", 20))
confirm_button.grid(row=5, column=0, columnspan=4)

main.mainloop()
```

# 실행결과

메인화면

Happy Drink:) — info  
 결제예정금액: 0원 — total\_label  
 원하는 업무를 선택해 주세요 — work

1. 주문하기	— order_button
2. 취소하기	— cancel_button
3. 결제하기	— confirm_button

② 주문화면: order\_window( )

→ 주문할 메뉴 번호와 수량을 입력하고 Enter 버튼을 누르면 calc\_plus( ) 함수가 작동하여 주문내역의 변경사항이 딕셔너리에 저장(반영)되고, 이를 토대로 총액을 계산하여 total\_money에 저장한다. total\_money는 total\_entry 창에 표시되고, 주문 버튼을 누르면 주문확정 페이지로 이동한다.

(한 번에 하나의 메뉴만 주문할 수 있다. 추가로 주문을 원하는 경우 주문확정 페이지에서 “no” 버튼을 누르고 메인화면으로 돌아간 다음, 다시 주문하기 업무를 선택하여 주문사항을 추가한다.)

# 이벤트 처리함수

```
def order_confirming(): # 주문확정
    order.destroy() # 기존에 띄웠던 창 파괴
    confirm_window() # 주문확정 창 생성
def calc_plus(): # 결제금액 계산
    global bill
    global total_money
    menu = int(menu_entry.get())
    count = int(count_entry.get())
    bill[menu] += count
    total_money = bill[1] * 1000 + bill[2] * 900 + bill[3] * 1000 +
        bill[4] * 800
    total_entry.insert(0, str(total_money))
```

# 화면 인터페이스 구성

```
order = Tk() # 주문 윈도우 생성
order.title("주문화면")
order.geometry("700x700") # 창 크기 조절하기
order.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

photo = PhotoImage(file='menu.gif') # 메뉴판 이미지 출력
imageLabel = Label(order, image=photo)
imageLabel.grid(row=0, column=0, rowspan=4, columnspan=4)

menu_label = Label(order, text="주문할 메뉴의 번호를 입력해 주세요: ",
                    font=("Consolas", 20))
menu_label.grid(row=4, column=0, columnspan=3)
menu_entry = Entry(order)
menu_entry.grid(row=4, column=3)

count_label = Label(order, text="수량을 입력해 주세요: ", font=("Consolas", 20))
count_label.grid(row=5, column=0, columnspan=3)
count_entry = Entry(order)
count_entry.grid(row=5, column=3)

enter_button = Button(order, text="Enter", command=calc_plus,
                      font=("Consolas", 15))
enter_button.grid(row=5, column=4)

total_label = Label(order, text="결제예정금액: ", font=("Consolas", 20))
total_label.grid(row=6, column=0, columnspan=3)
total_entry = Entry(order)
total_entry.grid(row=6, column=3, columnspan=2)
```

```

order_button = Button(order, text="주문", command= order_confirming,
font=("Consolas", 15))
order_button.grid(row=7, column=4)

order.mainloop()

```

# 실행결과



③ 취소화면: cancel\_window( )

→ 현재의 주문내역을 표시해준다.(각각의 줄을 라벨로 표시하고, 크기 조정을 통해 하나의 라벨처럼 보이게 하였다.)

→ 취소할 메뉴 번호와 수량을 입력하고 Enter 버튼을 누르면 calc\_minus( ) 함수가 작동하여 주문내역의 변경사항이 딕셔너리에 저장되고, 이를 토대로 총액을 계산하여 total\_money에 저장한다. total\_money는 total\_entry 창에 표시되고, 취소 버튼을 누르면 주문확정 페이지로 이동한다.

(한 번에 하나의 메뉴만 취소할 수 있다. 추가로 취소를 원하는 경우 주문확정 페이지에서 “no” 버튼을 누르고 메인화면으로 돌아간 다음, 다시 취소하기 업무를 선택하여야 한다.)

# 이벤트 처리함수

```
def cancel_confirming(): # 취소
    cancel.destroy() # 기존에 띄웠던 창 파괴
    confirm_window() # 주문확정 창 생성하기
def calc_minus(): # 결제금액 계산
    global bill # 함수 내에서 각각의 주문 수량 변경하기
    global total_money
    menu = int(menu_entry.get())
    count = int(cancel_entry.get())

    if count > bill[menu]: # 취소수량 > 주문수량인 경우 -> 입력메뉴/수량 초기화
        menu_entry.delete(0,END)
        cancel_entry.delete(0,END)
    else:
        bill[menu] -= count
        total_money = bill[1] * 1000 + bill[2] * 900 + bill[3] * 1000
                    + bill[4] * 800
        total_entry.insert(0, str(total_money))
```

# 화면 인터페이스 구성

```
cancel = Tk() # 취소 윈도우 생성
cancel.title("취소화면")
cancel.geometry("800x800") # 메인 화면 창 크기 조절하기
cancel.resizable(width=1, height=1) # 가로,세로 둘 다 크기

# 현재 주문 내역 표시
bill_label = Label(cancel, text="주문내역", font=("Consolas", 35),
                    bg="Cornflower Blue", width=21)
bill_label.grid(row=0, column=0, columnspan=4)

# 메뉴 표시 레이블 생성
b_list = ["1. 커피음료", "2. 초코우유", "3. 탄산음료", "4. 박카스"]
row_idx = 1
for b_text in b_list:
    Label(cancel, text=b_text, font=("Consolas", 25),
          bg="Cornflower Blue", width=15).grid(row=row_idx, column=0,
          columnspan=3)
    row_idx += 1
```

```

# 수량 표시 레이블 셋팅
c_list = list(bill.values())
row_index = 1
for c_text in c_list:
    Label(cancel, text=c_text, font=("Consolas", 25),
           bq="Cornflower Blue", width=15).grid(row=row_index,
           column=3)
    row_index += 1

# 취소 처리하기
menu_label = Label(cancel, text="취소할 메뉴의 번호를 입력해 주세요: ",
                    font=("Consolas", 20))
menu_label.grid(row=5, column=0, columnspan=4)
menu_entry = Entry(cancel)
menu_entry.grid(row=5, column=4)

cancel_label = Label(cancel, text="수량을 입력해 주세요: ",
                     font=("Consolas", 20))
cancel_label.grid(row=6, column=0, columnspan=4)
cancel_entry = Entry(cancel)
cancel_entry.grid(row=6, column=4)

enter_button = Button(cancel, text="Enter", command=calc_minus,
                      font=("Consolas", 15))
enter_button.grid(row=6, column=5)

total_label = Label(cancel, text="결제예정금액: ", font=("Consolas", 20))
total_label.grid(row=7, column=0, columnspan=4)
total_entry = Entry(cancel)
total_entry.grid(row=7, column=4, columnspan=2)

cancel_button = Button(cancel, text="취소", command=cancel_confirming,
                       font=("Consolas", 15))
cancel_button.grid(row=8, column=5)

cancel.mainloop()

```

# 실행결과



#### ④ 결제하기

→ 4-1) 주문확정: confirm\_window( )

주문하기, 취소하기 창에서 각각 주문, 취소 버튼을 누르면 실행됨  
주문내역을 라벨로 표시해주고, 결제예정금액을 표시해준다.

주문확정을 묻는 안내멘트를 라벨로 표시한다.

주문을 확정하려면 YES 버튼을 누른다.(결제하기 창으로 이동)

변경이 필요한 경우 NO 버튼을 누른다.(메인화면으로 이동)

# 이벤트 처리함수

```
def paying():
    confirm.destroy() # 현재 창 파괴
    pay_window() # 결제하기 창으로 이동

def main():
    confirm.destroy() # 현재 창 파괴
    main_window() # 메인화면으로 이동
```

# 화면 인터페이스 구성

```
confirm = Tk() # 주문확정 창 생성
confirm.title("주문확정화면")
confirm.geometry("800x800") # 메인 화면 창 크기 조절하기
confirm.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조절 가능

# 주문내역 출력
bill_label = Label(confirm, text="주문내역", font=("Consolas", 35),
                    bg="Cornflower Blue", width=21)
bill_label.grid(row=0, column=0, columnspan=4)

# 메뉴 표시 레이블 생성
b_list = ["1. 커피음료", "2. 초코우유", "3. 탄산음료", "4. 박카스"]
row_idx = 1
for b_text in b_list:
    Label(confirm, text=b_text, font=("Consolas", 25),
           bg="Cornflower Blue", width=15).grid(row=row_idx,
           column=0, columnspan=3)
    row_idx += 1

# 수량 표시 레이블 생성
c_list = list(bill.values())
row_index = 1
for c_text in c_list:
    Label(confirm, text=c_text, font=("Consolas", 25),
           bg="Cornflower Blue", width=15).grid(row=row_index,
           column=3)
    row_index += 1

# 주문확정 처리하기
total_label = Label(confirm, text="결제예정금액: " + str(total money) + " 원",
                    font=("Consolas", 20))
total_label.grid(row=5, column=0, columnspan=4)

confirm_label = Label(confirm, text="주문을 확정하시겠습니까?",
                      font=("Consolas", 20))
confirm_label.grid(row=6, column=0, columnspan=4)
```

```

yes_button = Button(confirm, text="YES", command=paying,
                    font=("Consolas", 15))
yes_button.grid(row=7, column=3)

no_button = Button(confirm, text="NO", command=main,
                  font=("Consolas", 15))
no_button.grid(row=7, column=4)

confirm.mainloop()

```

# 실행화면





→ 4-2) 결제하기: pay\_window( )

결제예정금액을 label로 표시

결제 금액을 입력받아 거스름돈(change) 계산

change > 0 => 거스름돈 창으로 이동

change < 0 => 잔액부족 창으로 이동

change == 0 => 결제완료 창으로 이동

# 이벤트 처리함수

```
def calc_pay(): # 거스름돈 계산 -> 처리
    global total_money
    global change
    pay_money = int(pay_entry.get()) # 지역변수
    change = pay_money - total_money

    if change > 0:
        pay.destroy() # 창 파괴
        coin_window() # 거스름돈 창으로 이동
    elif change < 0:
        total_money = -(change) # 지불한 금액 처리-> 결제금액 갱신
        pay.destroy() # 현재 창 파괴
        error_window() # 에러 창 띄우기
    else: # 결제완료
        pay.destroy() # 창 파괴
        end_window() # 결제완료 창
```

# 화면 인터페이스 구성

```
pay = Tk()
pay.title("결제화면")
pay.geometry("500x500") # 윈도우 크기 조절하기
pay.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

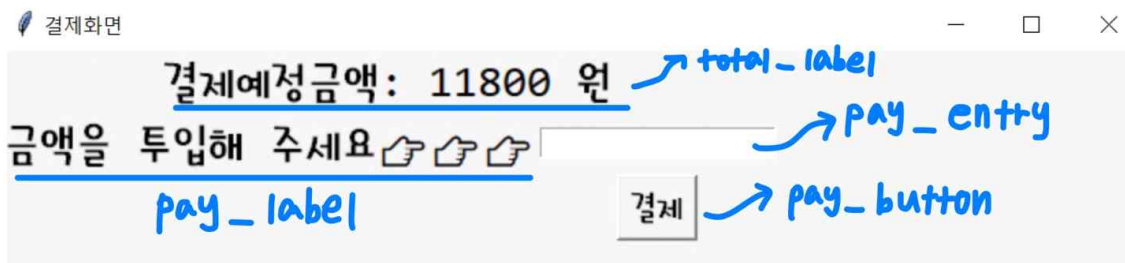
total_label = Label(pay, text="결제예정금액: " + str(total_money) + " 원",
                    font=("Consolas", 20))
total_label.grid(row=0, column=0, columnspan=4)

pay_label = Label(pay, text="금액을 투입해 주세요 □□□", font=("Consolas", 20))
pay_label.grid(row=1, column=0, columnspan=3)
pay_entry = Entry(pay)
pay_entry.grid(row=1, column=3)

pay_button = Button(pay, text="결제", command=calc_pay,
                   font=("Consolas", 15))
pay_button.grid(row=2, column=3)

pay.mainloop()
```

# 실행화면



→ 4-3) 후처리

i. 거스름돈 지불: coin\_window( )

거스름돈 계산 => 기존코드

거스름돈 표시: 거스름돈 총액, 500원짜리/100원짜리 개수를

각각의 라벨로 표시

종료문구를 라벨로 출력

종료버튼 -> 창 파괴

# 거스름돈 계산

```
global change
coin_500s = change // 500
new_change = change % 500 # 거스름돈 계산을 위한 임시변수
coin_100s = new_change // 100
```

# 이벤트 처리함수

```
def reset1():
    coin.destroy() # 현재 창 파괴
```

# 화면 인터페이스 구성

```
coin = Tk() # 거스름돈 지급을 위한 윈도우 생성
coin.title("거스름돈 화면")
coin.geometry("500x500") # 메인 화면 창 크기 조절하기
coin.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

coin_label = Label(coin, text="거스름돈: " + str(change) + " 원",
                    font=("Consolas", 20))
coin_label.grid(row=0, column=0, columnspan=4)

c500_label = Label(coin, text="500원: " + str(coin_500s) + " 개",
                    font=("Consolas", 20))
c500_label.grid(row=1, column=0, columnspan=4)

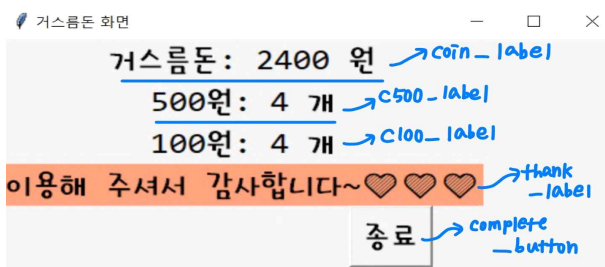
c100_label = Label(coin, text="100원: " + str(coin_100s) + " 개",
                    font=("Consolas", 20))
c100_label.grid(row=2, column=0, columnspan=4)

thank_label = Label(coin, text="이용해 주셔서 감사합니다~□□□",
                    bg="light salmon", font=("Consolas", 20))
thank_label.grid(row=3, column=0, columnspan=4)

complete_button = Button(coin, text="종료", command=reset1,
                           font=("Consolas", 20))
complete_button.grid(row=4, column=3)

coin.mainloop()
```

# 실행화면



ii. 잔액부족 메시지 띄우기: error\_window( )

부족한 금액 표시 -> 추가 결제(결제하기 창으로 다시 이동) -> 결제 -> ...

# 이벤트 처리함수

```
def go_back():
    error.destroy()
    pay_window() # 결제화면으로 되돌아가기
```

# 화면 인터페이스 구성

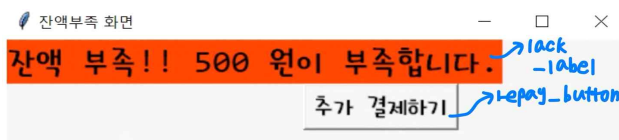
```
error = Tk() # 잔액부족 윈도우 생성
error.title("잔액부족 화면")
error.geometry("500x500") # 메인 화면 창 크기 조절하기
error.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

lack_label = Label(error, text="잔액 부족!! " + str(total money) +
                    " 원이 부족합니다.", font=("Consolas", 20), bg="OrangeRed")
lack_label.grid(row=0, column=0, columnspan=4)

repay_button = Button(error, text="추가 결제하기", command=go_back,
                      font=("Consolas", 15))
repay_button.grid(row=1, column=3)

error.mainloop()
```

# 실행화면



iii. 결제완료: end\_window()

# 이벤트 처리함수

```
def reset2():
    end.destroy() # 창 파괴하기
```

# 화면 인터페이스 구성

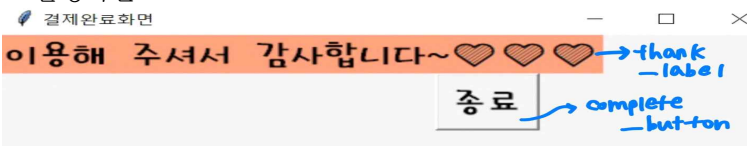
```
end = Tk() # 결제완료 창 생성
end.title("결제완료화면")
end.geometry("500x500") # 메인 화면 창 크기 조절하기
end.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

thank_label = Label(end, text="이용해 주셔서 감사합니다~□□□",
                    bg="light salmon", font=("Consolas", 20))
thank_label.grid(row=0, column=0, columnspan=4)

complete_button = Button(end, text="종료", command=reset2,
                          font=("Consolas", 20))
complete_button.grid(row=1, column=3)

end.mainloop()
```

# 실행화면



step 2) main\_window() 호출: 제일 처음에 메인화면 띄우기 위함

#### 4. 전체 코드

```
from tkinter import *
# 0. 전역 변수 설정
bill = {1: 0, 2: 0, 3: 0, 4: 0} # 주문하는 메뉴(Key) & 주문수량(Value)
                                # -> 수량은 일단 0으로 초기화된 상태
total_money = 0 # 결제금액
change = 0 # 거스름돈

# 1. 각 단계별 창 구성하기
# 동일한 창을 여러 번 활용하는 경우들이 존재하므로, 각각의 창 화면을 만드는 함수를 정의
-> 필요 시 호출하여 사용한다.
# 1-1. 초기화면(메인화면)
def main_window():
    # 이벤트 처리 함수
    def ordering(): # 주문하기 버튼을 눌렀을 때 실행
        main.destroy() # 초기화면 파괴
        order_window() # 주문하기 창 생성
    def canceling():
        main.destroy() # 초기화면파괴
        cancel_window() # 취소하기 창 생성
    def confirming():
        main.destroy() # 초기화면 파괴
        confirm_window() # 주문확정 창 생성
    # 메인 화면 윈도우 객체(틀) 생성
    main = Tk()
    main.title("메인화면")
    main.geometry("700x700") # 메인 화면 창 크기 조절하기
    main.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정 가능
    # 라벨 생성하기(info,total)
    info = Label(main, text="Happy Drink:)", font=("Consolas", 20))
    info.grid(row=0, column=0, columnspan=4)
    total_label = Label(main, text="결제예정금액: " + str(total_money) + "원",
font=("Consolas", 20))
    total_label.grid(row=1, column=0, columnspan=4)
    work = Label(main, text="원하는 업무를 선택해 주세요", font=("Consolas", 20))
    work.grid(row=2, column=0, columnspan=4)
    # 버튼 생성하기
    order_button = Button(main, text="1. 주문하기",
command=ordering,font=("Consolas", 20))
    order_button.grid(row=3, column=0, columnspan=4)
    cancel_button = Button(main, text="2. 취소하기",
```

```

command=canceling,font=("Consolas", 20))
    cancel_button.grid(row=4, column=0, columnspan=4)
    confirm_button = Button(main, text="3.      결제하기",
command=confirming,font=("Consolas", 20))
    confirm_button.grid(row=5, column=0, columnspan=4)

    main.mainloop()

# 1-2. 주문 화면 구성
def order_window():
    # 이벤트 처리 함수
    def order_confirming():
        order.destroy() # 기존에 떠있던 창 파괴
        confirm_window() # 주문확정 창 생성하기
    def calc_plus():
        global bill # 함수 내에서 각각의 주문 수량 변경하기
        global total_money
        menu = int(menu_entry.get())
        count = int(count_entry.get())
        bill[menu] += count
        total_money = bill[1] * 1000 + bill[2] * 900 + bill[3] * 1000 + bill[4] * 800
        total_entry.insert(0, str(total_money))

    # 인터페이스 구성하기
    order = Tk()
    order.title("주문화면")
    order.geometry("700x700") # 메인 화면 창 크기 조절하기
    order.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

    photo = PhotoImage(file='menu.gif')
    imageLabel = Label(order, image=photo)
    imageLabel.grid(row=0, column=0, rowspan=4, columnspan=4)

    menu_label = Label(order, text="주문할 메뉴의 번호를 입력해 주세요: ",
font=("Consolas", 20))
    menu_label.grid(row=4, column=0, columnspan=3)
    menu_entry = Entry(order)
    menu_entry.grid(row=4, column=3)

    count_label = Label(order, text="수량을 입력해 주세요: ", font=("Consolas", 20))
    count_label.grid(row=5, column=0, columnspan=3)

```

```

count_entry = Entry(order)
count_entry.grid(row=5, column=3)

enter_button = Button(order, text="Enter", command=calc_plus, font=("Consolas",
15))
enter_button.grid(row=5, column=4)

total_label = Label(order, text="결제예정금액: ", font=("Consolas", 20))
total_label.grid(row=6, column=0, columnspan=3)
total_entry = Entry(order)
total_entry.grid(row=6, column=3, columnspan=2)

order_button = Button(order, text="주문", command= order_confirming,
font=("Consolas", 15))
order_button.grid(row=7, column=4)

order.mainloop()

# 1-3. 취소 화면 구성
def cancel_window():
    # 이벤트 처리 함수
    def cancel_confirming():
        cancel.destroy() # 기존에 떠있던 창 파괴
        confirm_window() # 주문확정 창 생성하기
    def calc_minus():
        global bill # 함수 내에서 각각의 주문 수량 변경하기
        global total_money
        menu = int(menu_entry.get())
        count = int(cancel_entry.get())
        if count > bill[menu]: # 취소수량 > 주문수량인 경우
            menu_entry.delete(0,END)
            cancel_entry.delete(0,END)
        else:
            bill[menu] -= count
            total_money = bill[1] * 1000 + bill[2] * 900 + bill[3] * 1000 + bill[4] *
800
            total_entry.insert(0, str(total_money))

cancel = Tk()
cancel.title("취소화면")
cancel.geometry("800x800") # 메인 화면 창 크기 조절하기

```

```

cancel.resizable(width=1, height=1) # 가로,세로 둘 다 크기

# 현재 주문내역 표시
bill_label = Label(cancel, text="주문내역", font=("Consolas", 35), bg="Cornflower
Blue", width=21)
bill_label.grid(row=0, column=0, columnspan=4)

b_list = ["1. 커피음료", "2. 초코우유", "3. 탄산음료", "4. 박카스 "]
row_idx = 1
for b_text in b_list:
    Label(cancel, text=b_text, font=("Consolas", 25), bg="Cornflower Blue",
width=15).grid(row=row_idx, column=0,

        columnspan=3)
    row_idx += 1

c_list = list(bill.values())
row_index = 1
for c_text in c_list:
    Label(cancel, text=c_text, font=("Consolas", 25), bg="Cornflower Blue",
width=15).grid(row=row_index, column=3)
    row_index += 1

# 취소 처리하기
menu_label = Label(cancel, text="취소할 메뉴의 번호를 입력해 주세요: ",
font=("Consolas", 20))
menu_label.grid(row=5, column=0, columnspan=4)
menu_entry = Entry(cancel)
menu_entry.grid(row=5, column=4)

cancel_label = Label(cancel, text="수량을 입력해 주세요: ", font=("Consolas", 20))
cancel_label.grid(row=6, column=0, columnspan=4)
cancel_entry = Entry(cancel)
cancel_entry.grid(row=6, column=4)

enter_button = Button(cancel, text="Enter", command=calc_minus,
font=("Consolas", 15))
enter_button.grid(row=6, column=5)

total_label = Label(cancel, text="결제예정금액: ", font=("Consolas", 20))
total_label.grid(row=7, column=0, columnspan=4)

```

```

total_entry = Entry(cancel)
total_entry.grid(row=7, column=4, columnspan=2)

cancel_button = Button(cancel, text="취소", command=cancel_confirming,
font=("Consolas", 15))
cancel_button.grid(row=8, column=5)

cancel.mainloop()

# 1-4. 결제 화면 구성
# 1-4-1. 주문확정 화면 구성
def confirm_window():
    def paying():
        confirm.destroy() # 현재 창 파괴
        pay_window() # 결제하기 창으로 이동

    def main():
        confirm.destroy() # 현재 창 파괴
        main_window() # 메인화면으로 이동

    confirm = Tk()
    confirm.title("주문확정화면")
    confirm.geometry("800x800") # 메인 화면 창 크기 조절하기
    confirm.resizable(width=1, height=1) # 가로,세로 둘 다 크기
    # 주문내역 출력
    bill_label = Label(confirm, text="주문내역", font=("Consolas", 35), bg="Cornflower
Blue", width=21)
    bill_label.grid(row=0, column=0, columnspan=4)

    b_list = ["1. 커피음료", "2. 초코우유", "3. 탄산음료", "4. 박카스 "]
    row_idx = 1
    for b_text in b_list:
        Label(confirm, text=b_text, font=("Consolas", 25), bg="Cornflower Blue",
width=15).grid(row=row_idx, column=0,columnspan=3)
        row_idx += 1

    c_list = list(bill.values())
    row_index = 1
    for c_text in c_list:
        Label(confirm, text=c_text, font=("Consolas", 25), bg="Cornflower Blue",
width=15).grid(row=row_index, column=3)

```



```

    row_index += 1

    # 주문확정 처리하기
    total_label = Label(confirm, text="결제예정금액: " + str(total_money) + " 원",
font=("Consolas", 20))
    total_label.grid(row=5, column=0, columnspan=4)

    confirm_label = Label(confirm, text="주문을 확정하시겠습니까?", font=("Consolas",
20))
    confirm_label.grid(row=6, column=0, columnspan=4)

    yes_button = Button(confirm, text="YES", command=paying, font=("Consolas", 15))
    yes_button.grid(row=7, column=3)

    no_button = Button(confirm, text="NO", command=main, font=("Consolas", 15))
    no_button.grid(row=7, column=4)

    confirm.mainloop()

# 1-4-2. 결제화면 구성
def pay_window():
    # 이벤트 처리함수
    def calc_pay():
        global total_money
        global change
        pay_money = int(pay_entry.get()) # 지역변수
        change = pay_money - total_money

        if change > 0:
            pay.destroy() # 창 파괴
            coin_window() # 거스름돈 창으로 이동
        elif change < 0:
            total_money = -(change) # 지불한 금액 처리 -> 나머지 결제금액 저장
            pay.destroy() # 현재 창 파괴
            error_window() # 에러 창 띄우기
        else: # 결제완료
            pay.destroy() # 창 파괴
            end_window() # 결제완료 창

    pay = Tk()
    pay.title("결제화면")

```

```

pay.geometry("700x700") # 메인 화면 창 크기 조절하기
pay.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

total_label = Label(pay, text="결제예정금액: " + str(total_money) + " 원",
font=("Consolas", 20))
total_label.grid(row=0, column=0, columnspan=4)

pay_label = Label(pay, text="금액을 투입해 주세요💰💰💰", font=("Consolas", 20))
pay_label.grid(row=1, column=0, columnspan=3)
pay_entry = Entry(pay)
pay_entry.grid(row=1, column=3)

pay_button = Button(pay, text="결제", command=calc_pay, font=("Consolas", 15))
pay_button.grid(row=2, column=3)

pay.mainloop()

# 1-4-3. 후처리
# i) 거스름돈 지불 화면
def coin_window():
    # 거스름돈 계산하기
    global change
    coin_500s = change // 500
    new_change = change % 500
    coin_100s = new_change // 100

    # 이벤트 처리 함수
    def reset1():
        coin.destroy() # 현재 창 파괴

    coin = Tk()
    coin.title("거스름돈 화면")
    coin.geometry("500x500") # 메인 화면 창 크기 조절하기
    coin.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

    coin_label = Label(coin, text="거스름돈: " + str(change) + " 원", font=("Consolas",
20))
    coin_label.grid(row=0, column=0, columnspan=4)

    c500_label = Label(coin, text="500원: " + str(coin_500s) + " 개", font=("Consolas",
20))

```

```

c500_label.grid(row=1, column=0, columnspan=4)

c100_label = Label(coin, text="100원: " + str(coin_100s) + " 개", font=("Consolas",
20))
c100_label.grid(row=2, column=0, columnspan=4)

thank_label = Label(coin, text="이용해 주셔서 감사합니다~♥♥♥♥", bg="light
salmon", font=("Consolas", 20))
thank_label.grid(row=3, column=0, columnspan=4)

complete_button = Button(coin, text="종료", command=reset1, font=("Consolas",
20))
complete_button.grid(row=4, column=3)

coin.mainloop()

# ii) 잔액부족 페이지
def error_window():
    # 이벤트 함수
    def go_back():
        error.destroy()
        pay_window() # 결제 화면으로 되돌아가기

    error = Tk()
    error.title("잔액부족 화면")
    error.geometry("500x500") # 메인 화면 창 크기 조절하기
    error.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

    lack_label = Label(error, text="잔액 부족!! " + str(total_money) + " 원이 부족합니
다.",
                        font=("Consolas", 20), bg="OrangeRed")
    lack_label.grid(row=0, column=0, columnspan=4)

    repay_button = Button(error, text="추가 결제하기", command=go_back,
font=("Consolas", 15))
    repay_button.grid(row=1, column=3)

    error.mainloop()

# iii) 결제완료(거스름돈x)
def end_window():

```

```
# 이벤트 처리 함수
def reset2():
    end.destroy() # 창 파괴하기

end = Tk()
end.title("결제완료화면")
end.geometry("500x500") # 메인 화면 창 크기 조절하기
end.resizable(width=1, height=1) # 가로,세로 둘 다 크기 조정

thank_label = Label(end, text="이용해 주셔서 감사합니다~❤❤❤❤", bg="light salmon",
font=("Consolas", 20))
thank_label.grid(row=0, column=0, columnspan=4)

complete_button = Button(end, text="종료", command=reset2, font=("Consolas",
20))
complete_button.grid(row=1, column=3)

end.mainloop()

main_window() # 제일 처음에 메인 화면 표시해주기
```