C++프로그래밍 및 실습

Off-basket(오프라인 장바구니)

진척 보고서 #2

제출일자: 2023.12.17
제출자명: 김서연
제출자학번: 222400

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

온라인 쇼핑몰에는 다양한 옷들이 있지만, 우리가 옷을 직접 매장에서 구매하는 이유는 인터넷으로만 보았을 때와 실제로 입어보았을 때 옷 재질과 착용감의 차이 때문이다. 따라서 나 또한 바지나 겉옷과 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 쇼핑몰 앱들은 대부분 온라인 옷가게들만 제공하고 있어, 실제로 옷을

직접 입어보고 살 수 없다. 하지만 이 앱에는 사용자가 쇼핑하고자 하는 특정

지역의 오프라인 매장들로 만 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 재고 관리

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 사용자와 관리자 모두 활용할 수 있는 기능을 만들고 싶다. 따라서 관리자는 옷 수량을 업데이트하고, 그 업데이트 한 수량은 파일에 저장되어 관리하기 쉽게 만들 것이다. 또한 사용자는 이를 확인하는 기능을 구현해 볼 예정이다.

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

① 메뉴 선택하기

1. 재고 업데이트
2. 재고 확인
3. 수량 파일 확인
4. 종료하기

이렇게 4가지의 메뉴를 선택할 수 있는데 재고 업데이트와 수량파일 확인은 관리자가 사용하는 기능이고, 재고 확인하는 기능은 사용자가 사용하는 기능이다. 파일을 만들 때는 아침마다 재고가 바뀌어야 되기 때문에 ios::trunc를 사용해서 새로운 내용으로 매번 교체 할 예정이다.
(하지만 조금 더 고민해 볼 문제, trunc를 쓰면 프로그램을 시작할 때마다 초기화되기 때문에 정보가 저장되지 않음 => 최종까지 고민하고 해결)

② 재고 업데이트 기능 (관리자)

관리자는 아침마다 가게 이름, 매장에 있는 옷의 이름, 수량을 입력한다.

③ 재고 확인하기 (사용자)

사용자는 아침에 관리자가 입력한 실제 수량을 확인한다.

④ 수량 파일 확인하기 (관리자)

관리자가 입력한 재고를 파일에 저장해 관리한다. 이 기능은 관리자가 옷 재고를 한눈에 보기가 쉬워서 재고 관리하기 편하다.

⑤ 추가 기능(오류 처리)

메뉴를 선택할 때 숫자가 아닌 문자열을 잘 못 입력한 경우, "숫자만 입력해주세요" 라고 출력한다.

3) 원하는 옷 장바구니에 담아두기

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

① 옷 상세정보 출력하기

사용자가 선택한 옷 가게에 있는 옷 종류에 대한 정보가 표시된다.
ex) 옷 색상, 종류, 재질, 사이즈,

② 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

* 기능3까지 구현하긴 했지만 기능 3은 아직 수정해야 될 부분이 많아서 보고서에는 기능2까지만 쓰도록 하겠습니다!

1) 기능2 구현: 옷 재고 관리 (기능2.cpp, 기능.h, main.cpp)

<기능2. cpp>파일

(1) 메뉴 선택하기 (1. 재고 업데이트, 2. 재고 확인, 3. 수량 파일 확인, 4. 종료)

- 입출력

  입력: choice = 메뉴 선택

  출력: 구분선, 메뉴 리스트

- 설명

  4가지의 메뉴리스트를 출력하고, 사용자는 사용하고 싶은 기능을 선택할 수 있다.
  만약 사용자가 숫자가 아닌 문자열을 입력한 경우에는 validNumber Input 함수가 호출되어, 오류 메시지가 뜬다.

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

  If조건문을 이용해 입력된 값이 숫자인지 검증한다. 입력이 숫자가 아닐 경우에는 false를 반환하고, 숫자일 경우에는 true를 반환한다. 만약 숫자면 continue문을 통해 다음 반복으로 진행한다. 또한 입력 받은 값에 따라 다른 작업을 수행하는 switch 조건문을 사용했다.

- 코드 스크린샷

```
// 기능2 : 옷 재고 관리
void manageInventory(ClothingInventory& inventory, const std::vector<Store>& stores) {
    while (true) {
        // 2.1 메뉴 선택하기
        std::cout << "₩n2. 옷 재고 관리" << endl;
        std::cout << "========================================" << std::endl;
        std::cout << "1. 재고 업데이트" << std::endl;
        std::cout << "2. 재고 확인" << std::endl;
        std::cout << "3. 수량 파일 확인" << std::endl;
        std::cout << "4. 종료" << std::endl;
        std::cout << "========================================" << std::endl;
        std::cout << "메뉴를 선택해주세요: ";


        // 사용자에게 메뉴 선택 입력받기
        int choice;
        std::cin >> choice;
        if (!validNumberInput()) continue; // 숫자 입력 검증
        std::cin.ignore();

        switch (choice) {
```

(2) 재고 업데이트

- 입출력

  입력: storeName, itemName, quantiy (가게 이름, 상품 이름, 수량)

  출력: 구분선, 입력 안내 메시지, 가게를 찾지 못 했을 때의 메시지

- 설명

  사용자로부터 가게이름, 상품 이름, 그리고 수량을 입력 받아 재고를 업데이트한다. 사용자가 유효한 가게이름을 입력하지 않았을 경우, 오류 메시지를 표시하고, 유효한 입력인 경우에만 재고를 업데이트 할 수 있다.

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

  Bool storeFound = false 라는 논리형 자료형을 이용해 일치하는 가게를 찾았는지 여부를 추적한다. For 반복문을 사용해 사용자가 입력한 storeName과 일치하는 가게를 찾는다. If 조건문을 이용해서 만약 일치하는 가게를 찾으면 updateStock 함수를 호출하여 재고를 업데이트한다.

- 코드 스크린샷

```
// 2-2. 재고 업데이트
case 1: {
    // 사용자로부터 가게 이름, 상품 이름, 수량 입력받기
    std::string storeName, itemName;
    int quantity;
    std::cout << "가게 이름: ";
    getline(std::cin, storeName);
    std::cout << "상품 이름: ";
    getline(std::cin, itemName);
    std::cout << "수량: ";
    std::cin >> quantity;
    // 숫자 입력 검증
    if (!validNumberInput()) continue;
    std::cin.ignore();

    bool storeFound = false;  // storeFound 라는 변수 선언 : 일치하는 가게를 찾았는지 여부 추적
    for (const auto& store : stores) {
        // store 객체의 옷가게와 이름이 같으면 updateStock 함수 호출
        if (store.name == storeName) {
            inventory.updateStock(store, itemName, quantity);
            storeFound = true;
            break;
        }
    }
    // 가게이름이 다를 때
    if (!storeFound) {
        std::cout << "가게를 찾을 수 없습니다." << std::endl;
    }
    break;
}
```

(3) 재고 확인하기

- 입출력

  입력: storeName, itemName (가게 이름, 상품 이름)

  출력: 구분선, 입력 안내 메시지, 가게를 찾지 못 했을 때의 메시지

- 설명

  사용자로부터 가게 이름과 상품 이름을 입력 받아 해당 상품의 재고 상태를 확인한다. 사용자가 유효한 가게 이름을 입력하지 않았을 경우, 적절한 오류 메시지를 표시하고, 유효한 입력인 경우에만 재고 확인을 수행한다.

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

  Bool storeFound = false 라는 논리형 자료형을 이용해 일치하는 가게를 찾았는지 여부를 추적한다. For 반복문을 사용해 사용자가 입력한 storeName과 일치하는 가게를 찾는다. If 조건문을 이용해서 만약 일치하는 가게를 찾으면 checkStock 함수를 호출하여

해당 상품의 재고를 확인한다.

- 코드 스크린샷

```
// 2-3. 재고 확인하기 기능
case 2: {
    std::string storeName, itemName;
    std::cout << "가게 이름: ";
    getline(std::cin, storeName);
    std::cout << "상품 이름: ";
    getline(std::cin, itemName);

    bool storeFound = false; // storeFound 라는 변수 선언 : 일치하는 가게를 찾았는지 여부 추적
    for (const auto& store : stores) {
        // store 객체의 옷가게와 이름이 같으면 checkStock 함수 호출
        if (store.name == storeName) {
            inventory.checkStock(store, itemName);
            storeFound = true;
            break;
        }
    }
    // 가게이름이 다를 때
    if (!storeFound) {
        std::cout << "가게를 찾을 수 없습니다." << std::endl;
    }
    break;
}
```

(4) 수량 파일 확인하기

- 입출력

입력: 수량.txt (재고 정보 파일)

⇨ Ifstream을 사용하여 파일을 읽기 모드로 연다.

출력: 구분선, "옷 재고 확인" 문구 출력, 파일에 있는 각 줄의 내용 출력

- 설명

수량.txt 파일의 내용을 확인하는 기능을 수행한다. 만약 파일 열기에 성공하면 파일의 내용을 한 줄씩 읽어 출력하고, 실패하면 오류 메시지를 출력한다.

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

파일 입출력 관련 클래스인 ifstream을 사용해 file을 생성한다. If조건문을 사용해 파일을 열 수 없는 경우, 오류 메시지를 표준 에러 출력 스트림 'cerr'을 통해 출력한다. 만약 파일 열기에 성공하면 while 반복문과 'getline'을 사용해 파일의 각 줄을 'line' 변수에 저장하고, 콘솔에 출력한다. 마지막으로 file.close 를 통해 파일 스트림을 닫는다.

- 코드 스크린샷

```
// 2-4. 수량 파일 확인하기
case 3: {
    // 수량.txt 파일 열기
    ifstream file("수량.txt");
    string line;

    // 파일 열기를 실패했을 때
    if (!file) {
        cerr << "파일 오픈에 실패하였습니다." << endl;
    }
    // 파일 내용 읽기 및 출력
    else {
        cout << "=========================================" << endl;
        cout << "<옷 재고 확인>" << endl;
        // file의 내용을 한줄 씩 읽어 콘솔창에 출력
        while (getline(file, line)) {
            cout << line << endl;
        }
    }
    // 파일 닫기
    file.close();
    break;
}
```

(5) 입력 값 오류 처리

- 입출력

  입력: 사용자의 입력 값

  출력: 오류 메시지

- 설명

  사용자의 입력이 숫자인지 검증하는 함수이며, 숫자가 아닌 입력을 받았을 때 오류 메시지를 출력한다.

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

  validNumber Input 함수를 정의했다. 만약 숫자가 아닌 값을 입력하면 false를 반환하고, 입력이 숫자일 경우 true를 반환한다.

- 코드 스크린샷

```cpp
// 2-5 입력 값 오류 처리
bool validNumberInput() {
    if (std::cin.fail()) {
        std::cin.clear();
        std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '₩n');
        std::cout << "숫자만 입력해주세요." << std::endl;
        return false;
    }
    return true;
}
```

(6) 기타

- 입출력

  입력: 없음

  출력: 프로그램 종료 메시지, 잘못 선택했을 때의 오류 메시지

- 설명

  사용자가 프로그램 종료를 선택했을 때 종료하고, 예상된 범위 외의 선택을 했을 때 해당하는 경고 메시지를 표시하고 다시 선택하게 한다.

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

  Switch문 내에서 case 중 어느 것에도 해당하지 않는 모든 상황을 default에서 처리한다. 이 코드에서는 1부터 4가 아닌 다른 숫자를 입력했을 때 해당한다.

- 코드 스크린샷

```cpp
    // 프로그램 종료
    case 4: {
        std::cout << "프로그램을 종료합니다." << std::endl;
        return;
    }
    // 잘못 입력했을 때
    default: {
        std::cout << "잘못된 선택입니다. 다시 시도해 주세요." << std::endl;
        break;
    }
    }
}
```

<기능.h> 파일

=> ClothingInventory 클래스 정의하여 옷가게의 재고 관리 시스템 구현

(1) 파일에 재고 정보 저장하는 함수(saveToFile)

- 입출력

  입력: 없음

  출력: 수량.txt 파일

- 설명

  재고 정보를 파일에 저장하는 'saveToFile' 함수를 정의하고 있다.

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

  파일 출력 파일 스트림 클래스인 ofstream을 사용해서 "수량.txt" 파일을 쓰기 모드로 연다. 파일 쓰기 작업이 완료되면 close()를 호출하여 파일 스트림을 닫는다.

- 코드 스크린샷

```
// 기능 2: 옷가게의 재고 관리하는 클래스
class ClothingInventory {
private:
    map<string, int> stock;

    // 재고 정보를 파일에 저장하는 함수
    void saveToFile() {
        ofstream file("수량.txt");
        for (const auto& pair : stock) {
            file << pair.first << " " << pair.second << endl;
        }
        file.close();
    }
```

(2) 파일 읽는 함수 (loadFromFile)

- 입출력

  입력: 수량.txt 파일

  출력: 없음

- 설명

파일로부터 재고 데이터를 읽어 stock 맵에 로드하는 "loadFromFile"함수를 정의하고 있다. Ios::trunc를 사용해서 파일을 열 때 파일의 내용을 비워 관리자가 매일매일 새로운 데이터를 입력할 수 있게 했다.

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

입력 파일 스트림 클래스인 ifstream을 사용해 파일에서 데이터를 읽는다. Ios::trunc를 사용해서 새로운 내용으로 교체된다. 또한 while 반복문을 사용해 각 상품의 key와 수량을 읽어 들인다.

- 코드 스크린샷

```
// 파일을 읽는 함수
void loadFromFile() {
    // 매일매일 수량이 달라지기 때문에 파일이 열릴 때마다 내용이 비워짐 (고민)
    ifstream file("수량.txt", ios::trunc);
    string key;
    int quantity;
    while (file >> key >> quantity) {
        stock[key] = quantity;
    }
    file.close();
}
```

(3) 재고 업데이트하는 함수(updateStock)

- 입출력

입력: 가게이름, 상품 이름, 수량

출력: 재고 업데이트 정보

- 설명

재고 정보를 업데이트하고 이를 파일에 저장하여 재고 상태를 최신 상태로 유지한다. 사용자는 가게와 상품, 그리고 수량 정보를 통해 재고를 업데이트 할 수 있으며, 이러한 업데이트는 콘솔에 출력되고 파일에 저장된다.

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

재고를 업데이트하는 함수를 정의하였고, 파라미터 store는 Store 타입의 참조 변수로,

업데이트할 재고가 속한 상점을 나타낸다. Item도 string 타입의 참조 변수로, 업데이트할 재고의 상품 이름을 나타낸다. Quantity는 해당 상품의 새로운 재고 수량을 나타낸다. Const는 함수 내에서 수정되지 않음을 의미한다.

- 코드 스크린샷

```
public:
    ClothingInventory() { loadFromFile(); }
    // 재고 업데이트하는 함수
    void updateStock(const Store& store, const string& item, int quantity) {
        string key = store.name + ":" + item;
        stock[key] = quantity;
        cout << "가게: " << store.name << ", 상품: " << item
             << ", 수량: " << quantity << endl;
        saveToFile();
    }
```

(4) 재고 체크하는 함수(checkStock)

- 입출력

입력: 가게이름, 상품 이름, 수량

출력: 재고 정보, 재고가 없다는 메시지

- 설명

사용자는 상점과 상품의 이름을 통해 재고를 확인할 수 있으며, 해당 상품의 재고가 있으면 재고 수량을 콘솔에 출력하고, 없으면 없다는 메시지를 출력한다.

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

재고를 업데이트하는 함수를 정의하고 있다. 여기서 it은 특정원소를 가리키는 반복자이다. find함수를 통해 얻을 수 있다. 또한 ->인 포인터 접근 연산자를 사용해 반복자가 가리키는 객체(수량)에 접근한다.

- 코드 스크린샷

```
// 재고 체크하는 함수
void checkStock(const Store& store, const string& item) {
    string key = store.name + ":" + item;
    // stock 에서 key에 해당하는 요소 찾기
    auto it = stock.find(key);
    // 찾고자 하는 키가 존재하면 정보 출력하기
    if (it != stock.end()) {
        cout << "가게: " << store.name << ", 상품: " << item
            << ", 수량: " << it->second << endl;
    }
    else {
        cout << "재고가 없습니다."<< endl;
    }
}
;
```

<main.cpp> 파일

```
// 2. 옷가게 수량 확인하고, 업데이트하기
ClothingInventory inventory;
manageInventory(inventory, stores);
```

2) 테스트 결과

(1) 메뉴 선택하기

- 설명

4가지의 메뉴리스트를 출력하고, 사용자는 사용하고 싶은 기능을 선택할 수 있다.

- 테스트 결과 스크린샷

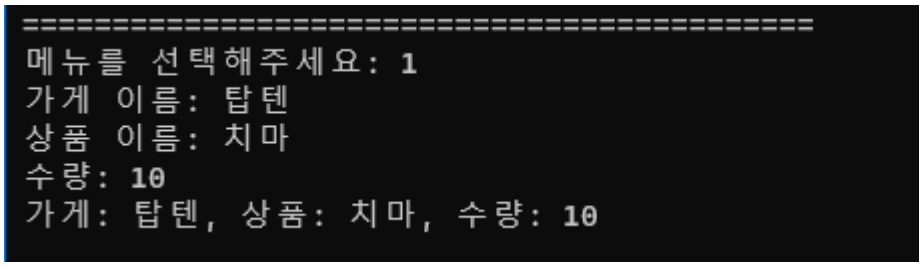
```
2. 옷 재고 관리
==========================================
1. 재고 업데이트
2. 재고 확인
3. 수량 파일 확인
4. 종료
==========================================
메뉴를 선택해주세요:
```

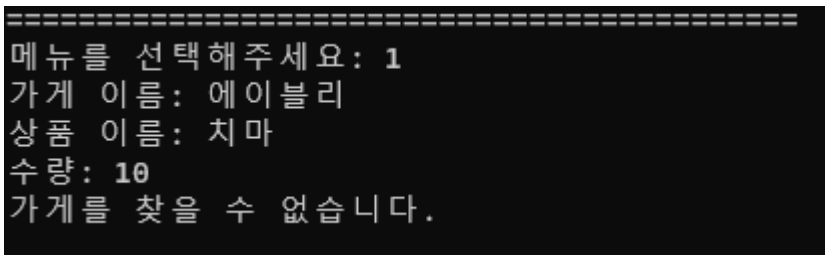
(2) 재고 업데이트

- 설명

  사용자로부터 가게이름, 상품 이름, 그리고 수량을 입력 받아 재고를 업데이트한다.
  사용자가 유효한 가게이름을 입력하지 않았을 경우, 오류 메시지를 표시하고, 유효한 입력인 경우에만 재고를 업데이트 할 수 있다.

- 테스트 결과 스크린샷



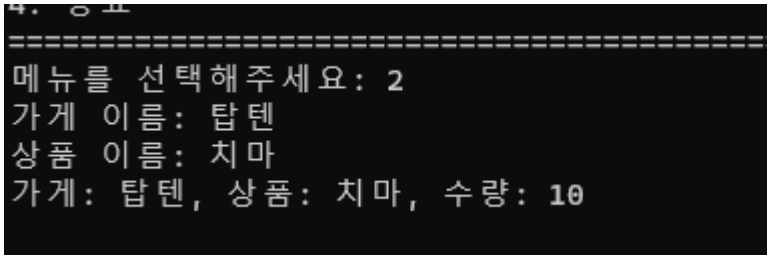


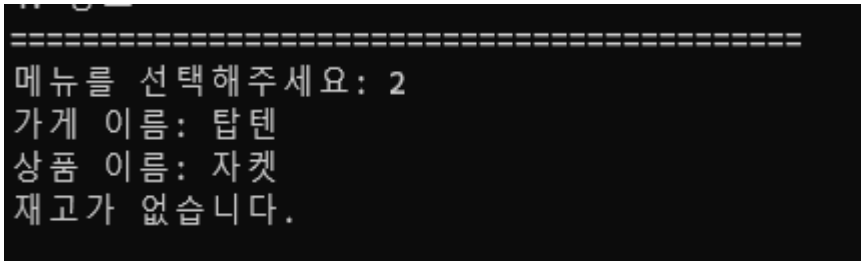
(3) 재고 확인하기

- 설명

  사용자로부터 가게 이름과 상품 이름을 입력 받아 해당 상품의 재고 상태를 확인한다.
  사용자가 유효한 가게 이름을 입력하지 않았을 경우, 적절한 오류 메시지를 표시하고, 유효한 입력인 경우에만 재고 확인을 수행한다.

- 테스트 결과 스크린샷





(4) 수량 파일 확인하기

- 설명

수량.txt 파일의 내용을 확인하는 기능을 수행한다. 만약 파일 열기에 성공하면 파일의 내용을 한 줄씩 읽어 출력하고, 실패하면 오류 메시지를 출력한다.

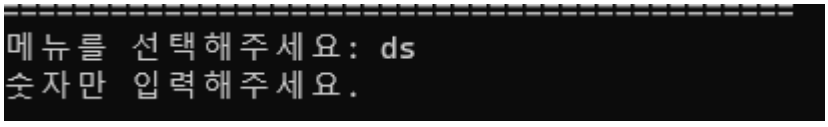
- 테스트 결과 스크린샷

(5) 입력 값 오류 처리하기

- 설명

사용자의 입력이 숫자인지 검증하는 함수이며, 숫자가 아닌 입력을 받았을 때 오류 메시지를 출력한다

- 테스트 결과 스크린샷



4. 계획 대비 변경 사항

1) 기능2 변경

- 이전

관리자가 옷 수량을 입력하면, 사용자가 실제 수량을 알 수 있는 기능

- 이후

관리자가 옷 수량을 입력하면, 옷 수량이 파일로 저장되어 관리하기 편하게 기능을 추가했다. 또한 사용자는 수량을 확인할 때 파일을 열어서 볼 수 있게 변경하였다.
또한 수량과, 메뉴에 숫자가 아닌 문자열을 입력했을 경우 오류가 뜨기 때문에 숫자만 입력해주라는 메시지를 만들었다.

- 사유

원래는 동적배열 벡터를 이용해서 수량을 관리하려고 했는데, 수업시간에 파일 입출력과 관련된 내용을 배우면서, 파일을 읽어 데이터를 객체에 저장하고, 객체들을 벡터에 저장 후 콘솔에 출력하는 코드를 실습으로 쳐본 적이 있다. 실습을 하면서 이번 기말 프로젝트에 이 내용을 적용하면, 사용자들이 조금 더 편하게 관리할 수 있을 거 같아서 이 기능을 추가하였다. 또한 숫자가 아닌 문자열을 입력할 때는 프로그램에 오류가 뜨기 때문에 오류를 막을 수 있는 기능도 추가했다.

5. 프로젝트 일정

업무		11/3	11/ 26	12/4	12/17	12/20	12/23
제안서 작성		완료					
기능1	세부기능1	완료					
	세부기능2	완료					
	세부기능3	완료					
	세부기능4	완료					
	세부기능5	완료					
기능2	세부기능 1,2,3,4		완료				
기능3	세부기능 1,2			---------(수정 필요)---------->			
기능4	세부기능1,2					---------->	
최종	마무리					---------->	