

Snack Manager Program

Together Inc 

기업 간식관리 프로그램

서해원

채여운

양재영



목 차

01 개요

팀원소개
-
프로그램
소개/ 목적



02 프로젝트상세

주요 사용구문
-
프로그램 시연
-
제품 설명서



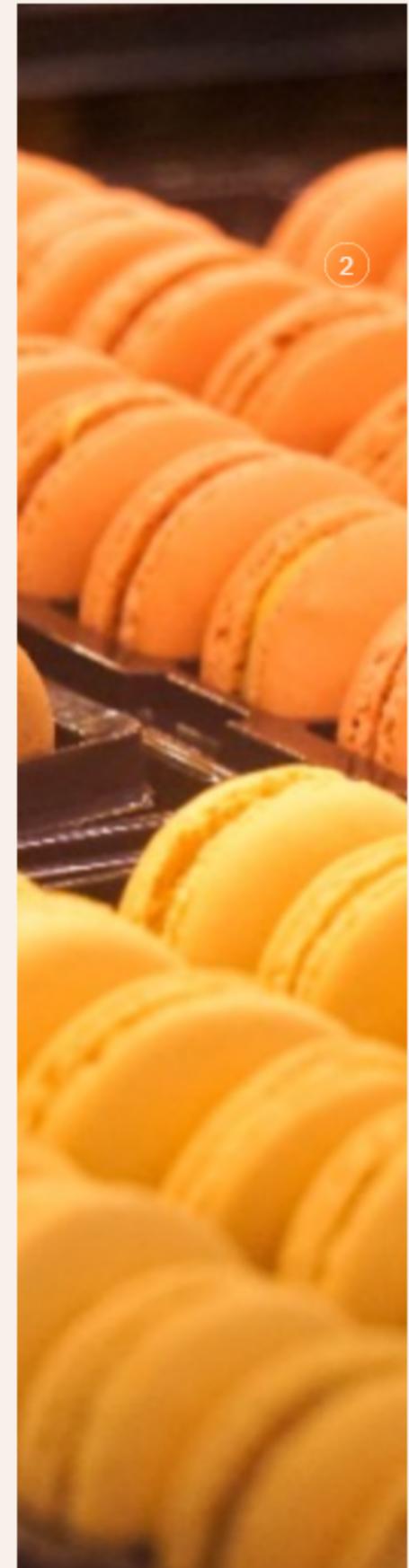
04 마치며

그룹원 느낀점



03 앞으로의 방향성

향후 계획



01 개요

팀원 소개



팀원 소개

서해원



- 로그인 시스템

- 사원 시스템

- 오류 검증 및 수정

채여운



- 아이디어 제시

- 간식관리자 시스템

- 오류 검증 및 수정

양재영



- PPT제작/발표

- 사원관리 시스템

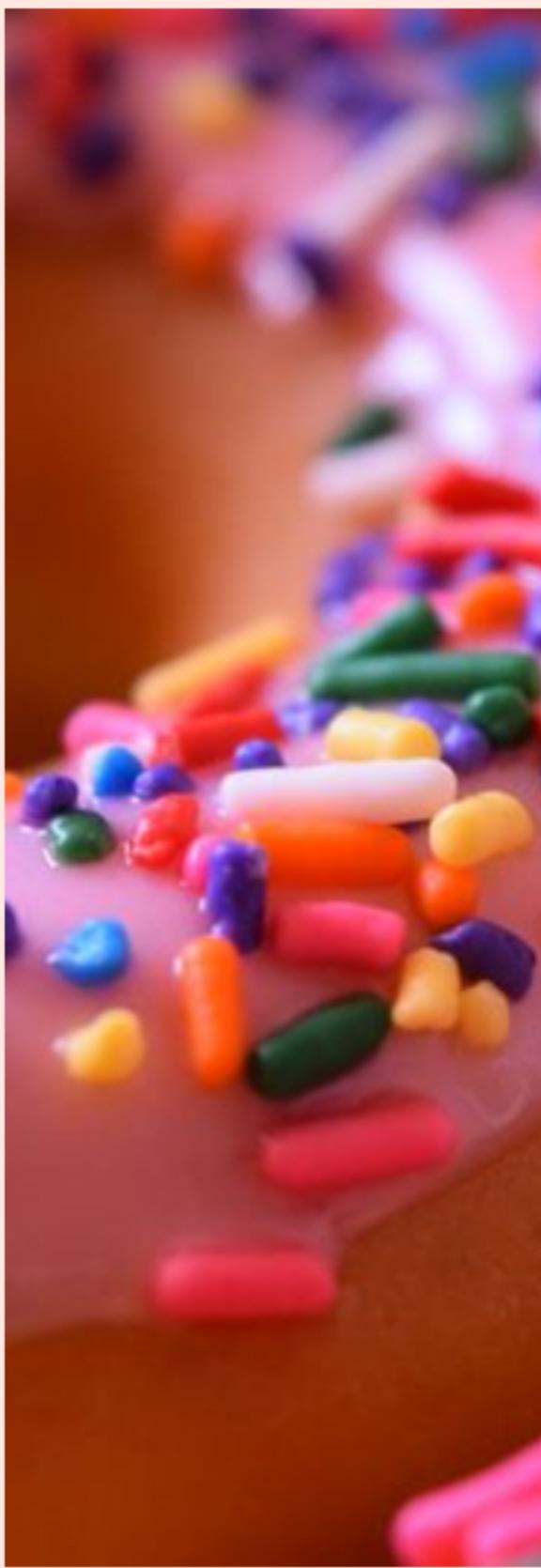
- 오류 검증 및 수정

01 개요

프로그램 소개



프로그램 소개



① 개발의도

회사가 직원에게 간식을 지급.
직원들의 사기와 생산성을 높일 수 있도록 하는
체계적인 간식 관리 프로그램을 제작 하였습니다.

간식 복지, 사무실의 소소하지만 확실한 행복입니다.
본 Together Inc의 간식 관리 프로그램은 체계적으로
간식을 관리해 직원들의 불만 없는 탕비실 간식 창고
이용을 돋습니다.

02 프로젝트상세

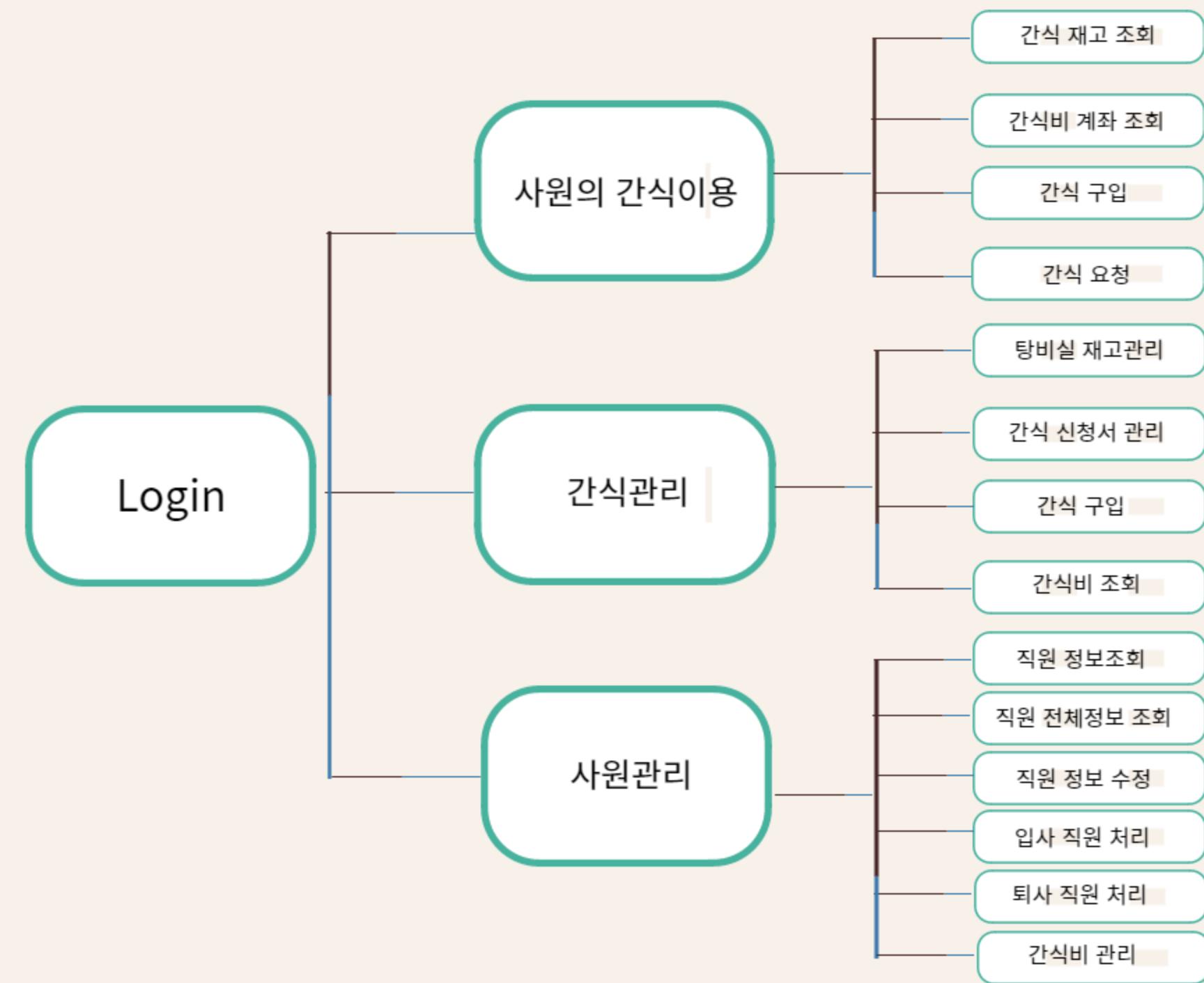
제품 설명서





02 제품 설명서

프로그램 구상도



02 프로젝트상세

주요 사용 구문



주요 사용 구문



Java 사용

많은량의 데이터를 관리 하기 위해 배열 사용

사원 정보, 간식 재고 정보, 간식비예산 내역 등

주요 사용 구문

Login 알고리즘

```
// 시작 페이지  
1개 사용 위치 : hwseo0705 +1  
public void mainMenu() {  
    System.out.println("Together Inc. 홈페이지에 오신것을 환영합니다.");  
  
    Employee employee;  
    while (true) { // 프로그램 종료할 때 까지 나오는 메인 페이지  
        System.out.println("===== 로그인 =====");  
        while (true) { // 아이디가 존재하는지 체크  
            System.out.print("ID: ");  
            String id = sc.nextLine();  
  
            employee = empMgr.searchNumber(id);  
  
            if (employee == null) {  
                System.out.println("존재하지 않는 아이디입니다. 다시입력해주세요");  
            } else {  
                break;  
            }  
        }  
  
        while (true) { // 비밀번호가 맞는지 체크  
            System.out.print("PWD: ");  
            String pwd = sc.nextLine();  
  
            if (pwd.equals(employee.getPassword())) {  
                break;  
            } else {  
                System.out.println("비밀번호를 잘못 입력하셨습니다. 다시 입력해주세요.");  
            }  
        }  
  
        // 로그인 성공했을 경우  
        char rank = '0';  
        if (employee.getNumber().equals("00000")) rank = '1';  
        else if (employee.getNumber().equals("00001")) rank = '2';  
  
        if (rank == '0') emp.employeePage(employee); // 직원 페이지  
    }  
}
```

1

```
else if (rank == '1') { // 사원 관리자 일 경우  
    boolean loop = true;  
    while (loop) { // 사원 관리자의 메인페이지  
        System.out.println("1. 직원");  
        System.out.println("2. 사원 관리자");  
        System.out.println("3. 로그인 페이지로 돌아가기");  
        System.out.print("=> ");  
  
        int choice = sc.nextInt();  
        sc.nextLine();  
  
        switch (choice) {  
            case 1: // 직원 페이지로 이동  
                emp.employeePage(employee);  
                break;  
            case 2: // 사원 관리자 페이지로 이동  
                mgr.EmployeeManagerMenu();  
                break;  
            case 9: // 로그인 페이지로 돌아가기  
                loop = false;  
                break;  
            default:  
                System.out.println("잘못 입력하셨습니다. 다시 입력해주세요.");  
        }  
    }  
}  
} else if (rank == '2') { // 간식 관리자 페이지  
    boolean loop = true;  
    while (loop) { // 간식 관리자의 메인페이지  
        System.out.println("1. 직원");  
        System.out.println("2. 간식 관리자");  
        System.out.println("3. 로그인 페이지로 돌아가기");  
        System.out.print("=> ");  
  
        int choice = sc.nextInt();  
        sc.nextLine();  
  
        switch (choice) {  
            case 1: // 직원 페이지로 이동  
                emp.employeePage(employee);  
                break;  
            case 2: // 사원 관리자 페이지로 이동  
                snackPage.SnackManagerMenu();  
                break;  
            case 9: // 로그인 페이지로 돌아가기  
                loop = false;  
                break;  
            default:  
                System.out.println("잘못 입력하셨습니다. 다시 입력해주세요.");  
        }  
    }  
}
```

4

```
1. 관리자일 경우  
- 사원관리자와 간식관리자는 2가지 메뉴를 이용할 수 있다.  
  
# 사원관리자는  
- 일반직원 메뉴 employeePage()  
- 사원관리 메뉴 EmployeeMangerMenu()  
  
# 간식관리자는  
- 일반직원 메뉴 employeePage()  
- 간식관리 메뉴 SnackMangerMenu()
```

2

1. 입력받은 사원 번호의 존재 여부 확인

- 사원관리자와 간식관리자는 2가지 메뉴를 이용할 수 있다.

2. 입력된 사원 번호와 비밀번호 일치 확인

사원관리자는

- 일반직원 메뉴 employeePage()
- 사원관리 메뉴 EmployeeMangerMenu()

3. 권리자 권한인지 일반사원인지 if문으로 구분

간식관리자는

- 일반직원 메뉴 employeePage()
- 간식관리 메뉴 SnackMangerMenu()

2. 일반사원일 경우

- 일반직원 메뉴 employeePage() 만 이용 가능.

주요 사용 구문

간식 재고관리 알고리즘

```
public class Snack {  
  
    4개 사용 위치  
    private String name; // 간식 이름  
  
    4개 사용 위치  
    private int price; // 간식 가격  
  
    4개 사용 위치  
    private int stock; // 간식 재고  
  
    6개 사용 위치  sokoving  
    public Snack(String name, int price, int stock) {  
        this.name = name;  
        this.price = price;  
        this.stock = stock;  
    }  

```

Snack 클래스의 컨트롤러(탕비실이라고 부름)

Snack[] snacks :

Snack 데이터를 저장할 수 있는 배열

snacks의 길이는 10

(최대 저장할 수 있는 간식의 종류 10개)

count :

snacks에 실제 있는 데이터의 수를 저장

팬트리에 새로 과자가 추가되거나 삭제될 때 수가 변한다)

money :

사원이 간식을 사고 낸 돈을 저장하는 필드

SnackPantry를 생성할 때 snacks 배열을 만들고 테스트 용으로 미리 간식 데이터를 넣어두었다.

필드로 이름, 가격, 재고를 가진

Snack 클래스 정의 Snack를 생성할 때 이름,

가격, 재고를 입력받는다

```
public class SnackPantry {
```

1개 사용 위치

```
public static final int SIZE = 10; // 최대 간식 종류 개수
```

14개 사용 위치

```
private Snack[] snacks; // 텡비실에 있는 간식들을 저장하는 배열
```

8개 사용 위치

```
private int count; // snacks 배열에 실제 있는 데이터 수
```

2개 사용 위치

```
private int money; // 사원이 간식을 사고 내는 돈을 저장하는 필드
```

1개 사용 위치 sokoving +1

```
public SnackPantry() {
```

```
    snacks = new Snack[SIZE];
```

```
    count = 4;
```

```
    snacks[0] = new Snack( name: "에이스", price: 1000, stock: 10);
```

```
    snacks[1] = new Snack( name: "사과", price: 1000, stock: 10);
```

```
    snacks[2] = new Snack( name: "커피", price: 1500, stock: 10);
```

```
    snacks[3] = new Snack( name: "과자", price: 2000, stock: 10);
```

```
}
```

주요 사용 구문

간식 재고관리

```
public SnackProgramMain() {  
    sc = new Scanner(System.in);  
    pantry = new SnackPantry();          // 팬트리 생성  
    request = new SnackRequest(pantry);  
    emp = new EmployeePage(pantry, request);  
    empMgr = new EmployeeManager();  
    mgr = new ManagerPage(empMgr);  
    snackPage = new SnackPage(pantry, request);  
}
```

class SnackProgramMain

메인 프로그램을 생성할 때 팬트리도 같이 생성되며

다른 클래스에도 팬트리를 인수로 넣어 팬트리 데이터를 공유할 수 있게 하였다.

주요 사용 구문

간식 재고관리

```
public class SnackPage {  
    2개 사용 위치  
    Scanner sc = new Scanner(System.in);  
  
    22개 사용 위치  
    private SnackPantry pantry;  
    7개 사용 위치  
    private SnackRequest request;  
    15개 사용 위치  
    private int budget = 100000;  
  
    1개 사용 위치  ↗ hwseo0705 +1  
    public SnackPage(SnackPantry pantry, SnackRequest request) {  
        this.pantry = pantry;  
        this.request = request;  
    }  
}
```

SnackPage의 필드로 SnackPantry pantry를 넣고

SnackPage를 생성할 때 매개변수로 SnackPantry를 받는다

SnackPage의 this.pantry에 SnackProgramMain에서 인수로 받은 pantry를 대입해 데이터를 연동.

주요 사용 구문

간식 재고관리

```
private void managePantryStock() {  
    while (true) {  
        System.out.println("\n# 탕비실 재고 관리 페이지입니다.");  
        System.out.println("=====");  
        pantry.printSnackPantry();  
        System.out.println("=====");  
        System.out.println("1. 재고 데이터 수정");  
        System.out.println("2. 재고 데이터 삭제");  
        System.out.println("3. 간식 추가 구매");  
        System.out.println("4. 간식 반품하기");  
        System.out.println("9. 이전 페이지로 돌아가기");  
    }  
}
```

탕비실 재고 관리 페이지입니다.

=====

과자 이름: 에이스 / 과자 가격: 1000 / 과자 재고: 10

과자 이름: 사과 / 과자 가격: 1000 / 과자 재고: 10

과자 이름: 커피 / 과자 가격: 1500 / 과자 재고: 10

과자 이름: 과자 / 과자 가격: 2000 / 과자 재고: 10

=====

1. 재고 데이터 수정

2. 재고 데이터 삭제

3. 간식 추가 구매

4. 간식 반품하기

9. 이전 페이지로 돌아가기

실행 예시

주요 사용 구문

사원 관리 - 예산내역 저장 및 출력

```
// 6-1번 메뉴 처리

1개 사용 위치 양상나 yangbnag +1 *

public void EmployeeGroupPayment() {
    e.groupPayment();

    DateFormat dateFormat = new SimpleDateFormat( pattern: "yyyy/MM/dd HH:mm:ss");
    Date date = new Date();
    list.insert(dateFormat.format(date), histroy: "전체지급", amount: 50000*e.existEmployee());
}
```

```
// 6-2번 메뉴 처리 특정 사원에게 간식비를 지급

1개 사용 위치 양상나 yangbnag +2

public void specialPayment() {
    System.out.println("===== 보너스 간식비를 지급 합니다.=====");
    String specialEmployee = inputStr( msg: "- 보너스를 지급받을 사원번호를 입력하세요. :");
    int snackCost = inputInt( msg: "지급할 금액을 입력하세요.\n");

    e.payment(specialEmployee, snackCost);

    DateFormat dateFormat = new SimpleDateFormat( pattern: "yyyy/MM/dd HH:mm:ss");
    Date date = new Date();
    list.insert(dateFormat.format(date), e.searchNumber(specialEmployee).getName(), snackCost);
}
```

주요 사용 구문

사원 관리 - 예산내역 저장 및 출력

```
// 간식비 계좌인 budget를 출력  
1개 사용 위치 『hwseo0705 +2  
public int printSnackCost(ListController list) {  
    ListController lc = list;  
    for (List l : lc.printList()) {  
        if (l != null) System.out.println(l.inform());  
    }  
    return budget;  
}
```

```
//6-3번 메뉴 처리 budget 조회  
1개 사용 위치 『yangbnag +1  
public void printBudget() { System.out.printf("=====현재 남은 간식비는 %d 입니다.\n\n", e.printSnackCost(list)); }
```

```
2022/06/07 13:18:31 | 전체지급 | 350000 |  
2022/06/07 13:18:48 | 김지우 | 50000 |  
=====현재 남은 간식비는 600000 입니다.=====
```

02 프로젝트 상세

- 프로그램 시연

- 어려웠던 점



어려웠던 점

- 서로 같은 페이지를 작업할 경우 병합할 때 충돌 문제가 생김
- 프로그램 구현을 위해 클래스를 설계하고
클래스 간 같은 데이터를 쓰게 하는 연동 작업이 어려웠다

03 앞으로의 방향성

기능추가



기능추가

- 현재 타입을 다르게 입력하면 인풋에러가 나며 프로그램 종료되는 문제가 있음. [해결해야 함.]
- 유통기한 체크, 수량이 떨어지면 자동주문, 더 많은 종류의 간식 추가 등 간식 관리 기능 확대.
- 현재 스낵프로그램메인 한 클래스가 데이터 생성과 화면 출력을 동시에 수행하고 있다. 그 기능들을 분리해 객체지향 시스템을 강화시키고 싶다.

04 마치며

조원별 느낀점



느낀점



간단한 프로젝트라 객체 생성/수정하는데
큰 문제가 없었지만 나중에 더욱 복잡한 구조를 가진 프로젝트를 할때는
개발자 간의 충돌도 많이 일어나고, 객체를 이해하고 사용하는데 문제가
많을 것이라는 것을 느꼈습니다.



Thank you!

SNACK MANAGER PROGRAM

서해원

채여운

양재영

