

C++ 프로그래밍 및 실습

레스토랑 청소 우선 순위 프로그램

진척 보고서 #2

제출일자: 2024-12-1

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, 손님 선호도를 저장하고 점유 여부는 false로 고정하는 객체
- 적용된 배운 내용 : 클래스
- 코드 스크린샷

```
class Table{    // 테이블 클래스
public:
    int number; // 테이블 번호
    int preference; // 손님 선호도(높을수록 우선순위 높음)
    bool occupy;    // 테이블 점유 여부

    Table(){    // 기본 생성자
        number=0;
        preference=0;
        occupy=false;
    }
    Table(int n,int p){
        number=n;
        preference=p;
        occupy=false;
    }
};
```

(2) Table 초기화 함수

- 설명 : 함수를 호출하면 해당 객체배열에 테이블 번호와 선호도를 입력해줌
- 적용된 배운 내용 : 객체 배열, 함수, 조건문

- 코드 스크린샷

```
// 테이블 초기화
void initialTables(Table tables[]){
    for(int i=0;i<MAXTABLE;i++){
        tables[i]=Table(i+1,10-i); // 테이블 번호 1~9, 선호도 10에서 감소
    }
}
```

(3) Table 입점 함수

- 입출력 : 배열과 번호를 입력받고 조건문 확인 후 멘트 출력
- 설명 : 테이블 번호 점검 후 해당 번호의 테이블의 점유여부를 true로 변경
- 적용된 배운 내용 : 객체 배열, 함수, 조건문

- 코드 스크린샷

```
// 테이블 입점
void entryTable(Table tables[],int tableNumber){
    if(tableNumber<1||tableNumber>MAXTABLE){
        cout<<"잘못된 테이블 번호입니다."<<endl;
        return;
    }

    if(!tables[tableNumber-1].occupy){
        tables[tableNumber-1].occupy=true;
        cout<<tableNumber<<"번 테이블 입점했습니다."<<endl;
    }
    else
        cout<<"이미 입점한 테이블입니다."<<endl;
}
```

(3) Table 퇴점 함수

- 입출력 : 배열과 번호를 입력받고 조건문 확인 후 멘트 출력
- 설명 : 테이블 번호 점검 후 해당 번호의 테이블의 점유여부를 false로 변경
- 적용된 배운 내용 : 객체 배열, 함수, 조건문

- 코드 스크린샷

```

// 테이블 퇴점
void leaveTable(Table tables[],int tableNumber){
    if(tableNumber<1||tableNumber>MAXTABLE){
        cout<<"잘못된 테이블 번호입니다."<<endl;
        return;
    }

    if(tables[tableNumber-1].occupy){
        tables[tableNumber-1].occupy=false;
        cout<<tableNumber<<"번 테이블 퇴점했습니다."<<endl;
    }
    else
        cout<<"이미 퇴점한 테이블입니다."<<endl;
}

```

(4) main 함수

- 설명 : 작동을 원하는 번호를 입력받고 해당 테이블 번호 입력받고 해당 함수 호출, 멘트 출력
- 적용된 배운 내용 : switch문, 객체 배열, 함수
- 코드 스크린샷

```

int main(){
    Table tables[MAXTABLE]; // 테이블 입력할 객체 배열
    initialTables(tables); // 테이블 초기화

    while(true){
        int choice;
        cout<<endl;
        cout<<"1. 입점 테이블 입력"<<endl;
        cout<<"2. 퇴점 테이블 입력"<<endl;
        cout<<"3. 테이블 청소 우선순위"<<endl;
        cout<<"4. 프로그램 종료"<<endl;
        cout<<"원하는 번호를 입력하세요:";
        cin>>choice;

        int tableNumber; // 입력한 테이블 번호 저장할 변수
        switch(choice){
            case 1:
                cout<<"입점 테이블 번호를 입력하세요:";
                cin>>tableNumber;
                entryTable(tables,tableNumber); // 입점 처리 함수
                break;

```

```

        case 2:
            cout<<"퇴점 테이블 번호를 입력하세요:";
            cin>>tableNumber;
            leaveTable(tables,tableNumber);//퇴점 처리 함수
            break;
        case 3:
            cleanTable(tables);//청소우선순위 함수
            break;
        case 4:
            cout<<"프로그램을 종료합니다"<<endl;
            return 0;
        default:
            cout<<"잘못된 입력입니다."<<endl;
            break;
    }
}
return 0;
}

```

2) 테스트 결과

(1) 입점/퇴점 테이블 입력

- 설명 : 입/퇴점 선택하고 해당 테이블 번호 입력하면 점유 여부 변경, 번호 점검 잘 되는지 확인

- 테스트 결과 스크린샷

<p>1. 입점 테이블 입력 2. 퇴점 테이블 입력 3. 테이블 청소 우선순위 4. 프로그램 종료 원하는 번호를 입력하세요:1 입점 테이블 번호를 입력하세요:2 2번 테이블 입점했습니다.</p>	<p>1. 입점 테이블 입력 2. 퇴점 테이블 입력 3. 테이블 청소 우선순위 4. 프로그램 종료 원하는 번호를 입력하세요:2 퇴점 테이블 번호를 입력하세요:2 2번 테이블 퇴점했습니다.</p>
<p>1. 입점 테이블 입력 2. 퇴점 테이블 입력 3. 테이블 청소 우선순위 4. 프로그램 종료 원하는 번호를 입력하세요:1 입점 테이블 번호를 입력하세요:2 이미 입점한 테이블입니다.</p>	<p>1. 입점 테이블 입력 2. 퇴점 테이블 입력 3. 테이블 청소 우선순위 4. 프로그램 종료 원하는 번호를 입력하세요:2 퇴점 테이블 번호를 입력하세요:6 이미 퇴점한 테이블입니다.</p>

(2) 프로그램 종료, 잘못된 입력

- 설명 : 4번 선택시 프로그램 종료, 잘못된 번호 입력시 멘트 출력
- 테스트 결과 스크린샷

```
1. 입점 테이블 입력
2. 퇴점 테이블 입력
3. 테이블 청소 우선순위
4. 프로그램 종료
원하는 번호를 입력하세요:5
잘못된 입력입니다.

1. 입점 테이블 입력
2. 퇴점 테이블 입력
3. 테이블 청소 우선순위
4. 프로그램 종료
원하는 번호를 입력하세요:4
프로그램을 종료합니다
PS C:\CPP2409-P> █
```

4. 계획 대비 변경 사항

1) 기능 1

- 이전 : 번호, 선호도, 점유 여부를 받는 테이블 생성자를 만들었다. 객체 배열에 함수를 통해서 입/퇴점시 마다 테이블을 추가해야 했다. 테이블 추가 함수에서는 모든 멤버변수를 입력해야했다.
- 이후 : 테이블 생성자에서 점유 여부는 false로 고정해두고 초기화 함수를 만들어 테이블 번호와 선호도를 미리 정해두었다. 그리고 입/퇴점 함수에 테이블 번호를 점검하는 부분도 추가했다.
- 사유 : mian에서 함수 입력으로 계속 테이블을 추가하며 작동하는 것이 아니라 터미널에서 원하는 번호를 입력받고 간단하게 테이블 번호로만 작동하게 만들고 싶었다.

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			완료	
기능3				----->