

C++ 프로젝트 및 실습

# 메뉴 추천 어플

진척 보고서 #1

제출일자: 2024-11-17

제출자명: 임연우

제출자학번: 234106

## 1. 프로젝트 목표 (16 pt)

### 1) 배경 및 필요성 (14 pt)

(본문: 12pt)

매 끼니마다 무엇을 먹어야 할지, 어디서 먹어야 할지 라는 고민을 수차례 반복함. 이 문제는 혼자일 때나, 여러 명일 때나, 쉽게 해결되지 않는 문제임. 이 문제를 해결하기 위해서 음식과 식당을 추천해주는 프로그램이 필요함.

### 2) 프로젝트 목표

무엇을 먹고싶은지 사용자의 니즈를 파악해 음식 메뉴를 추천하고, 사용자에게 적합한 식당을 추천하는 프로그램을 만드는 것

### 3) 차별점

기존 프로그램들은 식당과 음식들을 그저 나열하는데 그침. 여기서 나아가 사용자 선호도 우선의 순위 목록을 나열함으로써, 메뉴를 선택함에 도움을 주는 차별점을 가짐

## 2. 기능 계획

### 1) 전남대 근처 식당들의 행렬 지도 표시

- 설명 : 간단하게 표시된 형렬의 맵을 구현해 간략한 위치 확인

(1) 세부 기능 1 : 맵의 표시된 위치 확인

- 설명 : 맵의 좌표를 검색하면 무엇이 있는지 출력

(2) 세부 기능 2 : 내 위치 표시

- 설명 : 내 위치를 입력 후 배열 지도에 표시

## **2) 음식점 정보 검색**

- 설명 : 음식점의 목록과 특정 음식점의 정보 열람

## **3) 먹고 싶은 음식 추천**

- 설명 : 먹고 싶은 음식의 특징을 입력하면 그에 해당하는 음식을 추천해줌

## **4) 먹고 싶은 음식의 식당 추천**

- 설명 : 먹고 싶은 음식을 입력하면 그 음식을 파는 가게를 알려줌

## **5) 음식점을 가격, 거리, 평점 등의 기준으로 나열**

- 설명 : 가격, 거리, 평점, 등의 기준을 설정해서 사용자가 원하는 식당을  
고를 수 있도록 도움을 줌

### 3. 진척사항

#### 1) 기능 구현

// 함수의 설계와 설명은 따로 추가

##### (1) 구현한 기능 이름 : 음식점 정보 검색

```
int main(){

    int num = 1;
    int menu;
    int restauNum;
    int foodNum;
    bool foodFound = false;
    string desiredFeature;

    while(num){
        cout << "-----" << endl;
        cout << "[기능 메뉴]" << endl
            << "1. 식당 정보" << endl
            << "2. 음식 추천받기" << endl
            << "3. 식당 추천받기" << endl
            << "4. 종료" << endl
            << "실행할 기능을 선택하세요: ";

        cin >> menu;
        cout << endl;
    }
```

- 입력: num : while문의 반복을 조절

menu : 이 다음의 스위치문의 분기 설정

- 결과: 기능 메뉴 인터페이스 출력

- 설명: while문으로 반복하면서 프로그램 실행

- 적용된 배운 내용: 반복문

## case 1 실행 - 특정 음식점 정보 검색

```
switch (menu) {  
    //식당 정보  
    case 1:  
        DisplayRestaurantList();  
  
        cout << "어떤 식당의 정보를 원하십니까?" << endl  
            << "번호를 입력하십시오: ";  
  
        cin >> restauNum;  
        cout << endl;  
  
        DisplayRestaurant(restauNum);  
  
        cout << "음식 정보를 원하십니까?" << endl  
            << "1: 1번 음식" << endl  
            << "2: 2번 음식" << endl  
            << "3: 종료" << endl  
            << "원하는 서비스를 선택하세요: ";  
  
        cin >> foodNum;  
        cout << endl;  
  
        DisplayFoodInfo(restauNum, foodNum);  
  
        break;  
}
```

- 입력: menu : 스위치문 케이스 설정

restauNum : 원하는 식당에 해당하는 번호

foodNum : 원하는 음식에 해당하는 번호

- 결과: 식당 목록 출력

원하는 식당이 있는지 질문 출력

식당을 선택한 뒤 그 식당 정보 출력

음식의 정보를 원하는지, 어떤 음식을 원하는지 질문 출력

원하는 음식의 정보 출력

설명: void DisplayRestaurantList() 함수를 호출해 식당의 목록을 출력

void DisplayRestaurant(int restauNum) 함수를 호출해 식당의 정보를 출력

void DisplayFoodInfo(int restauNum, int foodNum) 함수를 호출해 음식의 정보를  
출력

- 적용된 배운 내용: 스위치, 함수

{ } void DisplayRestaurantList() 함수 구현

```
//식당의 목록을 출력하는 함수
void DisplayRestaurantList() {
    int num = 1;

    cout << "식당 목록:" << endl;
    for (const auto& restaurant : restaurantList) {
        cout << num << ". " << restaurant.GetRestauName() << endl;
        num += 1;
    }
    cout << endl;
}
```

- 입력: num : 나열되는 식당들에 번호를 부여하여 같이 출력됨

restaurantList : 모든 식당이 저장된 벡터<Restaurant> 배열

restaurant : 식당의 이름이 저장되어 출력됨

- 결과: 반복문으로 리스트의 모든 식당들의 이름이 번호가 매겨져 함께 출력

- 설명: vector<Restaurant> 타입의 리스트의 데이터가 restaurant에 저장

반복할 때마다 +1 되는 num 출력

restaurant가 Restaurant 클래스이므로, 해당 클래스 안에 정의된

string GetRestaurantName() 함수를 호출하여 이름 출력

- 적용된 배운 내용: 반복문, 배열, 함수, 포인터

{{ }} vector<Restaurant> restaurantList 벡터 배열

```
//식당 목록 벡터
vector<Restaurant> restaurantList = {woulmidang, miss420, speedbanjum,
joseon, shinsacheon};
```

- 적용된 배운 내용: 배열, 벡터

## (2) 구현한 기능 이름 : 먹고싶은 음식 추천

case 2 실행 - 먹고 싶은 음식 추천받기

```
//음식 추천받기
case 2:
    DisplayFeatures();
    cout << "원하는 음식의 특징을 입력하십시오: " ;
    cin >> desiredFeature;
    cout << endl;

    foodFound = false;
    for (const auto& restaurant : restaurantList) {
        foodFound = restaurant.RecommendFood(desiredFeature) ;
    }

    if (!foodFound) {
        cout << "특징에 맞는 음식이 없습니다." << endl;
    }

    break;
```

- 입력: desiredFeature : string 타입의 특징을 입력하여 저장  
foodFound : 특징에 맞는 음식이 있는지 확인하는 변수
- 결과: 특징 목록을 출력  
특징 목록 중 원하는 특징을 입력, 저장  
음식점을 돌며 원하는 음식이 있으면 이름을 출력  
원하는 음식이 없으면 "맞는 음식이 없음" 출력
- 설명: void DisplayFeatures() 함수를 호출해 모든 특징 목록을 출력  
원하는 특징을 입력받아 desiredFeature에 저장  
restaurantList에서 식당을 추출해 restaurant에 저장  
restaurant는 Restaurant클래스이므로 해당 클래스에 정의된  
bool RecommendFood(const string& desiredFeature) 함수를 호출  
함수 호출의 결과로 이름을 출력하고 함수 반환값을 foodFound에 저장  
원하는 음식이 없으면(= foodFound가 false이면) "맞는 음식이 없음" 출력
- 적용된 배운 내용: 스위치, 반복문, 조건문, 함수, 포인터

{ } void DisplayFeatures()함수

```
//특징 목록을 출력하는 함수
void DisplayFeatures() {
    cout << "모든 특징 목록:" << endl;
    for (const auto& feature : featureList) {
        cout << feature << " ";
    }
    cout << endl;
}
```

- 입력: featureList : 모든 특징 목록

feature의 데이터를 하나씩 저장

- 결과: 특징 목록에 들어있는 모든 특징을 출력

- 설명: vector<string> 타입의 특징 목록을 for문으로 하나씩 추출하여 출력

- 적용된 배운 내용: 반복문, 배열, 함수, 포인터

{{ }} featureList 특징 목록 벡터 배열

```
//특징 목록 벡터
vector<string> featureList = {"korean", "chinese", "japanese",
                             "vietnamese", "italian", "spicy", "sour"};
```

- 적용된 배운 내용: 배열, 벡터



{ } bool RecommendFood(const string& desiredFeature) 함수

```
//음식이 원하는 음식인지 확인하고, 맞으면 이름을 출력하는 함수
bool RecommendFood(const string& desiredFeature) const {
    bool foodFound = false;

    if (food1.HasFeature(desiredFeature)) {
        cout << food1.GetFoodName() << endl;
        foodFound = true;
    }

    if (food2.HasFeature(desiredFeature)) {
        cout << food2.GetFoodName() << endl;
        foodFound = true;
    }

    return foodFound;
}
```

- 입력: desiredFeature : 입력받은 string 타입의 원하는 특징

foodFound : 원하는 음식이 있는지 없는지 확인하는 값

-결과: 원하는 특징의 음식이 있으면 그 음식 이름을 출력

함수 종료 시 foodFound 값 true 반환

원하는 특징의 음식이 없으면 종료 시 false 반환

- 설명: 현재 확인 중인 식당에 파는 음식 food1, food2이 Food 클래스이므로

해당 클래스에 정의된 bool HasFeature(const string& desiredFeature) 함수를

호출

함수 호출의 반환값이 true이면 Food 클래스에 정의된 string GetFoodName() 함수를

호출하여 이름 출력

food1, food2 모두 진행하고 if문이 실행됐으면 true 반환

- 적용된 배운 내용: 조건문, 함수, 포인터

{{ }} bool HasFeature(const string& desiredFeature) 함수

```
//원하는 특징을 음식이 가지고 있는지 확인하는 함수
bool HasFeature(const string& desiredFeature) const {
    for (const auto& foodFeature : features) {
        if (desiredFeature == foodFeature) {
            return true;
        }
    }
    return false;
}
```

- 입력: desiredFeature : 입력받은 string 타입의 원하는 특징  
features : Food 클래스에 내재된 vector<string> 타입 벡터 배열: 특징을 저장  
foodFeature : features에서 하나씩 저장받음
- 결과: desiredFeature과 foodFeature가 같으면 true 반환, 다르면 false 반환
- 설명: 반복문으로 돌면서 음식이 가지고 있는 모든 feature를 desiredFeature과 비교  
비교 값에 따라 bool 값 반환
- 적용된 배운 내용: 반복문, 조건문, 배열, 함수, 포인터

```
//음식 이름을 반환하는 함수
string GetFoodName() const {
    return name ;
}
```

{{ }} string GetFoodName()

- 음식 이름을 반환
- 적용된 배운 내용: 함수

## 2) 테스트 결과

### (1) 음식점 정보 검색

- 설명 : 프로그램 시작 시, 기능 메뉴 출력

- 테스트 결과 스크린샷

```
PS C:\CPP2409-P> & 'c:\Users\user2\.vscode\extensions\ms-vscode.cpptools-1.22
.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-M
IEngine-In-4fwo51v1.1cf' '--stdout=Microsoft-MIEngine-Out-ni4irr2p.wag' '--std
err=Microsoft-MIEngine-Error-4t5ibk2z.kcv' '--pid=Microsoft-MIEngine-Pid-5whma
efv.ge2' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
-----
[기능 메뉴]
1. 식당 정보
2. 음식 추천받기
3. 식당 추천받기
4. 종료
실행할 기능을 선택하세요: 
```

- 설명 : 원하는 기능의 번호 입력 - 1번 입력

식당 목록을 출력하며 선택 대기

- 테스트 결과 스크린샷

```
실행할 기능을 선택하세요: 1

식당 목록:
1. woulmidang
2. misssaigon
3. speedbanjum
4. joseonjjambong
5. shinsacheon

어떤 식당의 정보를 원하십니까?
번호를 입력하십시오: 
```

- 설명 : 원하는 식당 선택 - 1번

선택한 식당의 정보 출력

- 테스트 결과 스크린샷

```
번호를 입력하십시오: 1

식당 이름: woulmidang
평점: 5
대표 메뉴: 1.buncha, 2.pho
위치: 0, 0

음식 정보를 원하십니까?
1: 1번 음식
2: 2번 음식
3: 종료
원하는 서비스를 선택하세요: █
```

- 설명 : 원하는 식당 선택 - 2번

선택한 식당의 정보 출력

- 테스트 결과 스크린샷

```
어떤 식당의 정보를 원하십니까?
번호를 입력하십시오: 2

식당 이름: misssaigon
평점: 4.2
대표 메뉴: 1.friedrice, 2.pho
위치: 0, 0

음식 정보를 원하십니까?
1: 1번 음식
2: 2번 음식
3: 종료
원하는 서비스를 선택하세요: █
```

- 설명 : 원하는 서비스 선택 - 1번

1번 음식 설명 후 다시 기능 메뉴로 복귀

- 테스트 결과 스크린샷

```
원하는 서비스를 선택하세요: 1

이름: buncha
가격: 1.4
특징: Vietnamese sour noodle

-----
[기능 메뉴]
1. 식당 정보
2. 음식 추천받기
3. 식당 추천받기
4. 종료
실행할 기능을 선택하세요: █
```

- 설명 : 원하는 서비스 선택 - 2번

- 테스트 결과 스크린샷

```
원하는 서비스를 선택하세요: 2

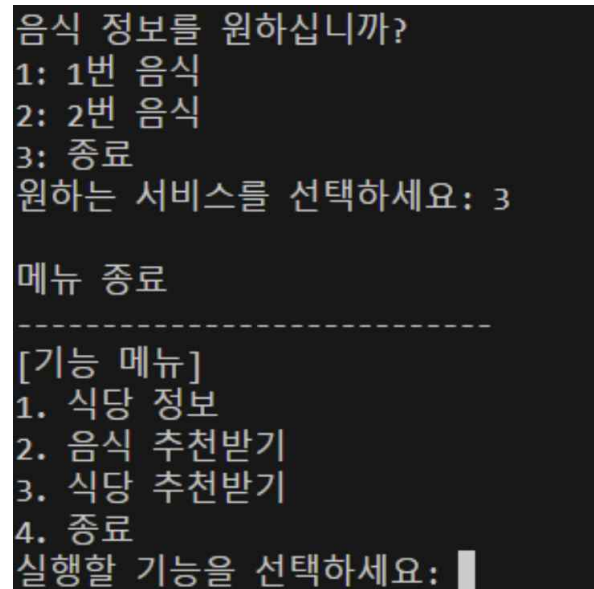
이름: pho
가격: 1.1
특징: Vietnamese beef noodle

-----
[기능 메뉴]
1. 식당 정보
2. 음식 추천받기
3. 식당 추천받기
4. 종료
실행할 기능을 선택하세요: █
```

- 설명 : 원하는 서비스 선택 - 3번

메뉴 종료 후 기능 메뉴 복귀

- 테스트 결과 스크린샷



```
음식 정보를 원하십니까?  
1: 1번 음식  
2: 2번 음식  
3: 종료  
원하는 서비스를 선택하세요: 3  
  
메뉴 종료  
-----  
[기능 메뉴]  
1. 식당 정보  
2. 음식 추천받기  
3. 식당 추천받기  
4. 종료  
실행할 기능을 선택하세요: 1
```

## (2) 먹고 싶은 음식 추천

- 설명 : 기능 2번 음식 추천 받기 선택

선택 가능한 특징 목록 출력

- 테스트 결과 스크린샷

```
[기능 메뉴]
1. 식당 정보
2. 음식 추천받기
3. 식당 추천받기
4. 종료
실행할 기능을 선택하세요: 2

모든 특징 목록:
korean chinese japanese vietnamese italian spicy sour
원하는 음식의 특징을 입력하십시오: 
```

- 설명 : 특징 입력 - korean

해당하는 음식 없으면, 다시 기능 메뉴 복귀

- 테스트 결과 스크린샷

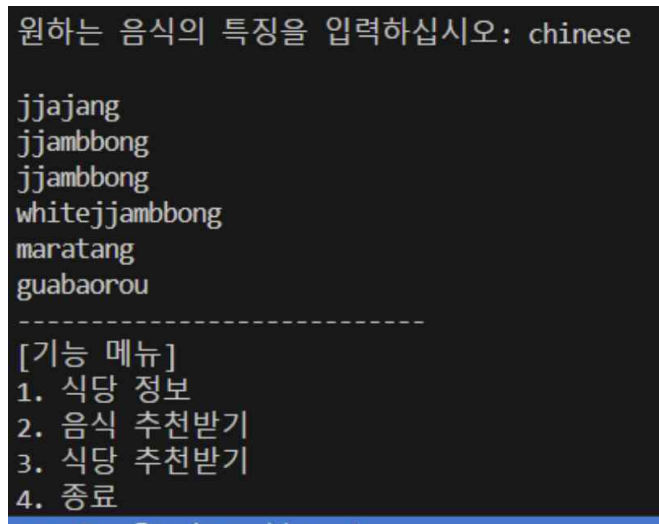
```
모든 특징 목록:
korean chinese japanese vietnamese italian spicy sour
원하는 음식의 특징을 입력하십시오: korean

특징에 맞는 음식이 없습니다.
-----
[기능 메뉴]
1. 식당 정보
2. 음식 추천받기
3. 식당 추천받기
4. 종료
실행할 기능을 선택하세요: 
```

- 설명 : 특징 입력 - chinese

선택한 특징을 가지는 음식 출력

- 테스트 결과 스크린샷



## 4. 계획 대비 변경 사항

### 1) 지도 관련 기능 합병

- 이전

기능1. 전남대 근처 음식점들을 행렬 맵으로 변환

기능2. 전남대 근처 음식점들을 행렬 맵으로 변환

기능3. 지도 표시

- 이후

기능1. 전남대 근처 식당들의 행렬 지도 표시와 세부 기능으로 분류

- 사유

: 한 기능에 포함될수 있는 작은 기능이기 때문에 이편이 간결함.



## 5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/10	11/17	.....
제안서 작성		완료			
기능2	세부기능1		완료		
기능3			완료		
기능4				진행 중	