

Table of Contents 목차

- 001
 DB
 및 화면
 설계
- 002 **GUI 화면**
- 003 내부 코드 리뷰





Database

DDL & DML

```
DROP TABLE od_list:
DROP TABLE product;
DROP TABLE t_od_list;
DROP SEQUENCE seq_oid;
-- 제품목록 테이블
CREATE TABLE product(
   pid NUMBER, -- idx 제품번호
   name VARCHAR2(60), -- 제품명
   price NUMBER -- 가격
):
-- 주문내역 테이블
CREATE TABLE t_od_list(
   oid NUMBER, -- 주문번호
   t_price NUMBER. -- 합계금액
   order_date date -- 주문일자
):
CREATE SEQUENCE seq_oid;
INSERT INTO product VALUES(1, '인터로킹 G 슬라이더 샌들 black', 490000);
INSERT INTO product VALUES(2, '인터로킹 G 슬라이더 샌들 white', 490000);
INSERT INTO product VALUES(3, '웹 패널 슬라이더 샌들', 440000);
INSERT INTO product VALUES(4, 'GG 수프림 프린트 슬라이더 샌들', 600000);
INSERT INTO product VALUES(5, '월드와이드 슬리퍼', 450000);
INSERT INTO product VALUES(6, '로고 가죽 슬리퍼', 790000);
commit;
```

상품은 6가지 상품으로 구성되어 있으며, 각각 제품번호, 제품명, 가격 데이터를 갖고 있습니다.

- 제품목록 테이블
 - 제품번호
 - 제품명
 - 가격
- 주문내역 테이블
 - 주문번호
 - 합계금액
 - 주문일자

GUI 설계

화면 설계



Frame 설계 코드

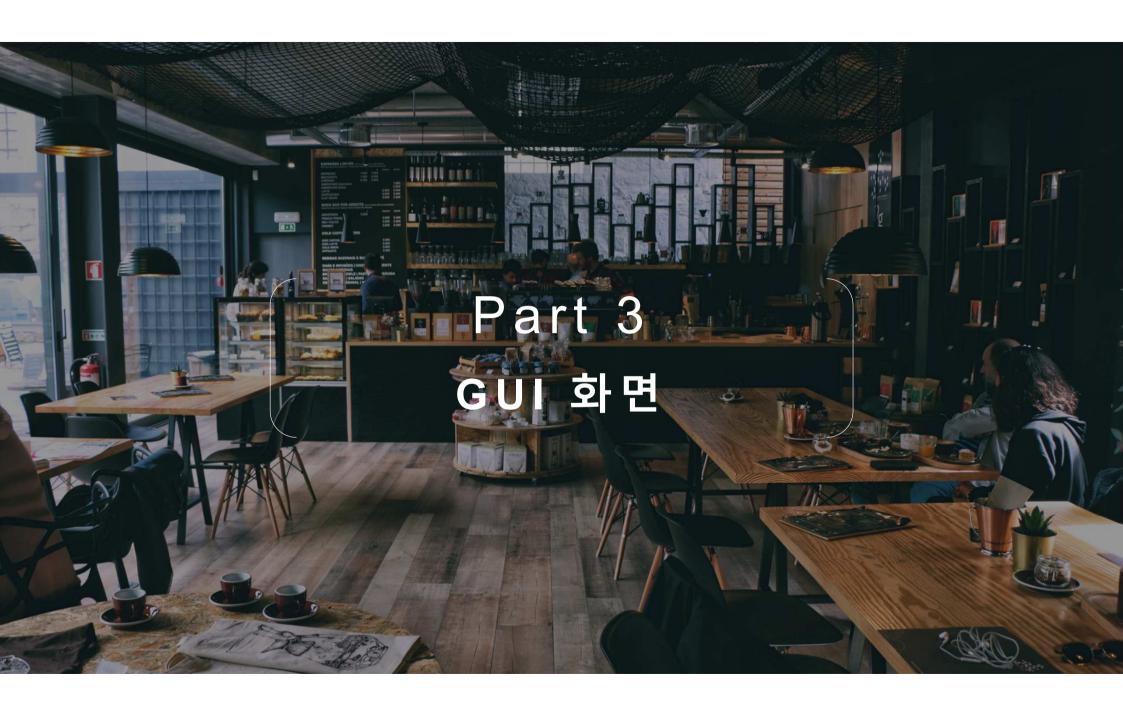
```
frame = tk.Frame(root, bd=5, bg='light blue', relief='groove') # 메인프레임
frame.pack()

L_frame = tk.Frame(root, bg='red') # 좌축 프레임
L_frame.pack(side=tk.LEFT, fill="y")

r_frame = tk.Frame(root, bg='blue') # 우축 프레임
r_frame.pack(side=tk.RIGHT, fill="both", expand=True)

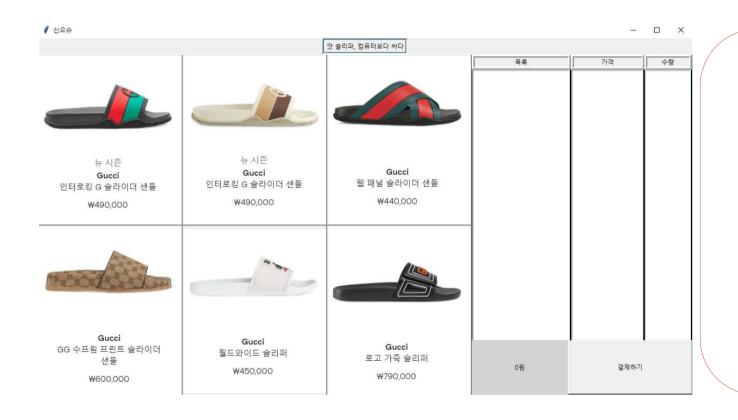
rt_frame = tk.Frame(r_frame, bg='black') # 우축상단 프레임
rt_frame.pack(side=tk.TOP, fill="both", expand=True)

rb_frame = tk.Frame(r_frame, bg='yellow') # 우축하단 프레임
rb_frame.pack(side=tk.BOTTOM)
```



화 면

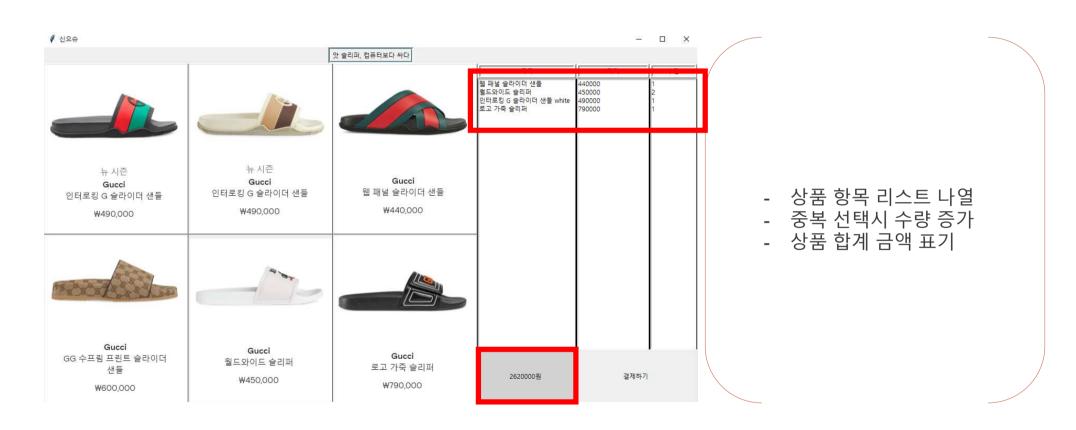
초기 화면



- 우측 프레임 세분화합계 금액 추가

화 면

상품 선택



화 면

결제 화면





좌측 프레임 구성

```
label = tk.Label(frame, text='앗 슬리퍼, 컴퓨터보다 싸다') # top 폭에 올라감
label pack()
img1 = ImageTk.PhotoImage(Image.open("image/s1.png"))
button1 = tk.Button(l_frame, text="메뉴1", width='270', height='320', image=img1, command=lambda: order_list(1))
button1.grid(row=0, column=0)
img2 = ImageTk.PhotoImage(Image.open("image/s2.png"))
button2 = tk.Button(l_frame, text="叫뉴2", width='270', height='320', image=imq2, command=lambda: order_list(2))
button2.grid(row=0, column=1)
img3 = ImageTk.PhotoImage(Image.open("image/s3.png"))
button3 = tk.Button(l_frame, text="메뉴3", width='270', height='320', image=img3, command=lambda: order_list(3))
button3.grid(row=0, column=2)
img4 = ImageTk.PhotoImage(Image.open("image/s4.png"))
button1 = tk.Button(l_frame, text="메뉴4", width='270', height='320', image=img4, command=lambda: order_list(4))
button1.grid(row=1, column=0)
imq5 = ImageTk.PhotoImage(Image.open("image/s5.png"))
button5 = tk.Button(l_frame, text="메뉴5", width='270', height='320', image=img5, command=lambda: order_list(5))
button5.grid(row=1, column=1)
img6 = ImageTk.PhotoImage(Image.open("image/s6.png"))
button6 = tk.Button(l_frame, text="메뉴6", width='270', height='320', image=img6, command=lambda: order_list(6))
button6.grid(row=1, column=2)
```

우측 프레임 구성

```
od_list = tk.Label(rt_frame, text='목록', width='25', borderwidth=6, relief="ridge")
od_list.grid(row=0, column=0)
odl_list = tk.Listbox(rt_frame, width='26', height=32)
odl_list.grid(row=1, column=0)
od_price = tk.Label(rt_frame, text='가격', width='18', borderwidth=6, relief="ridge")
od_price.grid(row=0, column=1)
odl_price = tk.Listbox(rt_frame, width='19', height=32)
odl_price.grid(row=1, column=1)
od_cnt = tk.Label(rt_frame, text='수량', width='11', borderwidth=6, relief="ridge")
od_cnt.grid(row=0, column=2)
odl_cnt = tk.Listbox(rt_frame, width='12', height=32)
odl_cnt.grid(row=1, column=2)
total_price = tk.Label(rb_frame, text="0원", width='25', height='7', borderwidth=3, background='lightgray')
total_price.pack(side=tk.LEFT)
pay_button = tk.Button(rb_frame, text="결제하기", width='39', height='7', borderwidth=3, command=lambda: pay())
pay_button.pack(side=tk.RIGHT, fill="both")
```

상품 클릭 함수

```
# 상품 클릭시

def order_list(cid):
    print(cid)

# 1. 연결 일어오기
    conn = oci.connect('ksk/ksk@127.0.0.1:1521/xe')
    print('연결 성공', conn.version)
# 2) 커서(Cursor) 일어오기
    cursor = conn.cursor()
# 3) SQL 문장

sql = 'SELECT * FROM product WHERE pid={}'.format(cid)
    print(sql)
# 4) SQL 실행
    cursor.execute(sql)
    datas = cursor.fetchall()
    data = datas[0]
    print(data)
```

```
state = True
for i in range(6):
 if data[1] == odl_list.get(i):
   cnt = odl_cnt.get(i)
   odl_cnt.delete(i)
   odl_cnt.insert(i, cnt + 1)
   state = False
if state:
 size = odl_list.size()
  odl_list.insert(size, data[1])
 odl_price.insert(size, data[2])
 odl_cnt.insert(size, 1)
total = total_price['text']
t_price = int(total[:len(total) - 1])
total_price['text'] = str(t_price + (int(data[2]))) + "원"
cursor.close()
conn.close()
```

결제 클릭 함수

```
def pay():
 conn = oci.connect('ksk/ksk@127.0.0.1:1521/xe')
 print('연결 성공', conn.version)
 cursor = conn.cursor()
 price = total_price['text'][:len(total_price['text']) - 1]
 sql = 'INSERT INTO t_od_list VALUES(seq_oid.nextval, {}, SYSDATE)'.format(
  price)
 print(sql)
 cursor.execute(sql)
 conn.commit()
 cursor close()
 conn.close()
 odl_list.delete(0, odl_list.size() - 1)
 odl_price.delete(0, odl_price.size() - 1)
 odl_cnt.delete(0, odl_cnt.size() - 1)
 total_price['text'] = '0원'
```

