

SH Cafe



신선호

Table of Contents


목 차

001 프로젝트 개요

002 알고리즘

003 프로젝트 코드 및 결과



The background image shows a warm, rustic interior. A dark wooden table is surrounded by yellow chairs. The wall is made of exposed brick. On the left, there's a wooden shelf with a coffee machine and some boxes. On the right, two wooden skis are leaning against the wall. The lighting is warm and ambient.

Part 1

프로젝트 개요

1. 프로젝트 개요

- Java언어의 부족한 부분을 메우기 위한 개인 프로젝트입니다.
- 항목별로 메뉴를 고른 뒤, 결제를 누르면 현재까지 선택한 메뉴들을 결제할 수 있습니다.
- 매출통계를 통해서 현재까지의 매출합계와 카페에서 가장 많이 팔린 메뉴를 알 수 있습니다.
- 활용기술 및 툴

Java

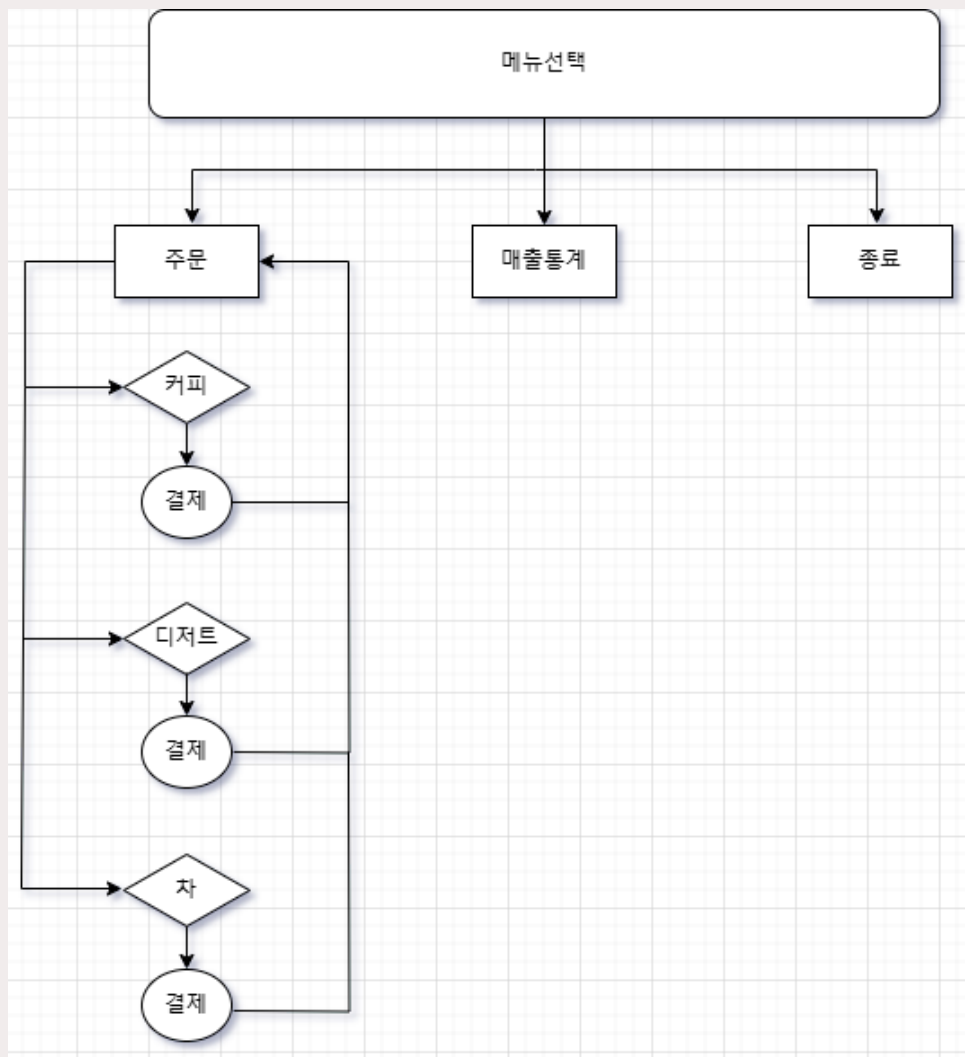
Eclipse

A close-up photograph of three hands holding coffee cups in a toast. The top hand holds a white cup with latte art. The bottom-left hand holds a dark cup with ice. The bottom-right hand holds a white cup with latte art. The background is a blurred wooden table with coffee-making equipment.

Part 2

알고리즘

2. 알고리즘



Part 3

프로젝트 코드 및 결과

3. 프로젝트 코드

<Main>

```
public class Main {
    private static Scanner scanner = new Scanner(System.in);
    private static Menu menu = new Menu();
    private static Order order = new Order();
    private static Payment payment = new Payment();

    public static void main(String[] args) {
        while (true) {
            menu.showMainMenu();
            int mainChoice = scanner.nextInt();
            switch (mainChoice) {
                case 1:
                    MainMenu();
                    break;
                case 2:
                    TotalStatistics();
                    break;
                case 3:
                    System.out.println("시스템 종료");
                    return;
                default:
                    System.out.println("잘못된 선택입니다."); } } }

    private static void MainMenu() {
        menu.showCategoryMenu();
        int categoryChoice = scanner.nextInt();
        switch (categoryChoice) {
            case 1:
                menu.showCoffeeMenu();
                ItemCount(menu.coffeeMenu, menu.coffeePrices);
                break;
            case 2:
                menu.showDessertMenu();
                ItemCount(menu.dessertMenu, menu.dessertPrices);
                break;
            case 3:
                menu.showTeaMenu();
                ItemCount(menu.teaMenu, menu.teaPrices);
                break;
            default:
                System.out.println("잘못된 선택입니다."); } } }
```

```
private static void ItemCount(String[] items, int[] prices) {
    int itemChoice = scanner.nextInt();
    if (itemChoice >= 1 && itemChoice <= items.length) {
        String item = items[itemChoice - 1];
        int price = prices[itemChoice - 1];
        System.out.println(item + "를 선택하셨습니다. 수량을 입력해주세요");
        int count = scanner.nextInt();
        order.addOrder(item, price, count);
        System.out.println("수량: " + count + "개");
        System.out.println("1.결제 2.주문목록");
        int choice = scanner.nextInt();
        if (choice == 1) {
            int totalOrderAmount = order.getTotalOrderAmount();
            boolean paymentSuccessful = payment.processPayment(totalOrderAmount);
            if (paymentSuccessful) {
                order.processPayment(totalOrderAmount);
                order.clearOrder();
            }
        } else if (choice == 2) {
            order.showOrderList();
        } else {
            System.out.println("잘못된 선택입니다.");
        }
    } else if (itemChoice == items.length + 1) {
        order.showOrderList();
    } else {
        System.out.println("잘못된 선택입니다.");
    }
}

private static void TotalStatistics() {
    order.showSalesStatistics();
}
```


3. 프로젝트 코드

<Menu>

```
public class Menu {  
    public String[] coffeeMenu = {"아메리카노", "카페라테", "카푸치노", "카페모카", "아포카토"};  
    public int[] coffeePrices = {4000, 4500, 4500, 5000, 5500};  
  
    public String[] dessertMenu = {"샌드위치", "에그타르트", "딸기 케이크", "치즈 케이크"};  
    public int[] dessertPrices = {6000, 3000, 7000, 7000};  
  
    public String[] teaMenu = {"녹차", "레몬차", "유자차"};  
    public int[] teaPrices = {3500, 3500, 3500};  
  
    public void showMainMenu() {  
        System.out.println(" ");  
        System.out.println("    메뉴선택    ");  
        System.out.println(" ");  
        System.out.println("=====");  
        System.out.println("▶1.주문 2.매출 통계 3.종료◀");  
        System.out.println("=====");  
    }  
  
    public void showCategoryMenu() {  
        System.out.println("1.커피 2.디저트 3.차");  
    }  
  
    public void showCoffeeMenu() {  
        System.out.println("1.아메리카노(4000원) 2.카페라테(4500원) 3.카푸치노(4500원) 4.카페모카(5000원) 5.아포카토(5500원) 6.주문목록");  
    }  
  
    public void showDessertMenu() {  
        System.out.println("1.샌드위치(6000원) 2.에그타르트(3000원) 3.딸기 케이크(7000원) 4.치즈 케이크(7000원) 5.주문목록");  
    }  
  
    public void showTeaMenu() {  
        System.out.println("1.녹차(3500원) 2.레몬차(3500원) 3.유자차(3500원)");  
    }  
}
```

3. 프로젝트 코드

<Order>

```
public class Order {
    private List<String> orderList = new ArrayList<>();
    private List<Integer> orderPrices = new ArrayList<>();
    private int totalSales = 0;
    private List<String> salesItems = new ArrayList<>();
    private List<Integer> salesCounts = new ArrayList<>();

    public void addOrder(String item, int price, int count) {
        int totalPrice = price * count;
        orderList.add(item + " x " + count);
        orderPrices.add(totalPrice);

        int index = salesItems.indexOf(item);
        if (index == -1) {
            salesItems.add(item);
            salesCounts.add(count);
        } else {
            salesCounts.set(index, salesCounts.get(index) + count);
        }
    }

    public int getTotalOrderAmount() {
        int total = 0;
        for (int price : orderPrices) {
            total += price;
        }
        return total;
    }
}
```

```
    public void clearOrder() {
        orderList.clear();
        orderPrices.clear();
    }

    public void processPayment(int amount) {
        totalSales += amount;
    }

    public int getTotalSales() {
        return totalSales;
    }

    public void showOrderList() {
        System.out.println("주문목록:");
        for (int i = 0; i < orderList.size(); i++) {
            System.out.println(orderList.get(i) + ": " + orderPrices.get(i) + "원");
        }
    }

    public void showSalesStatistics() {
        System.out.println("총 매출 통계: " + totalSales + "원");
        System.out.println("판매된 메뉴 목록:");
        for (int i = 0; i < salesItems.size(); i++) {
            System.out.println(salesItems.get(i) + " x " + salesCounts.get(i));
        }
    }
}
```

3. 프로젝트 코드

<Payment>

```
public class Payment {  
    private Scanner scanner = new Scanner(System.in);  
  
    public boolean processPayment(int amount) {  
        int payment;  
        while (true) {  
            System.out.println(amount + "원을 계산해주세요");  
            payment = scanner.nextInt();  
            if (payment >= amount) {  
                break;  
            } else {  
                System.out.println("결제금액이 부족합니다. 다시 결제해주세요.");  
            }  
        }  
  
        int change = payment - amount;  
        System.out.println("결제완료. 잔돈은 " + change + "원 입니다.");  
        return true;  
    }  
}
```


3. 프로젝트 결과

<주문>

메뉴선택

=====

▶1. 주문 2. 매출 통계 3. 종료◀

=====

1

1. 커피 2. 디저트 3. 차

1

1. 아메리카노(4000원) 2. 카페라테(4500원) 3. 카푸치노(4500원) 4. 카페모카(5000원) 5. 아포카토(5500원) 6. 주문목록

1

아메리카노를 선택하셨습니다. 수량을 입력해주세요

2

수량: 2개

1. 결제 2. 주문목록

2

주문목록:

아메리카노 x 2: 8000원

1. 커피 2. 디저트 3. 차

2

1. 샌드위치(6000원) 2. 에그타르트(3000원) 3. 딸기 케이크(7000원) 4. 치즈 케이크(7000원) 5. 주문목록

4

치즈 케이크를 선택하셨습니다. 수량을 입력해주세요

1

수량: 1개

1. 결제 2. 주문목록

2

주문목록:

아메리카노 x 2: 8000원

치즈 케이크 x 1: 7000원

1. 커피 2. 디저트 3. 차

3

1. 녹차(3500원) 2. 레몬차(3500원) 3. 유자차(3500원)

3

유자차를 선택하셨습니다. 수량을 입력해주세요

1

수량: 1개

1. 결제 2. 주문목록

1

18500원을 계산해주세요

15000

결제금액이 부족합니다. 다시 결제해주세요.

18500원을 계산해주세요

20000

결제완료. 잔돈은 1500원 입니다.

3. 프로젝트 결과

<매출 통계>

메뉴선택

=====

▶1.주문 2.매출 통계 3.종료◀

=====

2


총 매출 통계: 18500원

판매된 메뉴 목록:

아메리카노 x 2

치즈 케이크 x 1

유자차 x 1

A top-down view of four white ceramic coffee cups arranged in a square on a dark, textured wooden tray. Each cup contains a latte with intricate white foam art. The cups have dark blue handles. A large, semi-transparent light orange circle is centered over the cups, containing the text 'Thank You' and '감사합니다' in a dark, sans-serif font.

Thank You
감사합니다