

C++프로그래밍 및 실습

데이트 코스 추천 프로그램

진척 보고서 #3

제출일자:2024.12.15.

제출자명:손명현

제출자학번:234124

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

현대인은 일상에서 소중한 사람들과의 만남을 준비하며 장소와 활동을 찾는 데 고민과 시간이 많이 소모됨. 친구들과의 모임에서 "어디 가지?", "뭐 먹지?" 같은 질문으로 시간을 낭비하거나 만족스럽지 못한 결과를 겪는 경우가 많음. 또한, 가족 단위의 모임 시 인스타 감성에 치우친 장소나 지나치게 시끄러운 분위기 탓에 가족모임에 어울리지 않는 곳을 방문하여 불만족스러운 경험을 하게 되는 경우가 있음. 이러한 문제 해결을 위해, 다양한 모임 형태(친구, 가족, 연인 등)에 맞춤형 데이트 코스를 추천하는 프로그램이 필요함. 이 프로그램은 사용자가 시간을 절약하고 모임의 목적과 분위기에 맞는 최적의 코스를 추천받아 만족스러운 경험을 할 수 있도록 돕는 것을 목표로 함.

2) 프로젝트 목표

본 프로젝트의 목표는 사용자가 모임의 종류와 목적에 맞는 맞춤형 데이트 코스를 제공하는 것임. 사용자가 모임의 종류(친구, 가족, 연인 등)와 원하는 지역을 선택하면, 해당 모임에 어울리는 장소와 코스를 제안하는 프로그램을 개발할 예정임. 이 프로그램은 모임 특성과 장소의 분위기, 접근성 등을 종합적으로 고려하여 다양한 데이트 코스를 제공함으로써 사용자의 만족도를 높이하고자 함.

3) 차별점

현재 유사한 어플리케이션은 주로 20~30대를 타겟으로 하여 인스타 감성을 지향하고 있어, 가족모임처럼 다른 분위기를 선호하는 사용자들에게는 불편함이 있음. 본 프로그램은 이러한 문제를 해결하기 위해 다음과 같은 차별점을 가짐.

모임 특성 기반 추천: 친구, 가족, 연인 등 모임의 목적과 특성에 맞춰 장소와 코스를 추천함. 예를 들어, 가족 모임에는 조용하고 여유로운 분위기의 장소를, 친구 모임에는 활동적인 장소를 제안하여 각 모임 성격에 맞는 데이트 코스를 제공할 예정임.

장소 분위기와 걷기 난이도 고려: 모임 특성에 따라 장소의 분위기(조용함, 활기참 등)와 걷기 편의성 등 다양한 요소를 고려하여 코스를 구성함. 예를 들어, 연인 코스에는 산책하기 좋은 코스를 추천하고, 가족 단위의 코스에는 장시간 걷지 않아도 다양한 즐길 거리가 있는 장소를 제안함.

사용자 경험 중심의 맞춤형 코스 제공: 단순히 유명한 장소를 나열하는 것이 아닌, 사용자의 취향과 모임 특성에 따라 코스를 구성해 더욱 풍부한 경험을 제공한다.

2. 기능 계획

1) 기능 1: 맞춤형 데이트 코스 추천

설명: 사용자가 모임의 종류(연인, 친구, 가족 등)와 위치(도시 등)를 선택하면, 그에 맞춰 최적화된 데이트 코스를 추천하는 기능.

세부 기능 1: 모임 유형에 따른 필터링

설명: 사용자가 선택한 모임 유형에 따라 장소를 필터링. 예를 들어, 가족 모임을 위한 코스는 조용하고 편안한 장소를, 친구 모임은 활기찬 장소를 추천.

구체적인 기준:

가족 모임: 소음이 적고, 걷기 편리한 장소 추천.

친구 모임: 활기찬 분위기와 다양한 활동이 가능한 장소 추천.

연인 모임: 로맨틱하고 조용한 분위기의 장소 추천.

세부 기능 2: 위치에 따른 필터링

설명: 사용자가 선택한 도시에 맞는 장소를 추천. 예를 들어, 서울에서는 고급 레스토랑이나 명소를, 부산에서는 해변이나 야경이 좋은 장소를 추천.

구체적인 기준:

서울: 도심의 명소나 쇼핑몰, 고급 레스토랑 추천.

부산: 해변 근처의 활동적인 장소나 관광지 추천.

2) 기능 2: 랜덤 추천 및 재추천

설명: 사용자가 마음에 들지 않는 코스를 선택했을 때, 랜덤으로 다른 코스를 추천하는 기능.

세부 기능 1: 추천된 코스를 마음에 들지 않으면 재추천

설명: 사용자에게 추천된 코스가 마음에 들지 않으면, 재추천 기능을 통해 다른 코스를 추천.

구체적인 기준:

추천된 코스를 다시 보여주고, 다른 추천을 받거나 동일한 테마로 새로운 코스를 제시.

3) 기능 3: 코스 세부 정보 제공

설명: 각 추천된 코스에 대해 장소, 활동, 소요 시간 세부 정보를 제공.

세부 기능 1: 장소 및 활동 정보 제공

설명: 각 코스의 장소와 활동에 대한 정보를 상세하게 제공하여, 사용자가 원하는 코스를 쉽게 선택할 수 있도록 함.

3. 진척사항

1) 기능 구현

(1) 선택지 입력

- 입출력 : 사용자가 모임의 종류(가족, 애인, 친구) 중 하나를 선택하도록 입력을 받고, 선택에 따라 맞춤형 메시지를 출력하는 기능.

- 설명 : 프로그램은 콘솔을 통해 사용자와 상호작용하며, 사용자가 입력한 숫자(1, 2, 3)에 따라 적절한 모임 종류를 인식하여 메시지를 제공함. 올바르지 않은 입력에는 안내 메시지를 출력함.

- 적용된 배운 내용 (예: 반복문, 조건문, 클래스, 함수, 포인터 등) :

입출력: cin, cout를 사용해 사용자와 상호작용.

조건문: switch문으로 사용자의 입력값에 따라 다른 동작 수행.

기초 데이터 타입: int를 사용해 숫자 입력 처리.

- 코드 스크린샷

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int choice;
6
7     cout << "모임 종류를 선택하세요:\n";
8     cout << "1. 가족\n";
9     cout << "2. 애인\n";
10    cout << "3. 친구\n";
11    cout << "선택 (숫자 입력): ";
12    cin >> choice;
13
14    switch (choice) {
15        case 1:
16            cout << "가족 모임을 위한 추천 코스를 준비 중입니다.\n";
17            // 가족 모임에 적합한 코스 추천 기능 추가 가능
18            break;
19        case 2:
20            cout << "애인을 위한 추천 코스를 준비 중입니다.\n";
21            // 애인 모임에 적합한 코스 추천 기능 추가 가능
22            break;
23        case 3:
24            cout << "친구들과 함께 할 추천 코스를 준비 중입니다.\n";
25            // 친구 모임에 적합한 코스 추천 기능 추가 가능
26            break;
27        default:
28            cout << "올바른 숫자를 입력해 주세요.\n";
29            break;
30    }
31
32    return 0;
33 }
```

(2) 데이트 코스 추천 프로그램

설명: 사용자가 도시와 테마를 선택하면 그에 맞는 데이트 코스를 추천해주는 프로그램.

입출력:

입력: 도시 번호, 테마 번호

출력: 선택된 도시와 테마에 맞는 데이트 코스의 장소, 활동, 소요 시간

적용된 배운 내용:

조건문 (if-else) 사용하여 도시와 테마에 맞는 코스 선택

벡터(vector)와 구조체(struct)를 사용하여 여러 장소와 활동 정보를 관리

반복문(for)을 사용하여 추천 코스를 출력

함수 사용으로 코드 분리 및 관리

```
int cityChoice, themeChoice;

// 도시 선택
showCityOptions();
cout << "도시를 선택하세요 (1: 서울, 2: 부산, 3: 광주, 4: 순천, 5: 대전): ";
cin >> cityChoice;

// 테마 선택
showThemeOptions();
cout << "테마를 선택하세요 (1: 힐링, 2: 활동, 3: 문화, 4: 맛집): ";
cin >> themeChoice;

// 선택에 맞는 코스 출력
showCoursePlan(cityChoice, themeChoice);

return 0;

도시 선택지 출력
id showCityOptions() {
    cout << "도시 목록: 1: 서울, 2: 부산, 3: 광주, 4: 순천, 5: 대전" << endl;

    테마 선택지 출력
    id showThemeOptions() {
        cout << "테마 목록: 1: 힐링, 2: 활동, 3: 문화, 4: 맛집" << endl;

        도시와 테마에 맞는 데이트 코스를 출력하는 함수
        id showCoursePlan(int cityChoice, int themeChoice) {
            vector<DateCourse> courses;

            // 서울 도시와 테마에 따른 코스
            if (cityChoice == 1) {
                if (themeChoice == 1) {
                    courses.push_back({"서울", "힐링", {"서울숲", "성수동 카페거리", "창계천"}, {"산책", "카페 탐방", "산책"}, {60, 90, 60}});
                } else if (themeChoice == 2) {
                    courses.push_back({"서울", "활동", {"잠실 롯데월드", "석촌호수", "마구장"}, {"놀이기구", "호수 산책", "마구 응원"}, {120, 30, 90}});
                } else if (themeChoice == 3) {
                    courses.push_back({"서울", "문화", {"국립중앙박물관", "이태원 북카페", "한남동 갤러리"}, {"박물관 관람", "책 읽기", "갤러리 투어"},
0, 60, 90});
                } else if (themeChoice == 4) {
                    courses.push_back({"서울", "맛집", {"마포 돼지갈비", "홍대 디저트 카페", "경의선 술집"}, {"돼지갈비", "디저트", "산책"}, {60, 45,
}});
            }
        }
        // 부산 도시와 테마에 따른 코스
        else if (cityChoice == 2) {
            if (themeChoice == 1) {
                courses.push_back({"부산", "힐링", {"이기대", "오륙도 스카이라이프", "해운대 카페"}, {"산책", "스카이라이프", "카페"}, {60, 60, 90}});
            } else if (themeChoice == 2) {
                courses.push_back({"부산", "활동", {"송정 사탕", "해운대 요트", "광안리 수변공원"}, {"사탕", "요트", "산책"}, {120, 90, 60}});
            } else if (themeChoice == 3) {
                courses.push_back({"부산", "문화", {"감천문화마을", "영화의 전당", "부산현대미술관"}, {"책화 감상", "영화 관람", "미술관"}, {60,
0, 90});
            } else if (themeChoice == 4) {
                courses.push_back({"부산", "맛집", {"자갈치 시장", "국제시장", "부평항통마시갈"}, {"해산물", "간식", "마시갈"}, {60, 60, 90});
            }
        }
        // 광주 도시와 테마에 따른 코스
```

```

+     else if (cityChoice == 3) {
+         if (themeChoice == 1) {
+             courses.push_back({"광주", "힐링", {"무등산", "건널대 머타세귀이아길", "장림동 근대문화마을"}, {"산책", "문화 탐방"}, {120, 60,
+ 90}});
+         } else if (themeChoice == 2) {
+             courses.push_back({"광주", "활동", {"국립아시아문화전당", "물리광", "송광로 거리"}, {"전시 관람", "물러 타기", "쇼핑"}, {90, 120,
+ 90}});
+         } else if (themeChoice == 3) {
+             courses.push_back({"광주", "문화", {"518 민주화공원", "대인예술시장", "아트시네마"}, {"역사 탐방", "예술 작품 감상", "영화 관람"},
+ {90, 60, 120}});
+         } else if (themeChoice == 4) {
+             courses.push_back({"광주", "맛집", {"송정떡갈비", "동명동 카페거리", "국밥 맛집"}, {"떡갈비", "디저트", "국밥"}, {60, 45, 60}});
+         }
+     }
+     // 순천 도시와 테마에 따른 코스
+     else if (cityChoice == 4) {
+         if (themeChoice == 1) {
+             courses.push_back({"순천", "힐링", {"순천만 국가정원", "순천만 습지", "조계산 숲길"}, {"산책", "자연 탐방", "산책"}, {120, 90,
+ 60}});
+         } else if (themeChoice == 2) {
+             courses.push_back({"순천", "활동", {"드라마 촬영장", "채널 마술", "니이트 사파리"}, {"사진 촬영", "채널", "사파리"}, {90, 120,
+ 90}});
+         } else if (themeChoice == 3) {
+             courses.push_back({"순천", "문화", {"순천 향교", "낙안읍성", "전통 찻집"}, {"전통 문화 체험", "찻집 탐방"}, {90, 60, 45}});
+         } else if (themeChoice == 4) {
+             courses.push_back({"순천", "맛집", {"고막장식", "마시방", "간식 카페"}, {"고막장식", "마시방", "디저트"}, {60, 60, 45}});
+         }
+     }
+     // 대진 도시와 테마에 따른 코스
+     else if (cityChoice == 5) {
+         if (themeChoice == 1) {
+             courses.push_back({"대진", "힐링", {"대청호 오백리길", "한밭수목원", "유성온천"}, {"산책", "자연 탐방", "온천"}, {120, 90, 60}});
+         } else if (themeChoice == 2) {
+             courses.push_back({"대진", "활동", {"생심당", "카미스트 자연사박물관", "엑스포 다리"}, {"맛집", "박물관 체험", "산책"}, {60, 90,
+ 60}});
+         } else if (themeChoice == 3) {
+             courses.push_back({"대진", "문화", {"대진 예술의전당", "중앙시장", "갤러리"}, {"공연 관람", "시장 탐방", "미술관"}, {120, 60, 90}});
+         } else if (themeChoice == 4) {
+             courses.push_back({"대진", "맛집", {"활국수 맛집", "생심당 디저트", "범거리 포장마차"}, {"활국수", "디저트", "포장마차"}, {60, 45,
+ 90}});
+         }
+     }
+ }
+
+ // 선택된 코스가 없을 경우
+ if (courses.empty()) {
+     cout << "선택하신 도시와 테마에 맞는 코스가 없습니다." << endl;
+     return;
+ }
+
+ // 코스 출력
+ for (const auto& course : courses) {
+     displayCourse(course);
+ }
+ }
+
+ // 코스 출력 함수
+ void displayCourse(const DateCourse& course) {
+     cout << "\n***** " << course.city << " " << course.theme << " 테마 코스 ***** " << endl;
+     for (size_t i = 0; i < course.places.size(); i++) {
+         cout << "장소: " << course.places[i] << endl;
+     }
+ }

```

(3) 재추천 프로그램

입출력:

사용자가 특정 항목(영화, 음식, 여행지)에 대한 추천을 요청하도록 설계.

사용자가 추천받은 항목을 만족하지 않을 경우 "다른 추천"을 요청하면 새로운 항목을 제공.

설명:

프로그램은 사용자가 입력한 추천 카테고리(영화, 음식, 여행지)를 기반으로 첫 번째 추천을 제공.

사용자가 "다른 추천"을 선택하면, 같은 카테고리에서 추가로 항목을 추천함.

사용자가 만족하거나 프로그램을 종료하기를 원하면 프로세스를 종료함.

올바르지 않은 입력(잘못된 카테고리나 요청)을 처리하기 위해 오류 메시지를 제공.

적용된 배운 내용

입출력 : 사용자와의 상호작용을 위해 cin, cout 사용.

조건문 : if문과 else if문을 사용해 입력값을 처리하고, 상황에 따라 다른 동작 수행.

반복문 : 사용자가 만족하거나 종료 요청을 할 때까지 새로운 추천을 제공하는 while문 활용.

자료구조 : 각 카테고리의 추천 항목을 저장하기 위해 배열 사용.

```
displayCourse(recommendedCourses); // 추천 코스 출력

char retry;
do {
    cout << "다시 추천을 받으시겠습니까? (y/n): ";
    cin >> retry;
    if (retry == 'y' || retry == 'Y') {
        reRecommend(recommendedCourses, selectedCourses); // 재추천
        displayCourse(recommendedCourses); // 새 추천 리스트 출력
    }
} while (retry == 'y' || retry == 'Y'); // 재추천을 반복하는 루프

// 동적 할당된 메모리 해제
for (auto& course : courses) {
    delete course;
}
return 0; // 프로그램 종료
```

2) 테스트 결과

(1) 테스트한 기능 이름

- 설명

입력값	출력 결과	기대 결과	테스트 결과
1	가족 모임을 위한 추천 코스를 준비 중입니다.	가족 관련 메시지 출력	성공
2	애인을 위한 추천 코스를 준비 중입니다.	애인 관련 메시지 출력	성공
3	친구들과 함께 할 추천 코스를 준비 중입니다.	친구 관련 메시지 출력	성공
0	올바른 숫자를 입력해 주세요.	잘못된 입력 메시지 출력	성공
a (문자)	올바른 숫자를 입력해 주세요.	잘못된 입력 메시지 출력	성공

- 테스트 결과 스크린샷

모임 종류를 선택하세요:

1. 가족
2. 애인
3. 친구

선택 (숫자 입력): 1

가족 모임을 위한 추천 코스를 준비 중입니다.

모임 종류를 선택하세요:

1. 가족
2. 애인
3. 친구

선택 (숫자 입력): 2

애인을 위한 추천 코스를 준비 중입니다.

모임 종류를 선택하세요:

1. 가족
2. 애인
3. 친구

선택 (숫자 입력): 3

친구들과 함께 할 추천 코스를 준비 중입니다.

모임 종류를 선택하세요:

1. 가족
2. 애인
3. 친구

선택 (숫자 입력): 0

올바른 숫자를 입력해 주세요.

모임 종류를 선택하세요:

1. 가족
2. 애인
3. 친구

선택 (숫자 입력): a

올바른 숫자를 입력해 주세요.

테스트한 기능 이름	설명	입력	출력
데이트 코스 추천 프로그램	사용자가 선택한 도시와 테마에 맞는 데이트 코스를 출력.	입력: 도시 선택 - 순천 (4), 테마 선택 - 힐링 (1)	출력: 순천 힐링 데이트 코스 장소: 순천만 국가정원, 활동: 산책, 소요 시간: 120분 장소: 순천만 습지, 활동: 자연 탐방, 소요 시간: 90분 장소: 조계산 숲길, 활동: 산책, 소요 시간: 60분
데이트 코스 추천 프로그램	사용자가 선택한 도시와 테마에 맞는 데이트 코스를 출력.	입력: 도시 선택 - 서울 (1), 테마 선택 - 문화 (3)	출력: 서울 문화 데이트 코스 장소: 국립중앙박물관, 활동: 박물관 관람, 소요 시간: 90분 장소: 이태원 북카페, 활동: 책 읽기, 소요 시간: 60분 장소: 한남동 갤러리, 활동: 갤러리 투어, 소요 시간: 90분

```
도시 목록: 1: 서울, 2: 부산, 3: 광주, 4: 순천, 5: 대전
도시를 선택하세요 (1: 서울, 2: 부산, 3: 광주, 4: 순천, 5: 대전): 4
테마 목록: 1: 힐링, 2: 활동, 3: 문화, 4: 맛집
테마를 선택하세요 (1: 힐링, 2: 활동, 3: 문화, 4: 맛집): 1

***** 순천 힐링 데이트 코스 *****
장소: 순천만 국가정원
활동: 산책
소요 시간: 120분
-----
장소: 순천만 습지
활동: 자연 탐방
소요 시간: 90분
-----
장소: 조계산 숲길
활동: 산책
소요 시간: 60분
```

```
도시 목록: 1: 서울, 2: 부산, 3: 광주, 4: 순천, 5: 대전
도시를 선택하세요 (1: 서울, 2: 부산, 3: 광주, 4: 순천, 5: 대전): 1
테마 목록: 1: 힐링, 2: 활동, 3: 문화, 4: 맛집
테마를 선택하세요 (1: 힐링, 2: 활동, 3: 문화, 4: 맛집): 3

***** 서울 문화 데이트 코스 *****
장소: 국립중앙박물관
활동: 박물관 관람
소요 시간: 90분
-----
장소: 이태원 북카페
활동: 책 읽기
소요 시간: 60분
-----
장소: 한남동 갤러리
활동: 갤러리 투어
소요 시간: 90분
-----
```

재추천 기능 테스트

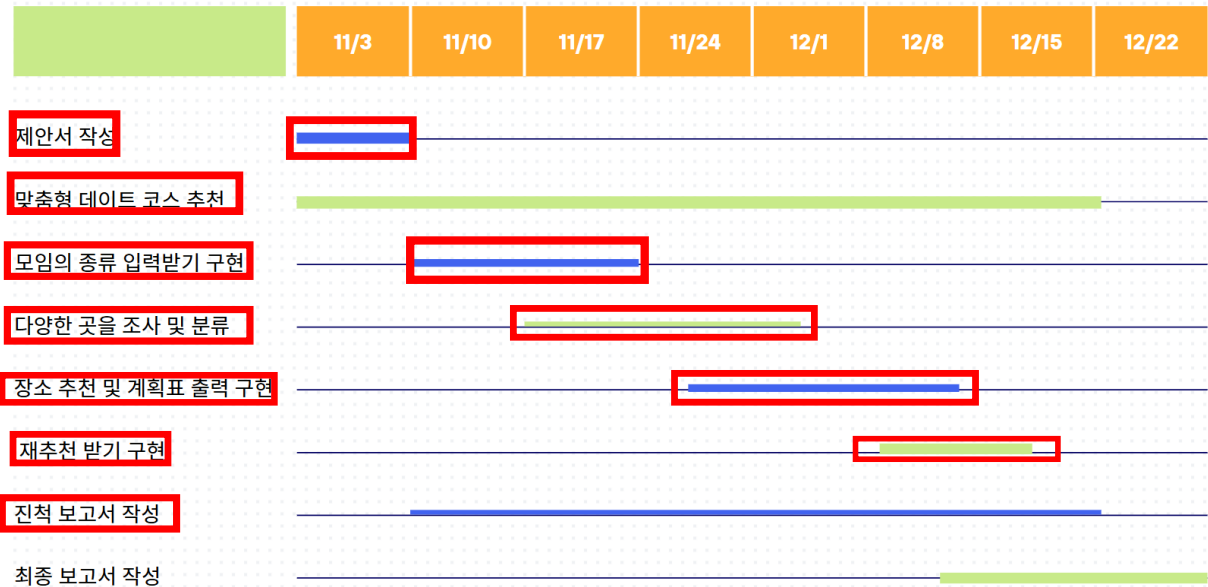
테스트 기능	설명	테스트 데이터	기대 결과	테스트 결과
1. 재추천 질문 출력 여부	사용자가 "n"을 입력했을 때, "마음에 들지 않는 코스가 있나요? (y/n):"라는 질문이 출력되는지 확인.	"n" 입력	"마음에 들지 않는 코스가 있나요? (y/n):" 질문 출력.	성공
3. 다른 추천 항목 출력	사용자가 "y"를 입력했을 때, 다른 추천 코스를 출력하는지 확인.	"y" 입력	이전 항목이 아닌 새로운 추천 코스 출력.	성공

마음에 들지 않는 코스가 있나요? (y/n): y
재추천을 원하는 코스 번호를 입력하세요 (1 ~ 4): 2

```
***** 추천 데이트 코스 *****
1. 장소: 낙안 전통 한식당
활동: 한식
소요 시간: 60분
주소: 전라남도 순천시 낙안읍성길
-----
2. 장소: 야시장
활동: 길거리 음식
소요 시간: 60분
주소: 전라남도 순천시 순천로 77
-----
3. 장소: 꼬막정식
활동: 꼬막정식
소요 시간: 60분
주소: 전라남도 순천시 순천만로 101
-----
4. 장소: 순천만 카페
활동: 커피와 디저트
소요 시간: 45분
주소: 전라남도 순천시 순천만로 99
-----
마음에 들지 않는 코스가 있나요? (y/n): y
재추천을 원하는 코스 번호를 입력하세요 (1 ~ 4): 2
선택하신 코스가 재추천되었습니다!

***** 추천 데이트 코스 *****
1. 장소: 낙안 전통 한식당
활동: 한식
소요 시간: 60분
주소: 전라남도 순천시 낙안읍성길
-----
2. 장소: 간식 카페
활동: 디저트
소요 시간: 45분
주소: 전라남도 순천시 순천만로 99
-----
3. 장소: 꼬막정식
활동: 꼬막정식
소요 시간: 60분
주소: 전라남도 순천시 순천만로 101
-----
4. 장소: 순천만 카페
활동: 커피와 디저트
```

3. 프로젝트 일정 (참고: 간트차트)



완료시 ☐ 로 표시