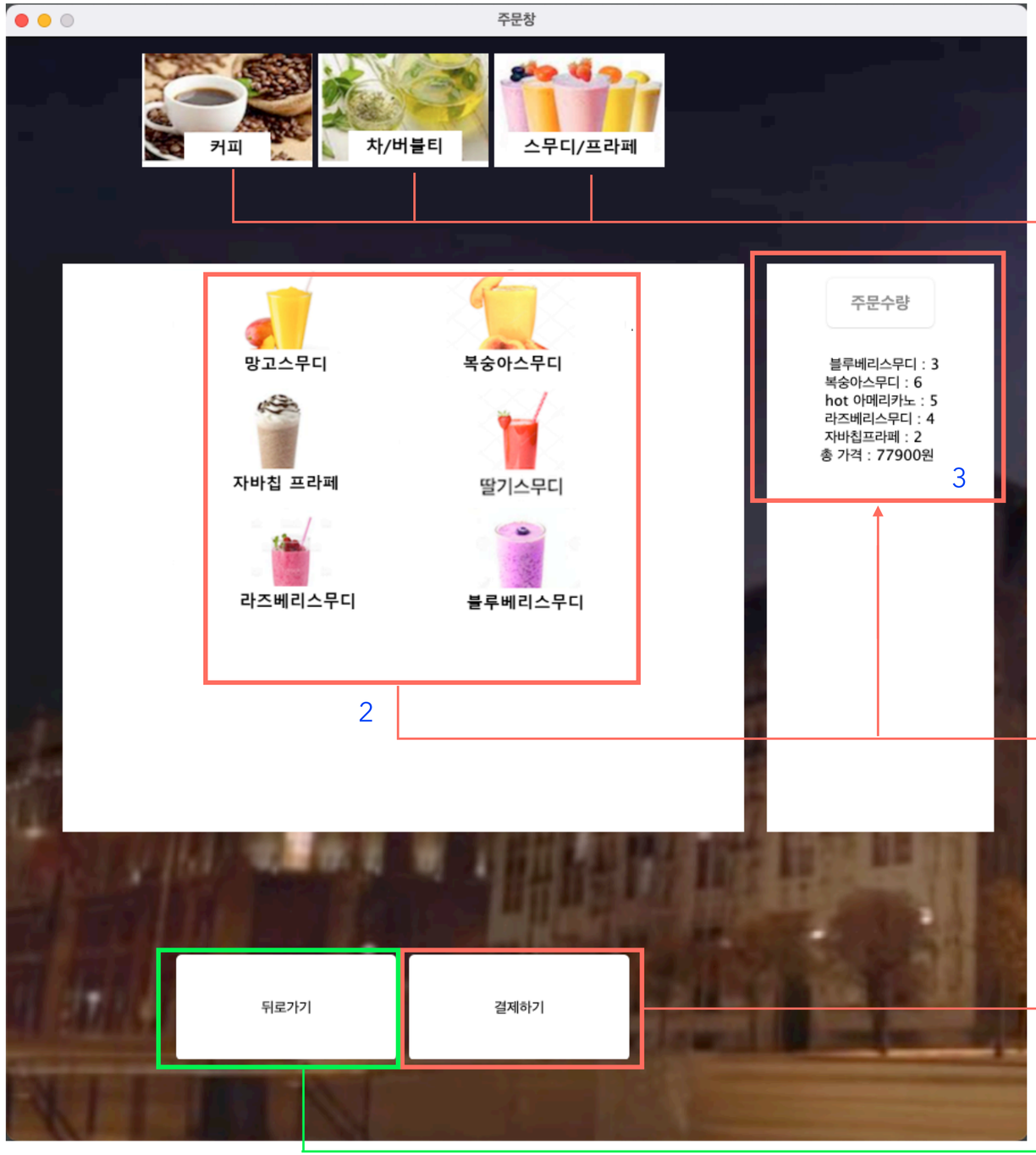
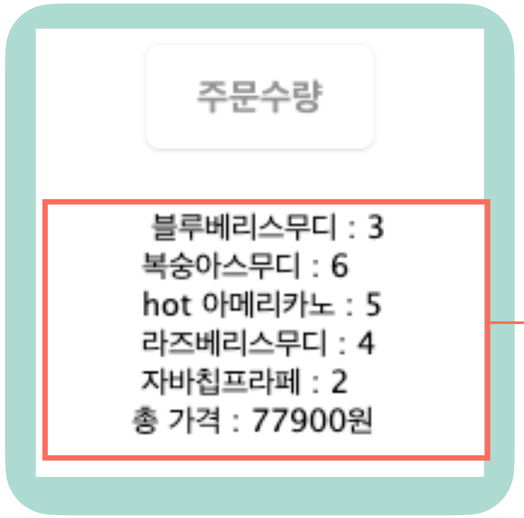


1. 카테고리 전환 버튼.
커피를 누르면 커피메뉴가, 차를 누르면 차 메뉴가,
스무디를 누르면 스무디 메뉴로 메뉴들이 전환됨.

2번 메뉴를 선택을 하면 3번 주문 수량 창에서
메뉴별 각 수량과 총가격을 확인이 가능하다.

결제버튼을 누르면 3번 주문 수량창에 입력된 정보가
DB KGOrder테이블에 저장이 되고 다음 페이지인
휴대전화번호 입력 페이지로 넘어간다.

뒤로가기 버튼을 누르면 이전 페이지인 메인화면으로 돌아간다.



Oracle DB의 kgorder table

	O_NUMBER	STRINGORDER
1	<null>	블루베리스무디:3개 복숭아스무디:6개 hot 아메리카노:5개 라즈베리스무디:4개 자바칩프라페:2개

주문수량은 주문을 받을때는 HashMap을 이용하여 메뉴별 수량과 총 가격을 저장을 하고 결제버튼을 누른 후 각 행의 key값과 value값을 한줄의 String값에다 저장하여 한줄로 만든 후 varchar2로 DB에 저장이 된다.

```java
public interface OrderREPO {
    //HashMap<String, Integer> orderInfo = null;
    void save(String menu, Integer count);
    Integer getBevCount(String name);
}
```

먼저 OrderREPO라는 인터페이스를 만들어서 메소드를 정의함.
DB에 주문내역을 올리기 전, 임시로 주문 내역을 담을
HashMap에 저장시키는 save()와
음료 수량을 나타내는 getBevCount() 메소드를 우선 만들어 준다

```java
class OrderRepository implements OrderREPO{
    HashMap<String, Integer> MemoryOrderRepository = new HashMap<>();

    @Override
    public void save(String menu, Integer count) {
        MemoryOrderRepository.put(menu, count);

    }

    @Override
    public Integer getBevCount(String name) {
        return MemoryOrderRepository.get(name);
    }



    public String passTotalString(){
        String totalOrder = "";
        for(String menu : MemoryOrderRepository.keySet()){
            totalOrder += menu+":"+MemoryOrderRepository.get(menu)+"  ";
        }
        System.out.println(totalOrder+"진짜 객체 전달 됨?");
        return  totalOrder;
    }
```

다음으로 OrderREPO를 구체화 하는 클래스 OrderRepository 를 만들고
save(menu, count)를 받으면 HashMap에 이름과 갯수가 저장되도록
설정하고, getBevCount(name)을 받으면 특정 매뉴의 수량을 리턴하도록 구체화 시킨다.

```java
public void addTotal(String bevName, Integer number, Integer price){

    if(orderRepository.getBevCount(bevName) == null){
        orderRepository.save(bevName, count: 1);
    } else{
        int k = orderRepository.getBevCount(bevName);
        orderRepository.save(bevName, count: k+1);
    }
    totalPrice += price;

}
```

OrderRepository에서 만든 save()와 getBevCount()를 이용해 메뉴를 클릭 했을때
이전 슬라이드에서 만들어둔 HashMap에 메뉴가 없으면
메뉴가 추가되도록 하고, 이미 있다면 수량에 1을 더해주도록 addTotal을 만들어 준다.

```java
public void showTotalBeverage(){
    totalOrder.setText(" ");

    for(String menu: orderRepository.MemoryOrderRepository.keySet()){
        String str1 = " "+ menu+" : "+orderRepository.getBevCount(menu);

        totalOrder.append(str1+"\n");
        totalPanel.add(totalOrder);

    }
    totalOrder.append("총 가격 : "+totalPrice+"원");
    System.out.println("총 가격 : "+totalPrice);

}
```

현재까지 추가한 메뉴목록과 개수, 총 가격을 알려주는 showTotalBeverage() 메소드로
향상된 for문을 이용하여 HashMap내의 key값과 value값을 한줄에 저장을하여 한줄씩
총 주문내역이 올라가는 패널인 totalPanel에 추가해준다.

```
if(e.getSource() == chargeButton){
    System.out.println("Dddd");System.out.println("결제시작");

    for(String menu: orderRepository.MemoryOrderRepository.keySet()){
        String str1 = "    "+ menu+":"+orderRepository.getBevCount(menu)+"개";
        stringOrder += str1;
    }
    System.out.println(stringOrder);

    orderDTO.setStringOrder(stringOrder);
    int re = orderDAO.insertOrder(orderDTO);
    if(re==1) System.out.println("방명록 저장에 성공");
    this.dispose();
    setVisible(false);
    Phone_Acc phone_acc = new Phone_Acc();
```

최종적으로 결제버튼을 누르게 되면 HashMap내에 저장된 key값과 value들 전체를
향상된 for문을 이용하여 한줄에다 더해준다
(관리자가 주문 내역을 한번에 볼 수 있게 하기 위해 한줄에 담아 DB로 보내기 위해)

```
con = DriverManager.getConnection(url, user, pwd);
sql = "insert into kgorder (StringOrder) values(?)";
pstmt = con.prepareStatement(sql);
```

orderDTO에서 정의해준 SQL문에 의해 최종적으로 DB에
한줄짜리 전체 주문내역이 들어가게 된다.