

2017. 10. 17.

2017 년 2 학기
[CSED 232] 객체지향프로그래밍
ASSN 1 보고서

제목:	피자가게 메뉴판
POVIS ID:	dj1122
학과:	산업경영공학과
학번:	20130220
이름:	이다진

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

목차

I. 개요	1
II. 전체 구조와 알고리즘.....	2
1. 전체 구조.....	2
2. 알고리즘	3
3. 클래스 구성	5
III. 프로그램 실행 방법	9
IV. 예제 및 실행 결과	11
V. 실행 결과에 대한 토의.....	11
1. 토론	2
2. 결론	3
3. 개선 방향.....	5

I. 개요

본 프로그램은 C++로 텍스트 파일을 읽고 클래스를 이용하여 "피자가게 메뉴 관리 및 주문"을 구현한다. 프로그램 구현을 통해 C++의 문법과 클래스 이용 방법을 습득한다. 리눅스 및 유사환경에서의 프로그램 실행을 통해 Text User Interface 환경에 익숙해진다. 이러한 과정을 통해 객체 지향 프로그래밍과 실생활의 연관성을 인식한다.

II. 전체 구조와 알고리즘

1. 전체 구조

- (1) "init_menu.txt" 파일을 읽는다.
- (2) 파일의 데이터로 토핑, 도우, 에지의 객체를 생성하고 각각의 리스트에 추가한다.
- (3) 메인 프로그램을 화면에 출력한다.
- (4) 0 입력: 모든 메뉴를 화면에 출력한다.
- (5) 1 입력: 토핑, 도우, 에지 이름을 입력하여 주문한다.
- (6) 2 입력: 메뉴의 종류를 선택하고 필요한 정보를 입력하여 메뉴를 추가한다.
- (7) 3 입력: 메뉴의 종류를 선택하고 메뉴 이름을 입력하여 메뉴를 삭제한다.
- (8) 4 입력: 메뉴의 종류를 선택하고 필요한 정보를 입력하여 메뉴를 검색한다.
- (9) 5 입력: 인기 메뉴를 화면에 출력한다.
- (10) 6 입력: 메인 프로그램을 종료한다.

2. 알고리즘

알고리즘은 전체 구조의 순서대로 작성되었다.

(1) "init_menu.txt" 파일을 읽는다.

- ① ToppingList, DoughList, EdgeList 를 생성한다.
- ② readInput 함수로 파일을 읽는다.

(2) 파일의 데이터로 토핑, 도우, 에지의 객체를 생성하고 각각의 리스트에 추가한다.

1) 토핑

- ① 각 행에서 탭(wt)으로 구분된 토핑의 이름, 모든 재료, 가격, 소개말을 임시 저장한다.
- ② 임시 저장한 데이터를 매개변수로 이용하여 토핑 생성자를 호출한다.
- ③ 토핑 생성자에서 모든 재료를 쉼표(,)로 구분하여 add_ingredient 메소드로 하나씩 재료로 추가한다.
- ④ 생성자 호출이 종료된다.
- ⑤ 리스트의 add 메소드로 리스트에 메뉴를 추가한다.
- ⑥ ①~⑤를 토핑의 개수만큼 반복한다.

2) 도우

- ① 각 행에서 탭(wt)으로 구분된 도우의 이름, 두께, 추가비용, 소개말을 임시 저장한다.
- ② 임시 저장한 데이터를 매개변수로 이용하여 도우 생성자를 호출한다.
- ③ 생성자 호출이 종료된다.
- ④ 리스트의 add 메소드로 리스트에 메뉴를 추가한다.
- ⑤ ①~④를 도우의 개수만큼 반복한다.

3) 에지

- ① 각 행에서 탭(wt)으로 구분된 에지의 이름, 모든 무스, 추가비용, 소개말을 임시 저장한다.
- ② 임시 저장한 데이터를 매개변수로 이용하여 에지 생성자를 호출한다.
- ③ 에지 생성자에서 모든 무스를 쉼표(,)로 구분하여 add_mousse 메소드로 하나씩 무스로 추가한다.
- ④ 생성자 호출이 종료된다.
- ⑤ 리스트의 add 메소드로 리스트에 메뉴를 추가한다.
- ⑥ ①~⑤을 에지의 개수만큼 반복한다.

(3) 메인 프로그램을 화면에 출력한다.

① 메인 프로그램을 화면에 출력한다.

(4) 0 입력: 모든 메뉴를 화면에 출력한다.

- ① 토핑 헤드라인을 화면에 출력한다.
- ② 토핑 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.
- ③ 도우 헤드라인을 화면에 출력한다.
- ④ 도우 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.
- ⑤ 예지 헤드라인을 화면에 출력한다.
- ⑥ 예지 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.

(5) 1 입력: 토핑, 도우, 예지 이름을 입력하여 주문한다.

- ① 토핑 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.
- ② 원하는 메뉴의 이름을 입력 받는다.
- ③ 리스트의 pick 메소드로 해당 메뉴의 주소 값을 반환한다
- ④ 주소 값의 inc_sellcount 메소드로 해당 메뉴의 sell_count 를 1 증가 시킨다.
- ⑤ 도우 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.
- ⑥ ②~④와 동일.
- ⑦ 예지 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.
- ⑧ ②~④와 동일.
- ⑨ 주문 결과를 화면에 출력한다.

(6) 2 입력: 메뉴의 종류를 선택하고 필요한 정보를 입력하여 메뉴를 추가한다.

① 메뉴의 종류를 입력 받는다.

1) 토핑

- ① 토핑의 이름, 재료, 가격, 소개말을 차례로 입력 받고 각각 임시 저장한다.
- ② 임시 저장한 데이터를 매개변수로 이용하여 토핑 생성자를 호출한다.
- ③ 토핑 생성자에서 모든 재료를 쉼표(,)로 구분하여 add_ingredient 메소드로 하나씩 재료로 추가한다.
- ④ 생성자 호출이 종료된다.
- ⑤ 리스트의 add 메소드로 리스트에 메뉴를 추가한다.

- ⑥ 토핑 헤드라인을 화면에 출력한다.
- ⑦ 토핑 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.

2) 도우

- ① 도우의 이름, 두께, 추가비용, 소개말을 차례로 입력 받고 각각 임시 저장한다.
- ② 임시 저장한 데이터를 매개변수로 이용하여 도우 생성자를 호출한다.
- ③ 생성자 호출이 종료된다.
- ④ 리스트의 add 메소드로 리스트에 메뉴를 추가한다.
- ⑤ 도우 헤드라인을 화면에 출력한다.
- ⑥ 도우 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.

3) 에지

- ① 에지의 이름, 무스, 추가비용, 소개말을 차례로 입력 받고 각각 임시 저장한다.
- ② 임시 저장한 데이터를 매개변수로 이용하여 에지 생성자를 호출한다.
- ③ 에지 생성자에서 모든 무스를 쉼표(,)로 구분하여 add_mousse 메소드로 하나씩 무스로 추가한다.
- ④ 생성자 호출이 종료된다.
- ⑤ 리스트의 add 메소드로 리스트에 메뉴를 추가한다.
- ⑥ 에지 헤드라인을 화면에 출력한다.
- ⑦ 에지 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.

(7) 3 입력: 메뉴의 종류를 선택하고 메뉴 이름을 입력하여 메뉴를 삭제한다.

- ① 메뉴의 종류를 입력 받는다.
- ② 삭제하고자 하는 메뉴의 이름을 입력 받는다.
- ③ 리스트의 del 메소드로 리스트에서 메뉴를 삭제한다.
- ④ 메뉴 헤드라인을 화면에 출력한다.
- ⑤ 메뉴 리스트의 show_all_menu 로 모든 메뉴를 화면에 출력한다.

(8) 4 입력: 메뉴의 종류를 선택하고 필요한 정보를 입력하여 메뉴를 검색한다.

- ① 메뉴의 종류를 입력 받는다.

1) 토핑

1-1) 재료로 검색

- ① 재료 이름을 입력받는다.
- ② 리스트의 `search_by_ingred` 메소드로 리스트에서 해당하는 메뉴들을 모두 화면에 출력한다.

1-2) 가격 범위로 검색

- ① 가격의 하한과 상한을 입력받는다.
- ② 리스트의 `search_by_price` 메소드로 리스트에서 해당하는 메뉴들을 모두 화면에 출력한다.

2) 도우

2-1) 두께로 검색

- ① 두께의 하한과 상한을 입력받는다.
- ② 리스트의 `search_by_thick` 메소드로 리스트에서 해당하는 메뉴들을 모두 화면에 출력한다.

3) 에지

1-1) 무스로 검색

- ① 무스 이름을 입력받는다.
- ② 리스트의 `search_by_mousse` 메소드로 리스트에서 해당하는 메뉴들을 모두 화면에 출력한다.

(9) 5 입력: 인기 메뉴를 화면에 출력한다.

- ① 토핑 리스트의 `show_most_popular` 메소드로 리스트에서 `sell_count` 값이 가장 높은 토핑을 화면에 출력한다.
- ② 도우 리스트의 `show_most_popular` 메소드로 리스트에서 `sell_count` 값이 가장 높은 도우를 화면에 출력한다.
- ③ 에지 리스트의 `show_most_popular` 메소드로 리스트에서 `sell_count` 값이 가장 높은 에지를 화면에 출력한다.

(10) 6 입력: 메인 프로그램을 종료한다

- ① 토핑 리스트가 가리키는 모든 Topping 을 삭제하는 동시에 메모리를 `delete` 로 반환한다.
- ② 도우 리스트가 가리키는 모든 Dough 를 삭제하는 동시에 메모리를 `delete` 로 반환한다.
- ③ 토핑 리스트가 가리키는 모든 Edge 를 삭제하는 동시에 메모리를 `delete` 로 반환한다.
- ④ 메인 프로그램의 `while` 문을 종료한다.

3. 클래스 구성

(1) Topping

- ① name: 토핑의 이름
- ② ingredient: 토핑의 재료. 최대 10 개.
- ③ price: 토핑의 가격
- ④ sell_count: 토핑의 판매 횟수
- ⑤ comment: 토핑의 소개말
- ⑥ next: 토핑 리스트에서 다음 토핑을 가리키는 Topping 형 포인터 변수
- ⑦ get method: ①~⑥을 리턴하는 메소드
- ⑧ set_method: name, price, comment, next 의 값을 입력받아 각각의 멤버 변수에 저장하는 메소드
- ⑨ add_ingredient: 토핑의 재료를 하나씩 입력받아 ingredient 배열에 추가하는 메소드.
- ⑩ inc_sellcount: 기존의 sell_count 에 1 을 더하는 메소드.
- ⑪ 생성자: 토핑의 이름, 모든 재료, 가격, 소개말을 입력받아 Topping 객체를 생성한다.

(2) ToppingList

- ① head: 리스트의 첫번째 인스턴스 주소를 저장하는 Topping 형 포인터 변수.
- ② size: 현재 리스트의 크기(총 인스턴스의 수).
- ③ add: 해당 인스턴스의 주소를 받아, 리스트에 해당 메뉴를 추가하는 메소드.
- ④ del: 메뉴 이름을 입력 받아, 해당 메뉴를 리스트로부터 제거하는 메소드.
- ⑤ pick: 메뉴 이름을 입력 받아, 해당 메뉴의 주소를 리턴하는 메소드.
- ⑥ show_all_menu: sell_count 를 제외한 모든 메뉴의 내용을 화면에 출력하는 메소드.
- ⑦ show_most_popular: 가장 많이 팔린 메뉴의 sell_count 를 제외한 내용을 화면에 출력하는 메소드.
- ⑧ seach_by_ingred: 재료명을 입력 받아, 해당 재료를 사용하는 모든 메뉴들의 내용을 화면에 출력하는 메소드.
- ⑨ seach_by_price: 가격의 하한선과 상한선을 입력 받아, 범위 내의 가격을 가진 모든 메뉴들의 내용을 화면에 출력하는 메소드.
- ⑩ get_size(사용자 정의 함수): size 를 리턴하는 메소드. size 가 화면에 출력되는 변수이므로 필요하다고 판단하였다. show_all_menu 메소드의 헤드라인에 사용된다.
- ⑪ get_head(사용자 정의 함수): head 를 리턴하는 메소드. head 는 리스트가 가리키는 Topping 객체의 주소 값이다. 메인 함수에서 프로그램 종료 직전 리스트 내의 동적 할당받은 객체를 모두 delete 로 해제 해야한다. 이때 메인 함수가 private 멤버인 head 에 접근하기 위해 필요하다고 판단하였다.

(3) Dough

- ① name: 도우의 이름.
- ② thickness: 도우의 두께.
- ③ extra_price: 도우의 추가비용.
- ④ sell_count: 도우의 판매 횟수.
- ⑤ comment: 도우의 소개말.
- ⑥ next: 도우 리스트에서 다음 도우를 가리키는 Dough 형 포인터 변수.

- ⑦ get method: ①~⑥을 리턴하는 메소드
- ⑧ set_method: 각각 name, extra_cost, comment, next 를 입력받아 해당 멤버 변수에 저장하는 메소드.
- ⑨ inc_sellcount: 기존의 sell_count 에 1 을 더하는 메소드.
- ⑩ 생성자: 도우의 이름, 두께, 추가비용, 소개말을 입력받아 Dough 객체를 생성한다.

(4) DoughList

- ① head: 리스트의 첫번째 인스턴스 주소를 저장하는 Dough 형 포인터 변수.
- ② size: 현재 리스트의 크기(총 인스턴스의 수).

- ③ add: 해당 인스턴스의 주소를 받아, 리스트에 해당 메뉴를 추가하는 메소드.
- ④ del: 메뉴 이름을 입력 받아, 해당 메뉴를 리스트로부터 제거하는 메소드.
- ⑤ pick: 메뉴 이름을 입력 받아, 해당 메뉴의 주소를 리턴하는 메소드.
- ⑥ show_all_menu: sell_count 를 제외한 모든 메뉴의 내용을 화면에 출력하는 메소드.
- ⑦ show_most_popular: 가장 많이 팔린 메뉴의 sell_count 를 제외한 내용을 화면에 출력하는 메소드.
- ⑧ seach_by_thick: 두께의 하한선과 상한선을 입력 받아, 범위 내의 두께를 가진 모든 메뉴들의 내용을 화면에 출력하는 메소드.
- ⑨ get_size(사용자 정의 함수): size 를 리턴하는 메소드. size 가 화면에 출력되는 변수이므로 필요하다고 판단하였다. show_all_menu 메소드의 헤드라인에 사용된다.
- ⑩ get_head(사용자 정의 함수): head 를 리턴하는 메소드. head 는 리스트가 가리키는 Dough 객체의 주소 값이다. 메인 함수에서 프로그램 종료 직전 리스트 내의 동적 할당받은 객체를 모두 delete 로 해제 해야한다. 이때 메인 함수가 private 멤버인 head 에 접근하기 위해 필요하다고 판단하였다.

(5) Edge

- ① name: 에지의 이름.
- ② mousse: 에지의 무스. 최대 3 개.
- ③ extra_price: 에지의 추가비용.
- ④ sell_count: 에지의 판매 횟수.
- ⑤ comment: 에지의 소개말.
- ⑥ next: 에지 리스트에서 다음 에지를 가리키는 Edge 형 포인터 변수.

- ⑦ get method: ①~⑥을 리턴하는 메소드
- ⑧ set_method: 각각 name, extra_cost, comment, next 를 입력받아 해당 멤버 변수에 저장하는 메소드.
- ⑨ add_mousse: 에지의 무스를 하나씩 입력받아 mousse 배열에 추가하는 메소드.
- ⑩ inc_sellcount: 기존의 sell_count 에 1 을 더하는 메소드.
- ⑪ 생성자: 에지의 이름, 모든 무스, 추가비용, 소개말을 입력받아 Edge 객체를 생성한다.

(6) EdgeList

- ① head: 리스트의 첫번째 인스턴스 주소를 저장하는 Edge 형 포인터 변수.
- ② size: 현재 리스트의 크기(총 인스턴스의 수).

- ③ add: 해당 인스턴스의 주소를 받아, 리스트에 해당 메뉴를 추가하는 메소드.
- ④ del: 메뉴 이름을 입력 받아, 해당 메뉴를 리스트로부터 제거하는 메소드.
- ⑤ pick: 메뉴 이름을 입력 받아, 해당 메뉴의 주소를 리턴하는 메소드.
- ⑥ show_all_menu: sell_count 를 제외한 모든 메뉴의 내용을 화면에 출력하는 메소드.
- ⑦ show_most_popular: 가장 많이 팔린 메뉴의 sell_count 를 제외한 내용을 화면에 출력하는 메소드.
- ⑧ seach_by_mousse: 무스명을 입력 받아, 해당 무스를 사용하는 모든 메뉴들의 내용을 화면에 출력하는 메소드.
- ⑨ get_size(사용자 정의 함수): size 를 리턴하는 메소드. size 가 화면에 출력되는 변수이므로 필요하다고 판단하였다. show_all_menu 메소드의 헤드라인에 사용된다.
- ⑩ get_head(사용자 정의 함수): head 를 리턴하는 메소드. head 는 리스트가 가리키는 Edge 객체의 주소 값이다. 메인 함수에서 프로그램 종료 직전 리스트 내의 동적 할당받은 객체를 모두 delete 로 해제 해야한다. 이때 메인 함수가 private 멤버인 head 에 접근하기 위해 필요하다고 판단하였다.

Ⅲ. 프로그램 실행 방법

본 프로그램은 MacOS 의 터미널에서 실행된다. Makefile 의 내용은 다음과 같다.

- ① main.cpp 파일을 제외한 나머지 6 개 .cpp 파일을 컴파일하고 각각의 .hpp 파일과 링킹하여 각각의 .o 파일을 생성한다.
- ② main.cpp 파일을 컴파일하고 6 개의 .hpp 파일과 링킹하여 main.o 파일을 생성한다.
- ③ main.o 와 6 개의 .o 파일을 링킹하여 assn1 실행 파일을 생성한다.

(1) 터미널을 실행한다.

(2) 소스 파일과 Makefile 이 모두 있는 디렉토리로 이동한다.

```
$cd <디렉토리명>
```

(3) 터미널 입력 라인에서 "make"를 입력하면 "make all"과 같은 효과가 나타나서 assn1 실행 파일이 생성된다.

```
$make
```

(4) "init_menu.txt"을 인자로 넣어 assn1 을 실행한다.

```
$/assn1 init_menu.txt
```

IV. 예제 및 실행 결과

0. 메인 프로그램 시작 및 모든 메뉴 보기

```
TEST — assn1 init_menu.txt — 105x25
>> 0
-TOPPING <3>
[이름 :슈퍼슈프림][재료 :피망,양파,파프리카,버섯][가격 :23500원][소개말 :가장 평범한 피자]
[이름 :불고기][재료 :불고기,피망,양파,파프리카,버섯][가격 :24000원][소개말 :불고기 향이 강한 피자]
[이름 :베이컨포테이토][재료 :웨이감자,베이컨,피망,양파,버섯][가격 :25000원][소개말 :베이컨과 감자 맛 피자]
-DOUGH <3>
[이름 :씬][두께 :0.5mm][추가비용 :0원][소개말 :토핑 맛을 살려줌]
[이름 :팬][두께 :2mm][추가비용 :0원][소개말 :배부름]
[이름 :프리미엄][두께 :1mm][추가비용 :5000원][소개말 :비쌌]
-EDGE <4>
[이름 :리치골드][무스 :고구마][추가비용 :1000원][소개말 :고구마와 스트링 치즈맛]
[이름 :리치골드엘로][무스 :고구마,감자][추가비용 :1000원][소개말 :감자와 스트링 치즈맛]
[이름 :치즈크러스트][무스 :][추가비용 :500원][소개말 :스트링 치즈맛]
[이름 :치즈바이트][무스 :고구마][추가비용 :1000원][소개말 :고구마 맛 약간과 스트링 치즈맛]

0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료

>> |
```

1. 주문하기

```
TEST — assn1 init_menu.txt — 105x36
>> 1
-TOPPING <3>
[이름 :슈퍼슈프림][재료 :피망,양파,파프리카,버섯][가격 :23500원][소개말 :가장 평범한 피자]
[이름 :불고기][재료 :불고기,피망,양파,파프리카,버섯][가격 :24000원][소개말 :불고기 향이 강한 피자]
[이름 :베이컨포테이토][재료 :웨이감자,베이컨,피망,양파,버섯][가격 :25000원][소개말 :베이컨과 감자 맛 피자]

원하시는 토핑 이름을 입력하세요.
>> 베이컨포테이토
-DOUGH <3>
[이름 :씬][두께 :0.5mm][추가비용 :0원][소개말 :토핑 맛을 살려줌]
[이름 :팬][두께 :2mm][추가비용 :0원][소개말 :배부름]
[이름 :프리미엄][두께 :1mm][추가비용 :5000원][소개말 :비쌌]

원하시는 도우 이름을 입력하세요.
>> 씬
-EDGE <4>
[이름 :리치골드][무스 :고구마][추가비용 :1000원][소개말 :고구마와 스트링 치즈맛]
[이름 :리치골드엘로][무스 :고구마,감자][추가비용 :1000원][소개말 :감자와 스트링 치즈맛]
[이름 :치즈크러스트][무스 :][추가비용 :500원][소개말 :스트링 치즈맛]
[이름 :치즈바이트][무스 :고구마][추가비용 :1000원][소개말 :고구마 맛 약간과 스트링 치즈맛]

원하시는 에지 이름을 입력하세요.
>> 리치골드

주문하신 пи자는 "씬 리치골드 베이컨포테이토" 입니다.
가격은 "26000"원 입니다.

0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료

>> |
```

2-1. 메뉴 추가(토핑)

```
TEST — assn1 init_menu.txt — 105x31
>> 2
추가하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 예지
>> 1
이름
>> 엄청난토핑
재료
>> 피망,양파,불고기,치즈
가격
>> 9999
소개말
>> 엄청난 맛이다
메뉴 추가가 완료되었습니다 .

-TOPPING <4>
[이름 :슈퍼슈프림][재료 :피망,양파,파프리카,버섯][가격 :23500원][소개말 :가장 평범한 피자]
[이름 :불고기][재료 :불고기,피망,양파,파프리카,버섯][가격 :24000원][소개말 :불고기 향이 강한 피자]
[이름 :베이컨포테이토][재료 :웨이감자,베이컨,피망,양파,버섯][가격 :25000원][소개말 :베이컨과 감자 맛 피자]
[이름 :엄청난 토핑][재료 :피망,양파,불고기,치즈][가격 :9999원][소개말 :엄청난 맛이다]

0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료

>> |
```

2-2. 메뉴 추가(도우)

```
TEST — assn1 init_menu.txt — 105x31
>> 2
추가하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 예지
>> 2
이름
>> 엄청난도우
두께
>> 100
추가비용
>> 999
소개말
>> 두꺼움
메뉴 추가가 완료되었습니다 .

-DOUGH <4>
[이름 :썬][두께 :0.5mm][추가비용 :0원][소개말 :토핑 맛을 살려줌]
[이름 :팬][두께 :2mm][추가비용 :0원][소개말 :배부름]
[이름 :프리미엄][두께 :1mm][추가비용 :5000원][소개말 :비쌈]
[이름 :엄청난 도우][두께 :100mm][추가비용 :999원][소개말 :두꺼움]

0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료

>> |
```

2-3. 메뉴 추가(에지)

```
TEST — assn1 init_menu.txt — 105x32
>> 2
추가하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 에지
>> 3
이름
>> 엄청난에지
무스
>> 치즈,체다치즈,모짜렐라치즈
추가비용
>> 500
소개말
>> 치즈 폭발
메뉴 추가가 완료되었습니다 .

-EDGE <5>
[이름 :리치골드][무스 :고구마][추가비용 :1000원][소개말 :고구마와 스트링 치즈맛]
[이름 :리치골드옐로][무스 :고구마,감자][추가비용 :1000원][소개말 :감자와 스트링 치즈맛]
[이름 :치즈크러스트][무스 :][추가비용 :500원][소개말 :스트링 치즈맛]
[이름 :치즈바이트][무스 :고구마][추가비용 :1000원][소개말 :고구마 맛 약간과 스트링 치즈맛]
[이름 :엄청난에지][무스 :치즈,체다치즈,몰라치즈][추가비용 :500원][소개말 :치즈폭발]

=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료
=====
>> |
```

3-1. 메뉴 삭제(토핑)

```
TEST — assn1 init_menu.txt — 105x28
>> 3
삭제하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 에지
>> 1
[이름 :슈퍼슈프림][재료 :피망,양파,파프리카,버섯][가격 :23500원][소개말 :가장 평범한 피자]
[이름 :불고기][재료 :불고기,피망,양파,파프리카,버섯][가격 :24000원][소개말 :불고기 향이 강한 피자]
[이름 :베이컨포테이토][재료 :웨이감자,베이컨,피망,양파,버섯][가격 :25000원][소개말 :베이컨과 감자 맛 피자]
[이름 :엄청난토핑][재료 :피망,양파,불고기,치즈][가격 :9999원][소개말 :엄청난 맛이다]
삭제하고자 하는 메뉴 이름을 입력하세요 .
>> 슈퍼슈프림
메뉴 삭제가 완료되었습니다 .

-TOPPING <3>
[이름 :불고기][재료 :불고기,피망,양파,파프리카,버섯][가격 :24000원][소개말 :불고기 향이 강한 피자]
[이름 :베이컨포테이토][재료 :웨이감자,베이컨,피망,양파,버섯][가격 :25000원][소개말 :베이컨과 감자 맛 피자]
[이름 :엄청난토핑][재료 :피망,양파,불고기,치즈][가격 :9999원][소개말 :엄청난 맛이다]

=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료
=====
>> |
```

3-2. 메뉴 삭제(도우)

```
TEST — assn1 init_menu.txt — 105x28
>> 3
삭제하고자 하는 메뉴의 종류를 선택하세요.
1: 토핑
2: 도우
3: 에지
>> 2
[이름:썬][두께:0.5mm][추가비용:0원][소개말:토핑 맛을 살려줌]
[이름:팬][두께:2mm][추가비용:0원][소개말:배부름]
[이름:프리미엄][두께:1mm][추가비용:5000원][소개말:비쌈]
[이름:엄청난도우][두께:100mm][추가비용:999원][소개말:두꺼움]
삭제하고자 하는 메뉴 이름을 입력하세요.
>> 팬
메뉴 삭제가 완료되었습니다.

-DOUGH <3>
[이름:썬][두께:0.5mm][추가비용:0원][소개말:토핑 맛을 살려줌]
[이름:프리미엄][두께:1mm][추가비용:5000원][소개말:비쌈]
[이름:엄청난도우][두께:100mm][추가비용:999원][소개말:두꺼움]
=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기 메뉴 추천
6: 프로그램 종료
=====
>> █
```

3-3. 메뉴 삭제(에지)

```
TEST — assn1 init_menu.txt — 105x25
>> 3
[이름:리치골드][무스:고구마][추가비용:1000원][소개말:고구마와 스트링 치즈맛]
[이름:리치골드엘로][무스:고구마,감자][추가비용:1000원][소개말:감자와 스트링 치즈맛]
[이름:치즈크러스트][무스:][추가비용:500원][소개말:스트링 치즈맛]
[이름:치즈바이트][무스:고구마][추가비용:1000원][소개말:고구마 맛 약간과 스트링 치즈맛]
[이름:엄청난에지][무스:치즈,체다치즈,렐라치즈][추가비용:500원][소개말:치즈폭발]
삭제하고자 하는 메뉴 이름을 입력하세요.
>> 리치골드엘로
메뉴 삭제가 완료되었습니다.

-EDGE <4>
[이름:리치골드][무스:고구마][추가비용:1000원][소개말:고구마와 스트링 치즈맛]
[이름:치즈크러스트][무스:][추가비용:500원][소개말:스트링 치즈맛]
[이름:치즈바이트][무스:고구마][추가비용:1000원][소개말:고구마 맛 약간과 스트링 치즈맛]
[이름:엄청난에지][무스:치즈,체다치즈,렐라치즈][추가비용:500원][소개말:치즈폭발]
=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기 메뉴 추천
6: 프로그램 종료
=====
>> █
```

4-1-1. 메뉴 검색(토핑 재료로 검색)

```
TEST — assn1 init_menu.txt — 105x25

>> 4
검색하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 애지
>> 1
검색 기준을 선택하세요 .
1: 재료로 검색
2: 가격으로 검색
>> 1
재료 이름을 입력해주세요 .
>> 불고기
[이름:불고기][재료:불고기,피망,양파,파프리카,버섯][가격:24000원][소개말:불고기 향이 강한 피자]
[이름:엄청난 토핑][재료:피망,양파,불고기,치즈][가격:9999원][소개말:엄청난 맛이다]
이상 총 2개가 검색되었습니다 .

=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기 메뉴 추천
6: 프로그램 종료
=====
>> █
```

4-1-2. 메뉴 검색(토핑 가격으로 검색)

```
TEST — assn1 init_menu.txt — 105x28

>> 4
검색하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 애지
>> 1
검색 기준을 선택하세요 .
1: 재료로 검색
2: 가격으로 검색
>> 2
가격 범위 중 하한을 입력해주세요 .
>> 0
가격 범위 중 상한을 입력해주세요 .
>> 1000000
[이름:슈퍼슈프림][재료:피망,양파,파프리카,버섯][가격:23500원][소개말:가장 평범한 피자]
[이름:불고기][재료:불고기,피망,양파,파프리카,버섯][가격:24000원][소개말:불고기 향이 강한 피자]
[이름:베이컨포테이토][재료:웨이감자,베이컨,피망,양파,버섯][가격:25000원][소개말:베이컨과 감자 맛 피자]
이상 총 3개가 검색되었습니다 .

=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기 메뉴 추천
6: 프로그램 종료
=====
>> █
```


4-2-1. 메뉴 검색(도우 두께로 검색)

```
TEST — assn1 init_menu.txt — 105x25
>> 4
검색하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 에지
>> 2
검색 기준을 선택하세요 .
1: 두께로 검색
>> 1
두께 범위 중 하한을 입력해주세요 .
>> 1.2
두께 범위 중 상한을 입력해주세요 .
>> 2.2
[이름:팬][두께:2mm][추가비용:0원][소개말:배부름]
이상 총 1개가 검색되었습니다 .

=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기 메뉴 추천
6: 프로그램 종료
=====
>> |
```

4-3-1. 메뉴 검색(에지 무스로 검색)

```
TEST — assn1 init_menu.txt — 105x23
>> 4
검색하고자 하는 메뉴의 종류를 선택하세요 .
1: 토핑
2: 도우
3: 에지
>> 3
검색 기준을 선택하세요 .
1: 무스로 검색
>> 1
무스 이름을 입력해주세요 .
>> 감자
[이름:리치글드엘로][무스:고구마,감자][추가비용:1000원][소개말:감자와 스트링 치즈맛]
이상 총 1개가 검색되었습니다 .

=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기 메뉴 추천
6: 프로그램 종료
=====
>> |
```

5. 인기 메뉴 추천

```
TEST — assn1 init_menu.txt — 105x17
>> 5
-TOPPING 인기메뉴
[이름:불고기][재료:불고기,피망,양파,파프리카,버섯][가격:24000원][소개말:불고기 향이 강한 피자]
-DOUGH 인기메뉴
[이름:썬][두께:0.5mm][추가비용:0원][소개말:토핑 맛을 살려줌]
-EDGE 인기메뉴
[이름:치즈바이트][무스:고구마][추가비용:1000원][소개말:고구마 맛 약간과 스트링 치즈맛]
=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료
=====
>> █
```

6. 메인 프로그램 종료

```
TEST — -bash — 105x12
=====
0: 모든 메뉴 보기
1: 주문하기
2: 메뉴 추가
3: 메뉴 삭제
4: 메뉴 검색
5: 인기메뉴 추천
6: 프로그램 종료
=====
>> 6
프로그램을 종료합니다...
rubyui-MacBook-Pro:TEST ruby$ █
```

V. 실행 결과에 대한 토의

1. 토론

(1) 리스트에서 모든 메뉴를 탐색하기 위한 반복문 작성 시 발생한 문제

```
void ToppingList::search_by_price(int min_price, int max_price){
    ToppingList tempList;
    Topping * tempPtr = this->head;
    while (true) {
        if(tempPtr->get_price() >= min_price && tempPtr->get_price() <=
max_price)
            tempList.add(tempPtr);
        tempPtr = tempPtr->get_next();
        if(tempPtr == NULL) // 추가한 부분
            break;          // 추가한 부분
    }
    tempList.show_all_menu();
    cout << "이상 총 " << tempList.size << "개가 검색되었습니다." << endl;
}
```

위의 메소드는 1)리스트에 있는 토핑 중 조건에 맞는 토핑을 임시 리스트에 저장하고 2)리스트의 show_all_menu 메소드를 호출한다. 모든 토핑을 검사하기 위하여 tempPtr 는 while 문이 반복될 때마다 다음 토핑을 가리킨다.

해당 메소드에 대한 테스트 중 while 문을 빠져나가지 못하는 문제가 발생하였다. 이때 tempPtr 이 마지막 토핑을 가리킬 때 tempPtr 의 next 는 NULL 을 가리킨다는 점을 이용해 반복문 탈출을 위한 if 문을 추가하였다.

(2) 메소드 호출 시 잘못된 타입의 매개변수를 입력한 문제

```
void DoughList::search_by_thick(float min_thick, float max_thick){...}
```

위의 메소드의 매개변수 타입은 float 이다. 하지만 메인 프로그램에서 메소드를 호출할 때 매개변수 자리에 int 타입의 변수를 대입하여 에러가 발생하였다.

이 문제는 해결하기 쉽지만 문제 자체를 발견하는 데에 많은 시간이 걸렸다. int 형은 숫자를 저장하기 위해 가장 자주 쓰이는 데이터 타입이다. int 형에 익숙해져서 int 형이 문제가 될 것이라고 생각하지 못한 것이다. 코드를 반복해서 검토한 후 문제를 발견하였다.

(3) string ingredient[10], string mousse[3] 입력 받기

```
cout << "재료" << endl << ">> "; cin >> ingredient_all;
...
toppingList.add(new Topping(name, ingredient_all, price, comment));
```

<1. 메인 함수에서 재료 입력을 위한 코드>

```
int add_count = 0;
int start_index = 0;
int found_index = 0;
for (int i = 0; i < ingredient_all.length(); i++) {
    if (ingredient_all[i] == ',') {
        found_index = i;
        add_ingredient(ingredient_all.substr(start_index,
        found_index-start_index)); add_count++;
        start_index = found_index + 1;
    }
    add_ingredient(ingredient_all.substr(start_index,
    ingredient_all.length() - start_index)); add_count++;
    if (add_count > 10)
        cout << "입력받은 재료는 10 개까지만 추가됩니다." << endl;
```

<2. 생성자 내에서 재료 입력을 위한 코드>

Topping 의 ingredient 나 Edge 의 mousse 는 쉼표(,)로 구분된 여러 개의 문자열을 각각 배열에 추가해야 한다. 수정 전에는 위의 기능을 구현하는 코드를 모두 메인 함수에 작성하여 메인 함수의 가독성이 낮았다.

프로그램 소스 코드를 완성해가면서, 메인 함수는 프로그램의 전체 구성을 보기 쉽게 나타내야 할 의무도 있다고 판단했다. 따라서 메인 함수에서의 재료 입력 처리는 코드 2 줄로 표현하고, 직접적인 구현은 생성자 내에서 처리하도록 수정하였다.

2. 결론

(1) 소스 파일 분리의 중요성

CS101 수강 중에는 소스 파일을 분리하는 것을 번거롭고 불필요하다고 여겼다. 하지만 본 프로그램을 작성하면서 소스 파일 분리의 필요성을 경험했다. 1)클래스마다 존재 이유와 기능이 명확히 구분되며 2)코드의 양이 많았기 때문이다. 특히 소스 파일을 분리하지 않았다면 코드를 수정할 때 몇백 줄이 넘는 모든 코드를 훑어야만 했을 것이다.

(2) STL 의 편리함

링크드 리스트의 개념을 이미 알고 있었기 때문에 STL(Standard Template Library)을 이용하지 않아도 금방 add, del, pick 등의 메소드를 구현할 수 있을 예상했다. 첫 소스 코드 작성

은 쉬웠지만 1)노드가 없을 때 2)첫번째 노드일 때 3)마지막 노드일 때 등의 모든 경우를 고려한 완성도 있는 메소드를 작성하는 것은 쉬운 일이 아니었다. 그리고 size 를 증감하는 부분을 빼먹는 등의 실수로 STL 을 사용하면 일어나지 않았을 오류들 때문에 프로그램 완성 시간이 길어졌다.

(3) Makefile 의 유용함

소스 코드는 통합개발도구 Xcode 를 통해 작성하였다. 이때 Xcode 에서 한국어 입력이 이루어지지 않아 Xcode 에서는 dump 를 받아 디버깅을 하고 코드를 수정하기만 하고 주로 터미널에서 프로그램을 실행했다. Xcode 에서는 단축키 하나면 컴파일 및 실행이 완료되는데 터미널에서는 일일이 명령어를 입력해야 하므로 번거로웠다. 그런데 Makefile 사용법을 익히니 컴파일 및 실행파일 생성이 명령어 한 줄에 이루어졌다. 실행은 `./<실행파일명> <텍스트파일명>` 을 입력해야 하지만 한 번 입력하고 나면 화살표 ↑ 키를 눌러 명령어를 불러올 수 있었으므로 단축키만큼 쉽고 간편했다.

3. 개선 방향

(1) 토픽, 도우, 에지를 상속하는 상위 클래스 이용

토픽, 도우, 에지는 아예 다른 사물을 표현하므로 각각 다른 클래스로 구현해야 한다. 하지만 각각의 클래스를 이루는 멤버 변수 및 멤버 메소드가 중복되는 부분이 많았다. 1)이러한 중복 부분이 작성된 상위 클래스를 하나 만들고 2)토픽, 도우, 에지의 클래스가 각각 상속받는다. 그러면 중복 부분이 사라져 코드의 양이 훨씬 줄어들 것이다. 또한 코드의 수정이 모든 클래스가 아닌 상위 클래스에서만 이루어지므로 유지보수에 용이할 것이다.

(2) search_except_ingred 메소드 추가

본 프로그램에는 토픽에 들어가는 재료명을 검색해 메뉴를 찾아주는 기능이 있다. 그런데 피망을 싫어하거나 올리브 알러지가 있는 사람들은 해당 재료가 들어가지 않은 메뉴판을 원할 것이다. 즉 search_by_ingred 메소드가 해당 재료를 포함하는 모든 메뉴를 출력했다면, search_except_ingred 메소드는 해당 재료를 포함하지 않는 모든 메뉴를 출력한다.