

C++프로그래밍및실습

물품 대여 관리 시스템

진척 보고서 #1



제출일자: 2023-11-26

이름: 안호균

학번: 190025

1. 프로젝트 목표

1) 배경 및 필요성

저는 현재 소프트웨어공학과 학생회의 일원으로서 활동하고 있습니다. 소프트웨어공학과 학생회는 학생들의 편의성 증진을 위해 여러가지 학생 사업을 진행하는데, 그 중 하나로 학과 학생들에게 우산, 보조배터리, 돗자리와 같이 학생들이 특정 상황에 필요 할만한 물품을 대여해주는 ‘물품 대여 사업’이 있습니다.

하지만 현재 물품의 대여 및 반환을 기록하는 일련의 과정은 모두 수기로 작성되고 있어, 이로 인해 발생하는 여러가지 문제점들이 있었습니다. 물품을 대여한 학생들에 대한 정보 관리, 대여 가능한 물품의 수량 관리, 물품 분실 및 파손 등의 상태 관리, 대여 가능한 물품이 더 이상 남아있지 않을 때 학생들이 이를 인지하지 못하고 방문을 하는 경우 등 여러 문제가 발생했으며, 무엇보다 물품의 대여 및 반납 기록을 관리하면서 발생하는 학생회 인원의 인력 및 시간적 낭비가 지속적으로 발생했습니다.

이러한 배경 속에서 학생회를 위한 효율성을 높인 자동화된 대여 물품 관리 시스템을 고안하게 되었습니다. 이를 통해 물품의 대여 및 반납 과정을 체계적으로 관리하고, 물품의 상태를 관리하며, 대여한 학생의 정보를 정확하게 관리하게 될 것이라는 기대를 가지고 있습니다.

2) 프로젝트 목표

1. 체계적인 물품 관리

- 물품 관리 시스템의 도입으로, 학생회가 보유하고 있는 모든 대여 물품의 현황을 실시간으로 파악하고 관리할 수 있도록 합니다.
- 물품의 추가, 대여, 반환 및 현황 조회 등의 기능을 통합하여 제공하여 관리 과정의 복잡성을 최소화합니다.

2. 높은 데이터의 정확성 및 투명성 확보

- 대여 및 반환 기록을 시스템화 하여 기존의 수기 작성 방식에 비해 오류를 줄이고 데이터 안전성을 높일 수 있으며, 물품 대여자 정보를 기록으로 남김으로써 해당 물품 관련 정보의 투명성을 제공하여 불필요한 오해나 실수를 방지합니다.

3. 관리 편의성 제고

- 학생회 관리 인원이 시스템을 통해 대여 가능한 물품의 수량, 상태 등의 정보를 손쉽게 확인하고, 대여 및 반환 절차를 간편하게 관리할 수 있도록 합니다.

4. 장기적인 데이터 관리 및 활용

- 시스템의 기록된 데이터를 파일에 저장함으로써 장기적으로 보관하며, 학생들의 물품 대여 추세를 파악하여 물품의 수량 추가 여부를 판단하는데 활용할 수 있습니다.

3) 차별점

소프트웨어공학과 학생회의 물품 대여 관리 시스템은 학생회의 기능적 필요성에 의해 최적화되어 설계되었습니다. 특히, 이 시스템은 물품의 대여 및 반환 정보를 파일로 저장하여 기록하므로, 학생회 내외부에 투명하게 정보 제공이 가능하며, 이를 통해 발생할 수 있는 오해나 갈등을 미연에 방지할 수 있습니다. 또한, 본 시스템을 통해 데이터 분석이 가능합니다. 학생회에서는 이를 활용해 대여 추세를 분석할 수 있으며, 이를 기반으로 물품의 수량 추가 여부를 판단할 수 있으며, 수요 분석을 기반으로 앞으로 추가하면 좋을 물품을 결정하는 데에도 활용할 수 있습니다.

2. 기능 계획

1. 물품 추가

- 학생회 관리자는 새로운 대여 물품을 시스템에 등록할 수 있습니다. 물품 이름, 총 재고량 등 필요한 정보를 입력하여 추가합니다.

2. 물품 대여

- 학생들이 원하는 물품을 대여할 수 있습니다. 물품을 대여할 때 소프트웨어공학과 소속임을 확인할 학과 정보, 학번, 이름을 함께 입력하여 대여 기록을 남깁니다.

3. 물품 반납

- 대여한 물품을 반납하는 기능입니다. 반납 시 대여자의 학번을 입력 받아 반납 대상자를 파악하고, 해당 물품의 재고를 복구합니다.

4. 모든 물품 리스트 보기

- 등록된 모든 물품의 목록을 조회할 수 있습니다. 각 물품의 총 수량, 재고량, 대여 중인 수량 등 상세한 정보가 함께 표시됩니다.

5. 대여 중인 물품 리스트 보기

- 현재 대여 중인 물품의 목록과 대여자 정보를 조회할 수 있습니다. 학생회 관리자는 이 기능을 통해 대여 현황을 확인할 수 있습니다.

6. 파일에 저장하기

- 현재 시스템의 데이터, 즉 물품 정보와 대여 정보 및 물품 재고를 파일로 저장할 수 있습니다. 이를 통해 데이터 백업과 이후 데이터 복구가 가능합니다.

7. 파일에서 불러오기

- 저장된 파일을 불러와 시스템의 데이터를 복구할 수 있습니다. 예기치 못한 상황에서 데이터 손실이 발생했을 때, 이 기능을 활용하여 데이터를 복원합니다.

8. 프로그램 종료

- 물품 대여 관리 프로그램을 안전하게 종료합니다.

3. 진척사항

1) 기능 구현

0. 라이브러리, 네임스페이스, 클래스 선언 및 정의

```
#include <iostream>
#include <vector>
#include <string>
#include <chrono>
#include <iomanip>

using namespace std;
using namespace chrono;
```

- **라이브러리:** 입출력, 벡터, 문자열, 시간 요소, 유효 숫자 포매팅을 위해 필요한 라이브러리들을 포함시킨다.
- **네임스페이스:** 편의를 위해 std와 chrono의 네임스페이스를 사용한다.

```
// Item 클래스: 물품 정보를 저장한다.
class Item {
public:
    string name; // 물품 이름
    int totalQuantity; // 물품의 총 수량
    int availableQuantity; // 현재 대여 가능한 물품 수량

    // 생성자: 물품 이름과 수량을 초기화한다.
    Item(string n, int q) : name(n), totalQuantity(q), availableQuantity(q) {}
};

// Renter 클래스: 대여자 정보를 저장한다.
class Renter {
public:
    string department; // 대여자의 학과
    string studentID; // 대여자의 학번
    string studentName; // 대여자의 이름
    string itemName; // 대여한 물품 이름
    system_clock::time_point rentedTime; // 대여한 시간
    system_clock::time_point dueTime; // 반납 예정 시간

    // 생성자: 대여자 정보와 대여 시간, 반납 예정 시간을 초기화한다.
    Renter(string dept, string id, string name, string item, system_clock::time_point rented, system_clock::time_point due)
    : department(dept), studentID(id), studentName(name), itemName(item), rentedTime(rented), dueTime(due) {}
};

vector<Item> items; // 모든 물품을 저장하는 벡터
vector<Renter> renters; // 모든 대여자를 저장하는 벡터
```

- **입출력:**
 1. name(Item): 물품 이름
 2. totalQuantity(Item): 물품의 총 수량
 3. availableQuantity(Item): 현재 대여 가능한 물품 수량

4. department(Renter): 대여자의 학과
5. studentID(Renter): 대여자의 학번
6. studentName(Renter): 대여자의 이름
7. itemName(Renter): 대여한 물품의 이름
8. rentedTime(Renter): 대여한 시간
9. dueTime(Renter): 반납 예정 시간

● **설명:** 물품 정보를 저장하는 'Item' 클래스와 물품 대여자의 정보를 저장하는 'Renter' 클래스를 선언하고 정보를 관리하기 위한 변수 및 함수를 선언합니다.

● **적용된 배운 내용:**

1. **클래스:**

- Item과 Renter는 클래스를 활용하여 데이터와 관련 기능을 하나의 단위로 묶어 표현합니다.
- Item 클래스는 물품에 대한 정보를, Renter 클래스는 물품을 대여한 사람의 정보를 저장합니다.

2. **멤버 변수:**

- Item 클래스는 name, totalQuantity, availableQuantity와 같은 멤버 변수를 가지고 있고, 이는 각 물품의 이름, 총 수량, 사용 가능한 수량을 표현합니다.
- Renter 클래스는 department, studentID, studentName, itemName, rentedTime, dueTime을 멤버 변수로 가지고 있어, 대여자의 정보 및 대여된 물품과 대여 시간, 반납 예정 시간을 표현합니다.

3. **생성자:**

- 두 클래스 모두 생성자를 정의하여 객체가 생성될 때 멤버 변수들을 초기화합니다.
- 예를 들어, Item 클래스의 생성자는 물품의 이름과 수량을 인자로 받아 객체를 초기화합니다.

4. **벡터:**

- vector<Item>과 vector<Renter>는 표준 라이브러리의 일부인 벡터를 사용합니다.

- 이 벡터들은 Item 객체와 Renter 객체를 동적으로 저장하는데 사용되며, 물품 목록과 대여자 목록을 관리하는 데 사용됩니다.

5. 기타 C++ 표준 라이브러리 기능:

- string은 문자열을 저장하고 처리하는 데 사용됩니다.
- 배운 내용은 아니지만, C++ 표준 라이브러리 중 chrono 라이브러리는 시간 관련 기능을 제공하여, 대여 시간과 반납 시간을 처리하는 데 사용됩니다.

1. 물품 추가 기능

```
// 물품을 추가하는 함수
void addItem() {
    string name;
    int quantity;

    cout << "₩n[물 품 추가]₩n";
    cout << "물 품 이름: ";
    cin >> name;
    cout << "물 품 수 량: ";
    cin >> quantity;

    items.push_back(Item(name, quantity));
}
```

● 입출력:

1. name: 물품 이름
2. quantity: 물품의 갯수

- **설명:** 이 기능은 프로그램의 관리자가 새로운 물품을 시스템에 추가할 수 있도록 하는 기능입니다. 사용자는 물품의 이름과 수량을 입력하고, 이 정보는 items 벡터에 저장됩니다.

● 적용된 배운 내용:

1. 함수:

- addItem은 프로그램에 새로운 물품을 추가하는 별도의 함수로 구현되어 있습니다. 이 함수는 사용자로부터 물품의 이름과 수량을 입력받아 처리합니다.

2. 입출력:

- cin과 cout을 사용하여 사용자로 부터 데이터를 입력받습니다. (추가할 물품 이름, 수량

3. 클래스:

- 물품 정보는 Item 클래스의 인스턴스로 저장됩니다. 이 클래스는 물품의 이름(name), 총 수량(totalQuantity), 현재 사용 가능한 수량(availableQuantity) 등의 멤버 변수를 가지고 있습니다.

4. 생성자:

- Item 클래스에 정의된 생성자는 물품 이름과 수량을 초기화합니다. Item(string n, int q) 형식의 생성자가 사용되며, 이는 items 벡터에 새 Item 객체를 추가할 때 호출됩니다.

5. 벡터:

- items는 vector<Item> 타입의 벡터로, 여러 Item 객체를 저장합니다. 사용자가 새 물품을 추가하면, 이 벡터에 새 Item 객체가 push_back 메서드를 통해 추가됩니다.

2. 물품 대여 기능

```
// 물품을 대여하는 함수
void rentItem() {
    string department, studentID, studentName, itemName;

    cout << "₩₩₩ 물품 대여₩₩₩\n";
    cout << "학과: ";
    cin >> department;
    cout << "학번: ";
    cin >> studentID;
    cout << "이름: ";
    cin >> studentName;
    cout << "대여하려는 물품 이름: ";
    cin >> itemName;

    bool itemFound = false;
    for (auto& item : items) {
        if (item.name == itemName) {
            itemFound = true;
            if (item.availableQuantity > 0) {
                item.availableQuantity--;

                auto now = system_clock::now();
                auto due = now + hours(48); // 48시간 후로 설정

                time_t dueTime_t = system_clock::to_time_t(due);
                struct tm dueTm;
                localtime_s(&dueTm, &dueTime_t);

                renters.push_back(Renter(department, studentID, studentName, itemName, now, due));
                cout << "대여 완료! 반납 예정 시간: " << put_time(&dueTm, "XY-Zm-Zd ZH:XM:XS") << endl;
                return;
            }
        }
        else {
            cout << "죄송합니다. 현재 해당 물품은 대여할 수 없습니다. (재고 부족)" << endl;
            return;
        }
    }

    if (!itemFound) {
        cout << "해당 물품을 찾을 수 없습니다." << endl;
    }
}
```


- **입출력:**

1. department: 대여자의 학과
2. studentID: 대여자의 학번
3. studentName: 대여자의 이름
4. itemName: 대여한 물품의 이름

- **설명:** 이 기능은 사용자가 시스템에 등록된 물품을 대여할 수 있게 하는 기능입니다. 물품을 대여할 때, 사용자는 사용자의 학과, 학번, 이름, 대여하려는 물품 이름을 입력하고, 반복문을 통해 items 벡터에서 물품을 찾습니다. 물품을 찾았다면 사용자에게 대여를 해주고 반납 예정 시간(48시간 후)를 설정합니다. 물품을 발견하지 못했다면 사용자에게 해당 물품이 대여 불가능함을 알립니다.

- **적용된 배운 내용:**

1. **함수:**

- rentItem은 물품을 대여하는 별도의 함수로 구현되어 있으며, 사용자로부터 대여자 정보와 대여할 물품의 이름을 입력받아 처리합니다.

2. **입출력:**

- cin과 cout을 사용하여 사용자로부터 데이터를 입력받고 (대여자의 학과, 학번, 이름, 대여할 물품 이름), 처리 결과를 화면에 출력합니다.

3. **반복문:**

- for 반복문은 items 벡터를 순회하며 사용자가 입력한 물품 이름과 일치하는 물품을 찾습니다.

4. **조건문:**

- if 조건문은 찾은 물품의 수량을 검사하여 대여 가능한지 여부를 판단합니다.

5. **벡터:**

- items 벡터는 등록된 모든 물품의 정보를 저장하며, renters 벡터는 대여된 물품과 대여자의 정보를 저장합니다.

3. 물품 반납 기능

```
// 물품을 반납하는 함수
void returnItem() {
    string studentID, itemName;

    cout << "\n[물품 반납]\n";
    cout << "학번: ";
    cin >> studentID;
    cout << "반납하려는 물품 이름: ";
    cin >> itemName;

    // 대여자 목록에서 해당 대여자 찾기
    for (size_t i = 0; i < renters.size(); ++i) {
        if (renters[i].studentID == studentID && renters[i].itemName == itemName) {
            // 해당 물품의 수량 증가
            for (auto& item : items) {
                if (item.name == itemName) {
                    item.availableQuantity++;
                    cout << "반납 완료!" << endl;
                    renters.erase(renters.begin() + i); // 대여자 목록에서 제거
                    return;
                }
            }
        }
    }

    cout << "반납할 물품을 찾을 수 없습니다." << endl;
}
```

- **입출력:**

1. studentID: 반납자의 학번
2. itemName: 반납하려는 물품의 이름

- **설명:** 이 기능은 사용자가 대여한 물품을 시스템에 반납할 수 있게 하는 기능입니다. 사용자는 반납을 위해 학번, 반납하려는 물품 이름을 입력하고, 대여자 목록을 담은 renters 벡터에서 해당 물품을 대여한 사용자를 찾았다면, 해당 물품의 수량을 원복시키고, 반납 완료 메시지를 출력하며 사용자를 대여자 목록에서 제거한다. 정보가 일치하지 않다면 반납할 물품을 찾을 수 없음을 알린다.

- **적용된 배운 내용:**

1. 함수:

- returnItem은 물품 반납 처리를 담당하는 별도의 함수로 구현되어 있습니다. 이 함수는 사용자로부터 반납할 물품의 이름과 대여자 정보를 입력받아 처리합니다.

2. **입출력:**

- cin과 cout을 사용하여 사용자로 부터 데이터를 입력받고 (학번, 반납하려는 물품의 이름) 처리 결과를 화면에 출력합니다.

3. **반복문:**

- for 반복문은 renters 벡터를 순회하며 해당 학번을 가진 대여자와 일치하는 물품을 찾습니다.

4. **조건문:**

- if 조건문은 대여자의 대여 정보가 입력된 정보와 일치하는지 검사합니다.

5. **벡터:**

- renters 벡터는 현재 대여 중인 모든 물품과 대여자의 정보를 저장합니다. 이 벡터에서 해당 대여자를 찾아 정보를 업데이트하거나 제거합니다.
- items 벡터는 등록된 모든 물품의 정보를 저장하며, 반납된 물품의 수량을 업데이트합니다.
- 반납처리가 완료되면, renters 벡터에서 해당 대여자 정보를 erase 메서드를 사용하여 제거합니다.

6. **클래스의 객체 접근:**

- Item 클래스와 Renter 클래스의 객체에 접근하여 물품의 수량을 수정하거나 대여자 정보를 업데이트합니다.

4. 모든 물품 보기 기능

```
// 모든 물품의 목록을 보여주는 함수
void viewItems() {
    cout << "\n[모든 물품 리스트]\n";
    for (const auto& item : items) {
        cout << item.name << " - 총 수량: " << item.totalQuantity << ", 남은 수량: " << item.availableQuantity
            << ", 대여 중: " << item.totalQuantity - item.availableQuantity << endl;
    }
}
```

- **입출력:**

1. items: 보유 중인 모든 물품을 담은 벡터

- **설명:** 이 기능은 사용자에게 시스템에 등록된 모든 물품의 목록을 보여주는 기능입니다. 물품 목록을 담은 items 벡터에서 정보를 꺼내 시스템에서 보유 중인 모든 물품에 대한 각각의 총 수량, 남은 수량, 대여중인 물품의 수를 출력합니다.

- **적용된 배운 내용:**

1. 함수:

- viewItems는 저장된 모든 물품의 리스트를 출력하는 별도의 함수로 구현되어 있습니다.

2. 반복문:

- for 반복문을 사용하여 items 벡터에 저장된 모든 Item 객체들을 순회합니다.

3. 벡터:

- items는 vector<Item> 타입의 벡터로, 다수의 Item 객체들을 저장합니다. 이 벡터는 시스템에 등록된 모든 물품의 정보를 담고 있습니다.

4. 입출력:

- cout을 사용하여 각 물품의 상세 정보(물품 이름, 총 수량, 사용 가능한 수량)를 화면에 출력합니다.

5. 클래스의 객체 접근:

- 반복문 내에서 각 Item 객체에 접근하여 물품의 이름, 총 수량, 사용 가능한 수량 등의 정보를 가져옵니다.

5. 대여 중인 물품 보기 기능

```
// 현재 대여 중인 물품과 대여자 정보를 보여주는 함수
void viewRenters() {
    cout << "ㄷㄷ[대여 중인 물품 리스트]ㄷㄷ";
    for (const auto& renter : renters) {
        time_t rentedTime_t = system_clock::to_time_t(renter.rentedTime);
        struct tm rentedTm;
        localtime_s(&rentedTm, &rentedTime_t);

        time_t dueTime_t = system_clock::to_time_t(renter.dueTime);
        struct tm dueTm;
        localtime_s(&dueTm, &dueTime_t);

        cout << "학과: " << renter.department
              << ", 학번: " << renter.studentID
              << ", 이름: " << renter.studentName
              << ", 대여 물품: " << renter.itemName << endl;
        cout << "대여 시간: " << put_time(&rentedTm, "%Y-%m-%d %H:%M:%S")
              << ", 반납 예정 시간: " << put_time(&dueTm, "%Y-%m-%d %H:%M:%S") << endl;
    }
}
```

- **입출력:**

1. renters: 대여자의 목록을 담은 벡터

- **설명:** 이 기능은 현재 대여되어 있는 모든 물품과 해당 대여자의 정보를 보여주는 기능입니다. 이 때, 대여자가 물건을 대여한 시간 및 반납 예정 시간을 불러와 대여자의 정보와 함께 출력합니다.

- **적용된 배운 내용:**

1. 함수:

- viewRenters는 현재 대여 중인 모든 물품과 대여자의 정보를 출력하는 별도의 함수로 구현되어 있습니다.

2. 반복문:

- for 반복문을 사용하여 renters 벡터에 저장된 모든 Renter 객체들을 순회합니다.

3. 벡터:

- renters는 vector<Renter> 타입의 벡터로, 대여 중인 물품과 대여자의 정보를 저장합니다.

4. 입출력:

- cout을 사용하여 각 대여자의 정보(학과, 학번, 이름, 대여한 물품) 및 대여 시간과 반납 예정 시간을 화면에 출력합니다.

5. 클래스의 객체 접근:

- 반복문 내에서 각 Renter 객체에 접근하여 대여자의 정보와 물품 대여 관련 정보를 가져옵니다.

메인 함수

```
// 메인 함수: 사용자 인터페이스와 프로그램의 메인 루프를 담당
int main() {
    while (true) {
        cout << "물품 관리 프로그램" << endl;
        cout << "1. 물품 추가" << endl;
        cout << "2. 물품 대여" << endl;
        cout << "3. 물품 반납" << endl;
        cout << "4. 모든 물품 리스트 보기" << endl;
        cout << "5. 대여 중인 물품 리스트 보기" << endl;
        cout << "0. 종료" << endl;
        cout << "선택: ";

        int choice;
        cin >> choice;

        switch (choice) {
            case 1: addItem(); break;
            case 2: rentItem(); break;
            case 3: returnItem(); break;
            case 4: viewItems(); break;
            case 5: viewRenters(); break;
            case 0: return 0;
            default: cout << "잘못된 선택입니다." << endl;
        }
    }

    return 0;
}
```

● 입출력:

1. choice: 사용자가 선택할 기능의 번호

- 설명: 메인 함수는 프로그램의 사용자 인터페이스 기능을 합니다. 사용자의 선택에 따라 앞서 정의한 기능별 함수들을 실행하도록 합니다.

- **적용된 배운 내용:**

1. 함수:

- 앞서 구현한 기능별 함수들을 사용자의 선택에 따라 실행합니다.

2. 반복문:

- while(true) 반복문을 사용하여 프로그램을 종료하는 명령이 올 때까지 무한 반복을 합니다.

3. 입출력:

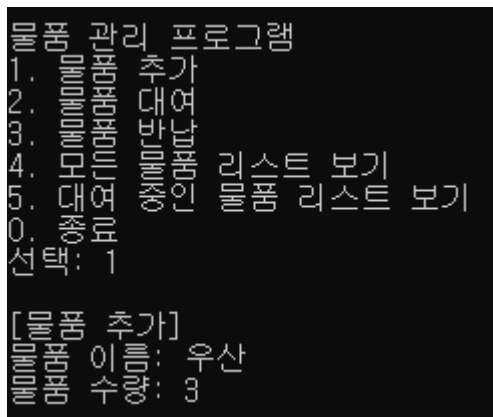
- cout을 사용하여 사용자의 기능 선택을 유도하는 사용자 인터페이스를 출력합니다.

4. 조건문:

- switch 문을 이용하여 사용자가 원하는 기능을 숫자로 선택하도록 합니다.

2) 테스트 결과

1. 물품 추가 기능

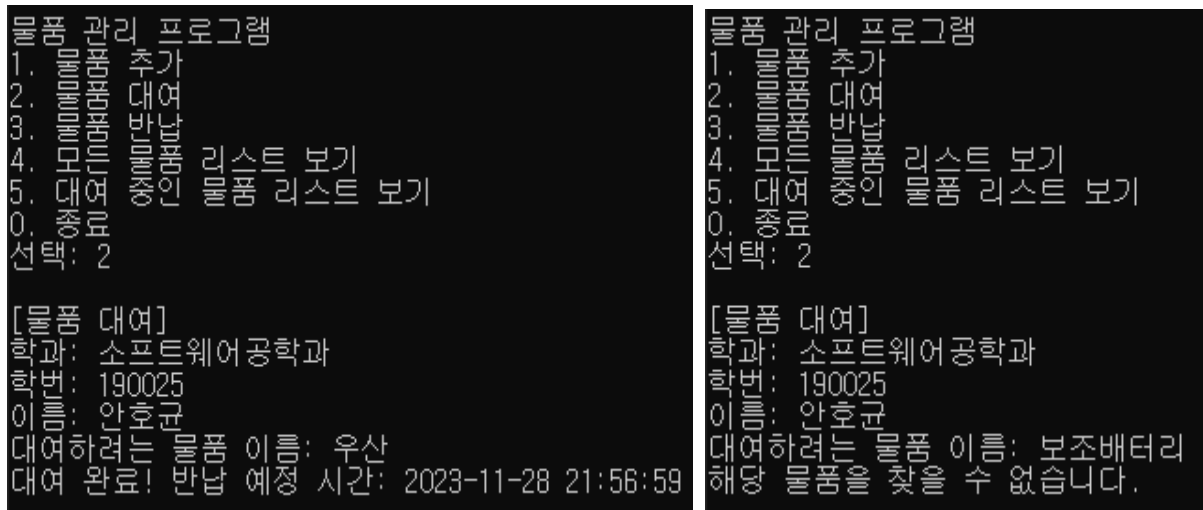


```
물품 관리 프로그램
1. 물품 추가
2. 물품 대여
3. 물품 반납
4. 모든 물품 리스트 보기
5. 대여 중인 물품 리스트 보기
0. 종료
선택: 1

[물품 추가]
물품 이름: 우산
물품 수량: 3
```

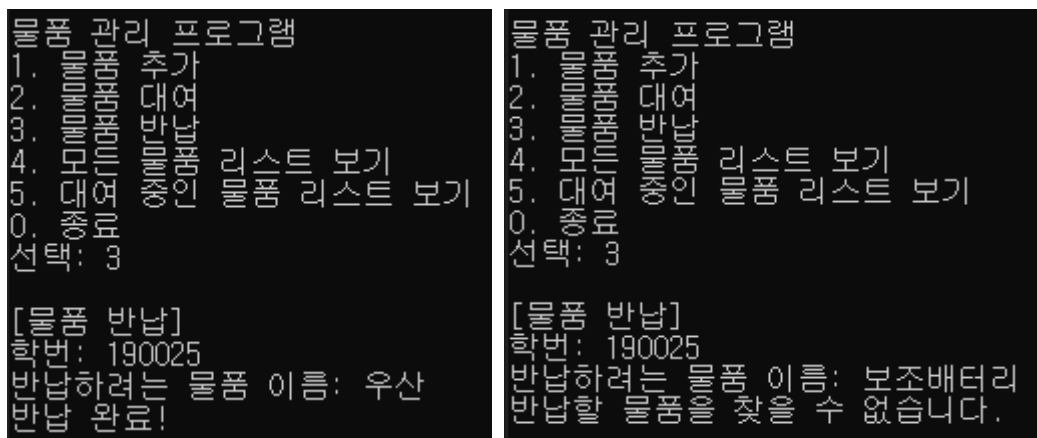
- **설명:** 프로그램의 관리자가 1번 기능을 선택하면, "물품 추가" 기능에 해당하는 함수를 실행합니다. 추가할 물품 이름과 수량을 입력하여 물품을 시스템에 추가합니다.

2. 물품 대여 기능



- **설명:** 사용자가 2번 기능을 선택하면, “물품 대여” 기능에 해당하는 함수를 실행합니다. 대여자의 학과, 학번, 이름을 입력하고, 대여하려는 물품 이름을 입력하여 해당 물품을 대여합니다. 이 때 현재 시간을 대여한 시간으로 저장하고, 현재 시간 기준 48 시간 이후를 반납 예정 시간으로 설정하여 사용자에게 출력합니다. 대여하려는 물품이 존재하지 않으면 물품을 찾을 수 없음을 출력합니다.

3. 물품 반납 기능



- **설명:** 사용자가 3번 기능을 선택하면, “물품 반납” 기능에 해당하는 함수를 실행합니다. 반납자의 학번, 반납하려는 물품 이름을 입력하여 해당 물품을 반납합니다. 반납이 완료되면 완료 메시지를 사용자에게 출력하고, 해당 물건이 없다면 찾을 수 없음을 출력합니다.

4. 모든 물품 보기 기능

```
물품 관리 프로그램
1. 물품 추가
2. 물품 대여
3. 물품 반납
4. 모든 물품 리스트 보기
5. 대여 중인 물품 리스트 보기
0. 종료
선택: 4

[모든 물품 리스트]
우산 - 총 수량: 3, 남은 수량: 3, 대여 중: 0
```

- **설명:** 사용자가 4번 기능을 선택하면, "모든 물품 리스트 보기" 기능에 해당하는 함수를 실행합니다. 사용자에게 시스템에 등록된 모든 물품에 대한 정보(총 수량, 남은 수량, 대여 중인 수량)를 출력합니다.

5. 대여 중인 물품 보기 기능

```
물품 관리 프로그램
1. 물품 추가
2. 물품 대여
3. 물품 반납
4. 모든 물품 리스트 보기
5. 대여 중인 물품 리스트 보기
0. 종료
선택: 5

[대여 중인 물품 리스트]
학과: 소프트웨어공학과, 학번: 190025, 이름: 안호균, 대여 물품: 우산
대여 시간: 2023-11-26 22:23:22, 반납 예정 시간: 2023-11-28 22:23:22
```

- **설명:** 사용자가 5번 기능을 선택하면, "대여 중인 물품 리스트 보기" 기능에 해당하는 함수를 실행합니다. 사용자에게 현재 물품을 대여중인 대여자에 대한 정보(학과, 학번, 이름, 대여 물품, 대여 시간, 반납 예정 시간)를 출력합니다.

4. 계획 대비 변경 사항

1) 물품 대여 시간 및 반납 예정 시간 표시 추가 완료

- **이전:** 물품 대여 시 대여자의 정보 및 물품 이름만으로 관리하였습니다.
- **이후:** 사용자의 물품 대여 시간 및 반납 예정 시간을 추가하였습니다.
- **사유:** 대여 시간과 반납 예정 시간을 기록함으로써 물품 대여 및 관리 프로세스의 효율성을 높였습니다. 또한, 시간을 기록하면 물품 대여 사업을 더욱 투명하게 운영할 수 있다는 장점이 있습니다. 더불어 반납 예정 시간을 넘길 경우에 대해 패널티를 적용하는 것을 추가하려고 합니다.

2) 물품 삭제 및 수정 기능 추가 예정

- **이전:** 현재 물품 관리는 추가만 가능하여 잘못 등록한 물품에 대한 수정이 불가능합니다.
- **이후:** 물품 삭제 및 수정 기능을 추가하여 물품 관리를 용이하게 합니다.
- **사유:** 관리자가 실수로 잘못된 정보를 입력했을 경우, 이를 수정할 수 있는 기능이 있어야 합니다. 물품 추가 과정에서 발생할 수 있는 실수를 바로잡을 수 있게 함으로써, 시스템의 정확성을 보장할 수 있습니다. 앞으로 추가할 파일을 통한 데이터 관리를 통해서도 물품의 삭제 및 수정을 할 수 있겠지만, 시스템을 통해서 정보를 수정하는 것이 더욱 직관적이고 관리에 용이하므로 해당 기능 추가를 계획하게 되었습니다.

3) 관리자 인증 기능 추가 예정

- **이전:** 물품 추가 및 대여 중인 물품 보기 출력 시 나오는 대여자 정보에 대한 접근을 모두가 할 수 있습니다.
- **이후:** 시스템 관리자 인증을 추가하여 민감한 데이터를 관리자만 접근할 수 있도록 합니다.
- **사유:** 관리자 인증은 민감한 데이터에 대한 보안을 강화하는데 필수적입니다. 보안이 보장되지 않으면 특정 사용자가 물품에 대한 관리를 오용하거나 남용할 수 있어 데이터의 신뢰성이 떨어질 수 있어 관리자 인증 기능을 추가할 것입니다.

+) 이외 앞으로 개선할 사항들

- 헤더파일을 통한 함수 관리
- 사용자 인터페이스 개선
- 완성도를 높일 여러 기능 추가

5. 프로젝트 일정

업무	11/3	11/26	12/4	12/23
제안서 작성	완료			
프로그램 종료 기능		완료		
물품 추가 기능		완료		
물품 대여 기능		완료		
물품 반납 기능		완료		
모든 물품 보기 기능		완료		
대여 중인 물품 보기 기능		완료		
물품 대여 시간 관리 기능		완료		
1차 중간 보고서 작성		완료		
파일에 저장하는 기능				
파일에서 불러오는 기능				
물품 연체 패널티 기능				
물품 삭제 및 수정 기능				
관리자 인증 기능				
2차 중간 보고서 작성				
헤더파일을 통한 함수 관리				
사용자 인터페이스 개선				
프로그램 최종 테스트 및 추가 기능 구현(필요시)				
최종 보고서 작성				