

C++프로그래밍 및 실습
Off-basket(오프라인 장바구니)
진척 보고서 #1

제출일자: 2023.11.26

제출자명: 김서연

제출자학번: 222400

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

온라인 쇼핑물에는 다양한 옷들이 있지만, 우리가 옷을 직접 매장에서 구매하는 이유는 인터넷으로만 보았을 때와 실제로 입어보았을 때 옷 재질과 착용감의 차이 때문이다. 따라서 나 또한 바지나 겹옷과 같은 옷은 주로 오프라인 매장에서 구매하는 편이다. 그러나 쇼핑을 하려는 마음이 생겼을 때, 어떤 옷 가게가 내 스타일과 맞는지 사전에 파악하기 어려워서 쇼핑에 많은 시간을 투자했고, 원하는 옷을 찾는 데 어려움을 겪었다.

이러한 문제점을 해결하기 위해서 생각해낸 게 "오프라인 장바구니"라는 앱이다. 이 앱을 사용하면 먼저 쇼핑을 하고자 하는 지역을 선택할 수 있다. 예를 들어 광주광역시 동구를 선택하게 되면, 해당 지역에 있는 여러 개인 옷 가게들이 리스트로 표시된다. 이러한 옷가게들을 하나씩 클릭하여 원하는 옷을 선택할 수 있다. 맘에 드는 옷은 앱의 장바구니에 넣어두고, 실제로 쇼핑을 하러 나가기로 결정했을 때 해당 옷가게를 찾아가서 직접 입어보고 구매 결정을 내릴 수 있다. 이 앱을 활용하면, 옷을 살 때 소요되는 시간을 절약할 뿐만 아니라 다양한 옷을 살펴볼 수 있으며, 자신이 원하는 옷을 직접 입어보고 구매할 수 있다. 특히 여성 구두와 같은 경우, 온라인으로 구매한 제품의 사이즈가 안 맞을 수 있기 때문에 이 앱을 활용하면 편할 거 같다.

2) 프로젝트 목표

고객들이 쇼핑하고자 하는 지역을 선택 후, 원하는 옷들을 장바구니에 담고, 옷의 수량을 체크하면서, 오프라인 매장에 대한 리뷰도 쓸 수 있게 만들 예정이다. 원래는 앱을 개발하고 싶었으나 실력이 부족해 이번 프로젝트는 광주광역시 동구 지역을 기준으로 작은 기능들을 구현해 볼 것이다

3) 차별점

기존 쇼핑물 앱들은 대부분 온라인 옷가게들만 제공하고 있어, 실제로 옷을

직접 입어보고 살 수 없다. 하지만 이 앱에는 사용자가 쇼핑하고자 하는 특정 지역의 오프라인 매장들로만 구성되어 있어, 담아 둔 옷들을 직접 착용해보고 구매할 수 있는 차별화된 기능을 제공한다.

2. 기능 계획

1) 옷 가게 위치 확인하기

① 상위지역 목록 출력하기

상위지역으로는 우리나라의 행정 구역인 1개의 특별시, 6개의 광역시, 7개의 도, 2개의 특별자치도, 1개의 특별자치시로 구성한다. 17개의 상위지역을 화면에 표시해 사용자가 선택할 수 있게 한다.

② 상위지역 입력 받기

사용자는 자신이 옷을 사고 싶은 상위지역을 17개중 하나 선택한다.

ex) 광주광역시

③ 선택된 상위지역에 대한 하위지역 목록 출력

선택한 상위지역에 대한 하위지역을 출력한다.

ex) 광주광역시 - 동구, 서구, 남구, 북구

전라남도 - 목포시, 여수시, 순천시, 나주시 등등

④ 하위지역 입력 받기

사용자는 선택한 상위지역에 대한 하위지역도 선택한다.

ex) 광주광역시 - 동구

⑤ 선택된 지역에 있는 옷가게들의 이름과 주소 출력

사용자가 상위지역과 하위지역을 선택하면 해당 지역의 옷가게들의 이름과 주소를 표시한다. 이는 사용자가 원하는 옷가게를 쉽게 찾을 수 있도록 도와준다.

2) 옷 수량 입력하고, 확인하기

옷 가게 관리자는 일상적으로 옷 수량을 업데이트해야 한다. 이렇게 하면 사용자들은 옷을 구매하기 전에 실제 수량을 확인할 수 있게 되며, 더욱 편리하게 쇼핑을 즐길 수 있다. 이 프로젝트에서는 사용자의 입장뿐만 아니라 관리자도 활용할 수 있는 기능을 만들고 싶다. 따라서 관리자는 옷 수량을 업데이트하고, 사용자는 이를 확인하는 기능을 구현해 볼 예정이다.

(단, 옷이 팔려서 수량이 줄어드는 기능은 구현하기 힘들기 때문에 아침마다 관리자가 직접 입력한 수량으로 체크할 것이다.)

① 옷 수량 입력하기 (관리자)

관리자는 아침마다 매장에 있는 옷의 이름과 수량을 입력한다.

② 옷 수량 확인하기 (사용자)

사용자는 아침에 관리자가 입력한 실제 수량을 확인한다.

3) 원하는 옷 장바구니에 담아두기

앱을 개발하는 게 아니라 어려움이 있지만 최대한 비슷하게 구현해 볼 예정이다.

① 옷 종류 출력하기

ex) 상의, 하의, 아우터, 신발

② 옷 상세정보 출력하기

사용자가 원하는 옷 종류를 선택하면 해당 종류에 대한 여러 옷에 관한 정보가 표시된다.

ex) 옷 색상, 종류, 재질, 사이즈,

③ 파일에 저장하기

사용자가 맘에 드는 옷들을 선택하면 이러한 선택사항이 파일에

저장된다. (여기서 파일은 앱의 장바구니와 같은 의미다.)

4) 오프라인 매장 리뷰 작성/확인하기

선호하는 스타일의 옷 매장을 쉽게 찾는 방법은 리뷰를 읽는 것이다.

리뷰를 통해 옷의 재질과 품질이 좋은 지 확인할 수 있어서 조금 더 쉽게 옷을 고를 수 있다.

① 리뷰 작성하기

사용자는 그 옷 매장에 대해 리뷰를 남길 수 있다.

ex) 넉넉한 핏이고 생각보다 두꺼워서 따뜻하게 잘 입고 다녔어요!!

색감은 좀 어두운 거 같아요. 그래도 예뻐서 만족합니다.

② 리뷰 확인하기

사용자들은 옷의 재질과 색상을 미리 확인하기 위해 리뷰를 참고할 수 있다. 장바구니에 담아두고 매장에 방문하기 전에 리뷰를 통해 선별하면, 쇼핑할 때 시간을 절약할 수 있다.

3. 진척사항

1) 옷 가게 위치 확인하기

(1) 상위지역 목록 출력하기

- 입출력

입력: majorRegions = 상위 지역 목록

⇒ 이 벡터에는 출력하고자 하는 상위 지역들의 이름이 문자열 형태로 저장되어 있다.

출력: 구분선, majorRegions 벡터에 있는 상위 지역 목록

- 설명

상위 지역들의 이름을 담고 있는 문자열 벡터를 받아, 그 내용을 출력한다.

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

PrintMajorRegions 함수를 정의하고 있다.

함수의 매개변수를 정의할 때 동적 배열을 가지고 있는 벡터를 사용하였다.

또한 참조자를 사용해 매개변수가 전달될 때 원본 데이터에 대한 참조(주소)만 전달되게 만들었다. 이는 데이터 복사가 없어 메모리 사용량과 처리 시간을 절약할 수 있다. majorRegions 벡터의 각 요소들을 출력하기 위해 for 반복문을 사용하였다.

이는 0부터 시작해 majorRegions 크기보다 작을 때까지 반복한다.

이렇게 정의한 함수는 상위지역의 목록을 출력하는 기능을 한다.

- 코드 스크린샷

```
// 기능1 함수 정의
// 1.1 상위 지역의 목록을 출력하는 함수
void printMajorRegions(const std::vector<std::string>& majorRegions) {
    std::cout << "======" << std::endl;
    // 반복문을 이용해 majorRegions에 있는 모든 요소들 출력하기
    for (int i = 0; i < majorRegions.size(); i++) {
        std::cout << i + 1 << ": " << majorRegions[i] << std::endl;
    }
    std::cout << "======" << std::endl;
}
```

(2) 상위지역 입력 받기

- 입출력

입력:

1. majorRegions = 상위 지역 목록

2. majorRegionIndex = 사용자 입력 값

출력: 구분선, 선택한 상위 지역의 인덱스

- 설명

사용자에게 상위 지역 인덱스를 입력 받아, 해당 인덱스를 반환한다.

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

selectMajorRegion 함수를 정의하고 있다.

함수의 매개변수를 정의할 때 동적 배열을 가지고 있는 벡터를 사용하였다.

또한 참조자를 사용해 매개변수가 전달될 때 원본 데이터에 대한 참조(주소)만 전달되게 만들었다. 이는 데이터 복사가 없어 메모리 사용량과 처리 시간을 절약할 수 있다.

사용자가 선택한 상위 지역을 출력하기 위해서 사용자로부터 원하는 지역의 인덱스를 입력하라는 메시지를 출력하고, 입력 받은 인덱스 값을 함수의 호출자에게 반환한다.

- 코드 스크린샷

```
// 1.2 상위지역 입력 받는 함수
int selectMajorRegion(const std::vector<std::string>& majorRegions) {
    std::cout << "원하는 상위 지역을 선택하세요: ";
    int majorRegionIndex;
    std::cin >> majorRegionIndex;
    std::cout << "===== " << std::endl;
    return majorRegionIndex;
}
```

(3) 선택된 상위 지역에 대한 하위 지역 목록 출력

- 입출력

입력:

1. minorRegions

-> 이 매개변수는 상위 지역의 이름을 키로 하고, 해당 상위 지역에 해당하는 하위 지역들의 이름 목록을 값으로 하는 map이다.

2. selectedMajorRegion

-> 이 매개변수는 사용자가 선택한 상위 지역의 이름이다.

출력: 구분선, 선택한 상위 지역에 해당하는 하위 지역들의 목록

- 설명

주어진 상위 지역에 해당하는 하위 지역들의 목록을 사용자에게 출력한다.

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

printMinorRegions 함수를 정의하고 있다.

함수의 매개변수를 정의할 때 동적 배열을 가지고 있는 벡터를 사용하였다.

또한 참조자를 사용해 매개변수가 전달될 때 원본 데이터에 대한 참조(주소)만 전달되게 만들었다. 이는 데이터 복사가 없어 메모리 사용량과 처리 시간을 절약할 수 있다.

반복문 for문을 이용해 각 벡터의 하위 지역에 대해 반복한다.

selectedMajorRegion을 키로 사용하여 이에 해당하는 하위 지역들의 벡터를 반환한다.

auto&를 이용해 자동 타입 추론을 사용하였고, 하위 지역의 이름을 참조한다.

- 코드 스크린샷

```
// 1.3 상위 지역에 대한 하위 지역 목록을 사용자에게 출력하는 함수
void printMinorRegions(
    const std::map<std::string, std::vector<std::string>>& minorRegions,
    const std::string& selectedMajorRegion) {
    int i = 1;
    // 반복문을 이용해 selectedMajorRegion에 해당하는 하위 지역들의 벡터를 반환
    for (const auto& region : minorRegions.at(selectedMajorRegion)) {
        std::cout << i++ << ": " << region << std::endl;
    }
    std::cout << "======" << std::endl;
}
```

(4) 하위 지역 입력 받기

- 입출력

입력:

1. minorRegions = 하위 지역 목록
2. selectedMajorRegion = 사용자가 선택한 상위 지역 목록
3. minorRegionIndex = 사용자 입력 값

출력: 구분선, 선택한 하위 지역

- 설명

사용자에게 하위 지역 인덱스를 입력 받아, 해당 인덱스를 반환한다.

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

selectMinorRegion 함수를 정의하고 있다.

함수의 매개변수를 정의할 때 동적 배열을 가지고 있는 벡터를 사용하였다.

또한 참조자를 사용해 매개변수가 전달될 때 원본 데이터에 대한 참조(주소)만 전달되게 만들었다. 이는 데이터 복사가 없어 메모리 사용량과 처리 시간을 절약할 수 있다.

사용자가 선택한 하위 지역을 출력하기 위해서 사용자로부터 원하는 지역의 인덱스를 입력하라는 메시지를 출력하고, 입력 받은 인덱스 값을 함수의 호출자에게 반환한다.

- 코드 스크린샷

```
// 1.4 하위지역 입력 받는 함수
int selectMinorRegion(
    const std::map<std::string, std::vector<std::string>>& minorRegions,
    const std::string& selectedMajorRegion) {
    std::cout << selectedMajorRegion << " 내에서 하위 지역을 선택하세요: ";
    int minorRegionIndex;
    std::cin >> minorRegionIndex;
    std::cout << "===== " << std::endl;
    return minorRegionIndex;
}
```

(5) 선택된 지역에 있는 옷가게들의 이름과 주소 출력

- 입출력

입력:

1. stores = Store 객체들의 벡터
2. majorRegion = 상위 지역 목록
3. minorRegion = 하위 지역 목록

출력: 구분선, 선택한 지역에 해당하는 옷 가게들의 이름과 주소
해당 옷 가게가 없는 경우, 없다는 메시지

- 설명

사용자가 선택한 지역에 대한 옷 가게 이름과 주소를 출력한다.

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

printStores 함수를 정의하고 있다.

함수의 매개변수를 정의할 때 동적 배열을 가지고 있는 벡터를 사용하였다.

또한 참조자를 사용해 매개변수가 전달될 때 원본 데이터에 대한 참조(주소)만 전달되게 만들었다. 이는 데이터 복사가 없어 메모리 사용량과 처리 시간을 절약할 수 있다.

stores 벡터의 모든 Store객체를 순서대로 처리하기 위해 반복문 for문을 사용했다.

auto&를 이용해 자동 타입 추론을 사용하였고, Store 객체를 참조한다.

if 반복문을 사용해 사용자가 선택한 majorRegion과 minorRegion이 일치한지 확인하고, 일치하면 옷 가게의 이름과 주소를 출력한다.

만약 일치하지 않으면 "옷가게가 없다"는 메시지를 출력한다.

- 코드 스크린샷

```
// 1.5 선택된 지역에 있는 옷가게들의 이름과 주소 출력하는 함수
void printStores(const std::vector<Store>& stores,
                 const std::string& majorRegion,
                 const std::string& minorRegion) {
    bool found =
        false; // found 변수 설정 (주어진 지역내에서 가게를 찾았는지에 대한 여부)
    std::cout << majorRegion << " " << minorRegion
        << "에 있는 옷 가게 목록:" << std::endl;
    // store벡터의 모든 Store 객체를 순서대로 한 번씩 처리하는 반복문
    for (const auto& store : stores) {
        // 찾고자 하는 지역에, 옷가게가 있을 경우 이름과 주소 출력
        if (store.majorRegion == majorRegion && store.minorRegion == minorRegion) {
            std::cout << store.name << ": " << store.address << std::endl;
            found = true;
        }
    }

    // 해당 지역에 옷가게가 없을 경우
    if (!found) {
        std::cout << majorRegion << " " << minorRegion << "에는 옷 가게가 없습니다."
            << std::endl;
    }
}
```

(6) 기타

1. 헤더 파일

```
#include <iostream>
#include <map>
#include <string>
#include <vector>
using namespace std;

// 가게 정보를 담는 구조체
struct Store {
    std::string name;
    std::string majorRegion;
    std::string minorRegion;
    std::string address;
};

// 함수 선언들
void printMajorRegions(const std::vector<std::string>& majorRegions);
int selectMajorRegion(const std::vector<std::string>& majorRegions);
void printMinorRegions(
    const std::map<std::string, std::vector<std::string>>& minorRegions,
    const std::string& selectedMajorRegion);
int selectMinorRegion(
    const std::map<std::string, std::vector<std::string>>& minorRegions,
    const std::string& selectedMajorRegion);
void printStores(const std::vector<Store>& stores,
    const std::string& majorRegion,
    const std::string& minorRegion);
```

2. 소스 파일 (위의 기능 함수들)

3. main 파일

```
#include "기능1.h"

int main() {
    // 기능1. 옷가게 위치 확인
    // 지역, 옷가게 정의하기
    std::vector<Store> stores = {
        {"엠플레이그라운드", "광주광역시", "동구", "광주광역시 동구 충장로 90"},
        {"광주에이블", "광주광역시", "동구", "광주광역시 동구 충장로안길 25"},
        {"탑텐", "광주광역시", "동구", "광주광역시 동구 충장동 충장로 87"},
        {"댄디", "광주광역시", "동구", "광주광역시 동구 금남로1가"},
        {"데이지", "광주광역시", "동구", "광주광역시 동구 불로동 107-2"},
        {"MODA", "광주광역시", "서구",
            "광주광역시 서구 쌍촌동 상무민주로76번길 11-2"},
        {"구월 의상실", "광주광역시", "북구", "광주광역시 북구 우치로100번길 12"},
        {"착한옷가게", "광주광역시", "남구", "광주광역시 남구 백운동 594-8번지"},
        // 등등...
    };

    std::vector<std::string> majorRegions = {
        "광주광역시", "서울특별시", "대전광역시", "대구광역시",
        "울산광역시", "부산광역시", "인천광역시", "세종특별자치시",
        "강원특별자치도", "제주특별자치도", "전라남도", "전라북도",
        "경기도", "충청북도", "충청남도", "경상북도",
        "경상남도"
    };

    std::map<std::string, std::vector<std::string>> minorRegions = {
        {"광주광역시", {"동구", "남구", "북구", "서구"}},
        {"전라남도", {"목포시", "여수시", "순천시", "나주시", "..."}},
        {"전라북도", {"전주시", "군산시", "..."}},
        {"서울특별시", {"강남구", "종로구", "..."}}
    };
    // 등등...
}
```

```

// 1.1 majorRegions에 담긴 상위지역들의 목록 출력
printMajorRegions(majorRegions);

// 1.2 상위지역 입력 받기
int majorRegionIndex = selectMajorRegion(majorRegions);
std::string selectedMajorRegion = majorRegions[majorRegionIndex - 1];

// 1.3 선택된 상위지역에 대한 하위지역 목록 출력
printMinorRegions(minorRegions, selectedMajorRegion);

// 1.4 하위지역 입력 받기
int minorRegionIndex = selectMinorRegion(minorRegions, selectedMajorRegion);
std::string selectedMinorRegion =
    minorRegions[selectedMajorRegion][minorRegionIndex - 1];

// 1.5 선택된 지역에 있는 옷가게들의 이름과 주소 출력
printStores(stores, selectedMajorRegion, selectedMinorRegion);

return 0;

```

2) 테스트 결과

(1) 상위지역 목록 출력하기

- 설명

상위 지역들의 이름을 담고 있는 문자열 벡터를 받아, 그 내용을 출력한다

- 테스트 결과 스크린샷

```

=====
1: 광주광역시
2: 서울특별시
3: 대전광역시
4: 대구광역시
5: 울산광역시
6: 부산광역시
7: 인천광역시
8: 세종특별자치시
9: 강원특별자치도
10: 제주특별자치도
11: 전라남도
12: 전라북도
13: 경기도
14: 충청북도
15: 충청남도
16: 경상북도
17: 경상남도
=====

```

(2) 상위지역 입력 받기

- 설명

사용자에게 상위 지역 인덱스를 입력 받아, 해당 인덱스를 반환한다.

- 테스트 결과 스크린샷

```
=====
원하는 상위 지역을 선택하세요: 1
=====
```

(3) 선택된 상위 지역에 대한 하위 지역 목록 출력

- 설명

주어진 상위 지역에 해당하는 하위 지역들의 목록을 사용자에게 출력한다.

- 테스트 결과 스크린샷

```
=====
1: 동구
2: 남구
3: 북구
4: 서구
=====
```

(4) 하위 지역 입력 받기

- 설명

사용자에게 하위 지역 인덱스를 입력 받아, 해당 인덱스를 반환한다.

- 테스트 결과 스크린샷

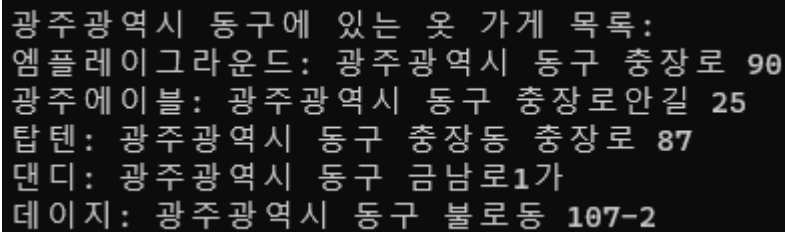
```
=====
광주광역시 내에서 하위 지역을 선택하세요: 1
=====
```

(5) 선택된 지역에 있는 옷가게들의 이름과 주소 출력

- 설명

사용자가 선택한 지역에 대한 옷 가게 이름과 주소를 출력한다

- 테스트 결과 스크린샷



```
광주광역시 동구에 있는 옷 가게 목록:  
엠플레이그라운드: 광주광역시 동구 충장로 90  
광주에이블: 광주광역시 동구 충장로안길 25  
탑텐: 광주광역시 동구 충장동 충장로 87  
댄디: 광주광역시 동구 금남로1가  
데이지: 광주광역시 동구 불로동 107-2
```

4. 계획 대비 변경 사항

- 전체적으로 기능을 조금 더 상세하게 나눴다.

1) 기능2 변경

- 이전

“사용자들이 옷 수량 확인하기” 기능

- 이후

관리자가 옷 수량을 입력하면, 사용자가 실제 수량을 알 수 있게 기능을 변경했다.

- 사유

제안서를 작성할 때 까지만 해도, 수업 시간에 동적배열에 대한 학습이 없어 정적 데이터로만 프로젝트를 개발하려고 생각했다. 그러나 최근의 수업에서 동적 배열인 벡터를 배워서 이걸 응용해 관리자가 옷 수량을 직접 등록하고 사용자가 이 정보를 확인할 수 있도록 기능을 변경하려고 한다. 이렇게 하면 학습에도 도움이 되며, 수업시간에 배운 내용을 기반으로 프로젝트를 진행할 수 있다.

2) 기능3 변경

- 이전

“원하는 옷 장바구니에 담아두기” 기능

원래는 이미지를 넣어서 그 이미지를 저장하는 방식으로 프로젝트를 만들 계획이었다.

- 이후

옷 종류를 표시하고, 사용자가 특정 종류를 택하면 해당 종류의 옷에 대한 상세 정보 (색깔, 재질, 크기, 종류) 가 화면에 나타난다. 사용자는 원하는 옷을 선택하여 그 정보를 파일에 저장하는 방식으로 기능을 수정하였다.

- 사유

이 프로젝트가 한 학기 최종 프로젝트인 만큼 수업 시간에서 배운 내용을 주로 활용하고자 한다. 이미지를 삽입하고 저장하는 부분은 수업 내용과는 무관하여, 이 프로젝트와는 어울리지 않는다고 판단하였다. 그래서 수업 시간에 학습한 내용을 기반으로 구현하기 위해 기능 3번을 변경하려고 한다.

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/ 26	12/4	12/9	12/14	12/23
제안서 작성		완료					
기능1	세부기능1	완료					
	세부기능2	완료					
	세부기능3	완료					
	세부기능4	완료					
	세부기능5	완료					
기능2	세부기능1,2		----->				
기능3	세부기능 1,2,3			----->			
기능4	세부기능1,2			----->			
최종	마무리					----->	

