



JAVA PROJECT - CAFE KIOSK

# 자바 프로젝트 카페 키오스크 제작

팀명 KIMCHEESE





CHAPTER 1



# KIOSK 구조 계획



POINT. 01

## MAIN

- ▶ 메인 실행 프로그램
- ▶ 프로그램 흐름 제어
- ▶ 주문 반복

POINT. 02

## MENU

- ▶ 메뉴 목록 관리
- ▶ 메뉴 정보 제공
- ▶ 추천 메뉴 표시

POINT. 03

## ORDER

- ▶ 주문 처리
- ▶ 추가 주문 여부
- ▶ 수량/가격 계산
- ▶ 주문 상세 정보

POINT. 04

## PAYMENT

- ▶ 결제 방식 및 포인트 적립 여부
- ▶ 결제 완료 표시
- ▶ 영수증 출력 여부
- ▶ 주문번호 표시

ORDER SYSTEM

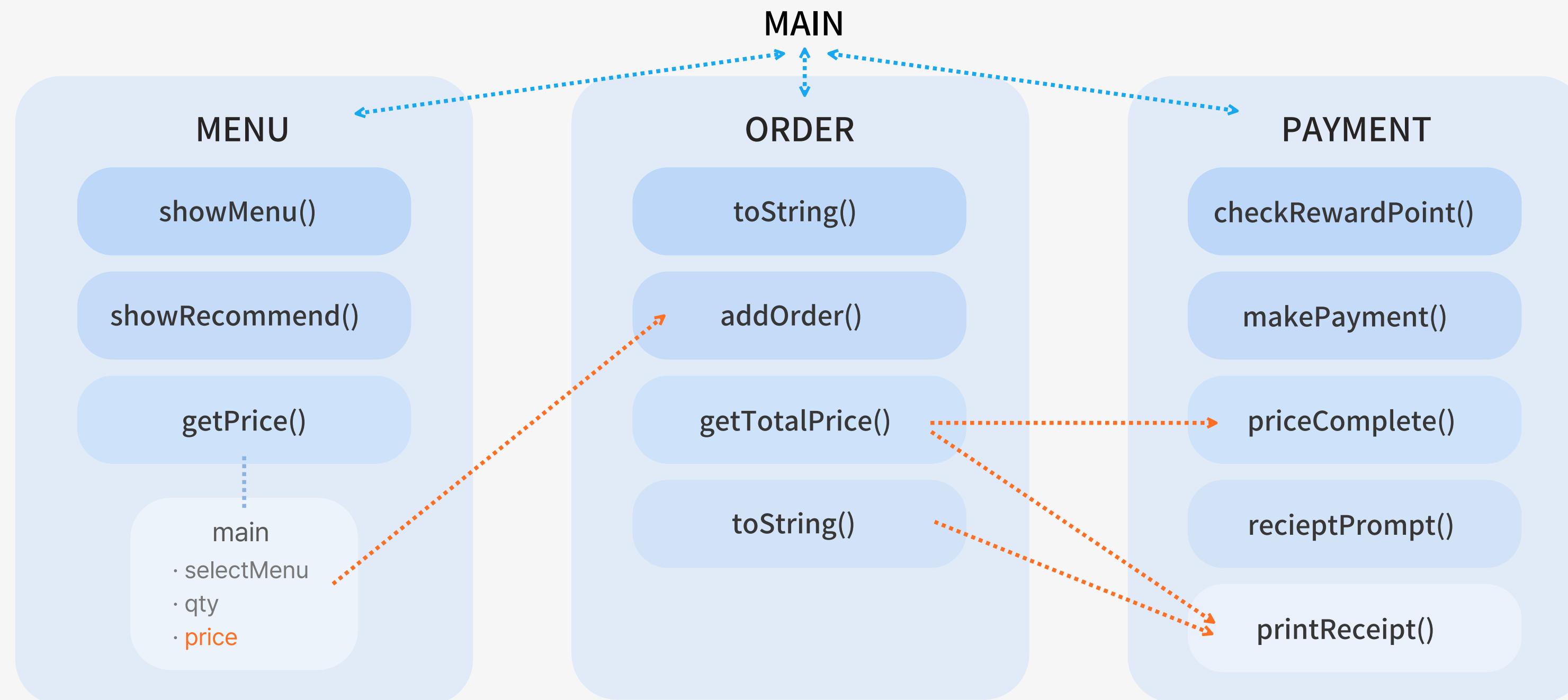




CHAPTER 02



# 키오스크 진행순서



※ MAIN과 PAYMENT는 참조 변수(REFERENCE VARIABLE)를 통해 같은 SCANNER 객체를 공유



CHAPTER 03



# 문제점과 해결 방안

#P1

SCANNER 중복 오류

```
package com.cafe.kiosk;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner user = new Scanner(System.in);
        Menu menu = new Menu();
        while (true) {
            Order order = new Order();
            Payment payment = new Payment(order, user);
```



#S1

PAYMENT SCANNER 대신에 MAIN SCANNER 공유

```
private Scanner sc; //선언만 함
public Payment(Order order, Scanner user) {
    this.order = order;
    this.sc = user; //하나만 사용
}
```

- MAIN에서 만든 SCANNER를 받아 입력 충돌 방지 / 다른 클래스엔 SCANNER공유를 위해 선언 필수



CHAPTER 03



# 문제점과 해결 방안

#P2

입력 버퍼 문제(INPUT BUFFER ISSUE)

```
//영수증 출력여부
public void receiptPrompt(){
    while(true) {
        System.out.println("영수증을 출력하시겠습니까?(Y/N)\n:");
        String input = sc.next(); //sc.next();
```

```
// 2. 수량 입력
System.out.println("수량을 입력해주세요:");
int qty = Integer.parseInt(user.nextLine().trim());
order.addOrder(selectMenu, qty, price);

System.out.println("추가 주문하시겠습니까? (y/n)");
String more = user.nextLine().trim();
```



#S2

NEXT() 대신 NEXTLINE()으로 처리

```
// 영수증 출력 여부
public void receiptPrompt() {
    while (true) {
        System.out.println("영수증을 출력하시겠습니까?(Y/N)\n:");
        String input = sc.nextLine().trim(); //바꿈
```

```
// 수량 입력
System.out.println("수량을 입력해주세요:");
int qty;
while (true) {
    try {
        qty = Integer.parseInt(user.nextLine().trim());
        break;
    } catch (Exception e) {
        System.out.println("숫자로 입력해주세요:");
    }
}
int price = menu.getPrice(selectMenu);
order.addOrder(selectMenu, qty, price);
```

```
// 추가 주문
while (true) {
    System.out.println("추가 주문하시겠습니까? (Y/N)");
    String more = user.nextLine().trim();
```

MAIN과 PAYMENT의 입력 방식 /  
NEXT()와 NEXTLINE()의 차이점



CHAPTER 03



# 문제점과 해결 방안

#P3

주문 번호 (값 누적 실패 / 증가 안됨)

```
package com.cafe.kiosk;

import java.util.Scanner;

public class Payment {

    private boolean reward; //포인트 적립여부
    private String pay; //결제 방식
    private Order order; //주문
    private int orderNumber = 1; //주문번호
    private int rewardPoint = 0; //적립금
```



#S3

주문번호를 STATIC 변수로 선언

```
package com.cafe.kiosk;

import java.util.Scanner;

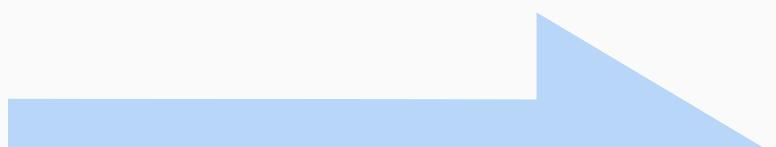
public class Payment {

    private boolean reward;
    private String pay;
    private Order order;
    private static int orderNumber = 1;
    private int rewardPoint = 0;
```

--- • STATIC 사용으로 주문번호 누적

JAVA PROJECT (KIOSK)

# 실제 구현 바탕 코딩 시연



※ 해당 페이지 이후는 ECLIPSE 시연으로 진행됩니다.