

C++ 프로그래밍 및 실습

# 레스토랑 청소 우선 순위 프로그램

진척 보고서 #1

제출일자: 2024-11-17

제출자명: 장유은

제출자학번: 234199

## **1. 프로젝트 목표 (16 pt)**

### **1) 배경 및 필요성 (14 pt)**

레스토랑에서 손님이 퇴점하면 해당 자리를 청소해야 함. 하지만 레스토랑이 넓고 손님이 많아 퇴점한 자리인지 육안으로 확인이 어려움. 손님이 퇴점한 뒤 청소해야 할 자리를 알려주는 프로그램이 필요함.

### **2) 프로젝트 목표**

현재 테이블에 손님이 있는지, 퇴점했는지 확인하고 청소해야 할 테이블을 알려주는 프로그램을 만드는 것을 목표로 함.

### **3) 차별점**

퇴점한 테이블이 여러 개일 경우, 손님들의 선호도가 높은 좌석의 테이블을 먼저 청소하도록 우선순위를 알려줌.

## **2. 기능 계획**

### **1) 기능 1 테이블 클래스, 입/퇴점 함수**

- 설명 : 테이블 클래스를 만들어 번호, 점유 여부, 선호도를 입력받는다. 입점한 테이블을 객체배열로 저장하는 함수를 만들고 퇴점한 테이블도 객체배열로 저장하는 함수를 만든다.

### **2) 기능 2 입력받고 알려주기**

- 설명 : 입점을 입력할지, 퇴점을 입력할지, 청소순위를 알고 싶은지 터미널에서

입력받고 알려주는 기능

### 3) 기능 3 청소 우선순위

- 설명 : 퇴점한 테이블 중 선호도가 높은 테이블을 분석하고 청소의 우선순위를 알려준다.

## 3. 진척사항

### 1) 기능 구현

#### (1) Table 객체

- 설명 : 테이블 번호, 점유 여부, 손님 선호도를 저장하는 객체
- 적용된 배운 내용 : 클래스
- 코드 스크린샷

```
4  class Table{
5  public:
6      int number; // 테이블 번호
7      bool occupy; // 테이블 점유 여부
8      int preference; // 손님 선호도(높을수록 우선순위 높음)
9
10     Table(){ // 기본 생성자
11         number=0;
12         occupy=false;
13         preference=0;
14     }
15     Table(int n,bool o,int p){
16         number=n;
17         occupy=o;
18         preference=p;
19     }
20 };
```

#### (2) Table 추가함수

- 설명 : 객체배열에 테이블 추가

- 적용된 배운 내용 : 객체 배열, 함수, 조건문
- 코드 스크린샷

```

22  const int maxTable=9;
23  Table tables[maxTable]; // 테이블 관리하기 위한 객체 배열
24  int tableCount=0; // 테이블 개수 세는 변수
25
26  // 테이블 추가 함수
27  void addTable(int n,bool o,int p){
28      if(tableCount<maxTable)
29          tables[tableCount++]=Table(n,o,p);
30      else
31          cout<<"빈 테이블이 없습니다."<<endl;
32  }

```

### (3) Table 퇴점 처리 함수

- 설명 : 퇴점한 테이블의 점유 여부를 false로 바꾸고 퇴점 처리된 테이블을 찾아 새로운 객체 배열에 저장
- 적용된 배운 내용 : 객체 배열, 함수, 조건문, 반복문
- 코드 스크린샷

```

34  // 테이블 퇴점 처리하는 함수
35  void outTable(int n){
36      tables[n-1].occupy=false; // 퇴점 테이블의 점유 여부를 false로 바꿈
37  }
38  // 퇴점 된 테이블 중 우선순위 알려주는 함수
39  void cleanTable(){
40      Table cleanTables[maxTable]; // 퇴점한 테이블 관리하기 위한 객체 배열
41      int cleanCount=0; // 퇴점 테이블 개수 세는 변수
42
43      // 퇴점한 테이블 찾기
44      for(int i=0;i<tableCount;i++){
45          if(!tables[i].occupy)
46              cleanTables[cleanCount++]=tables[i];
47      }
48
49      if(cleanCount==0){
50          cout<<"청소할 테이블이 없습니다."<<endl;
51          return;
52      }

```

## 4. 계획 대비 변경 사항

### 1) 기능 1

- 이전 : 테이블 선호도
- 이후 : 테이블 클래스, 입/퇴점 함수
- 사유 : 테이블 클래스로 원래 계획했던 기능 1과 2를 한번에 저장함

### 2) 기능 2

- 이전 : 입점, 퇴점 테이블
- 이후 : 입력받고 알려주기
- 사유 : 현재 main함수에서 입점/퇴점 입력하고 우선순위 알려주는 함수를 호출하는 것으로 되어있지만 터미널에서 알고 싶은 것을 입력받고 알려주는 것으로 바꾸고 싶음

## 5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무	11/3	11/17	12/1	12/15
제안서 작성	완료			
기능1		완료		
기능2			----->	
기능3				----->