

**1. 던파 캐릭터 인벤토리를 본 후 다음을 작성하세요.**

가. 인벤토리 클래스를 설계하세요

- C++로 작성하며 전체 구조를 쉽게 확인할 수 있게 작성하세요

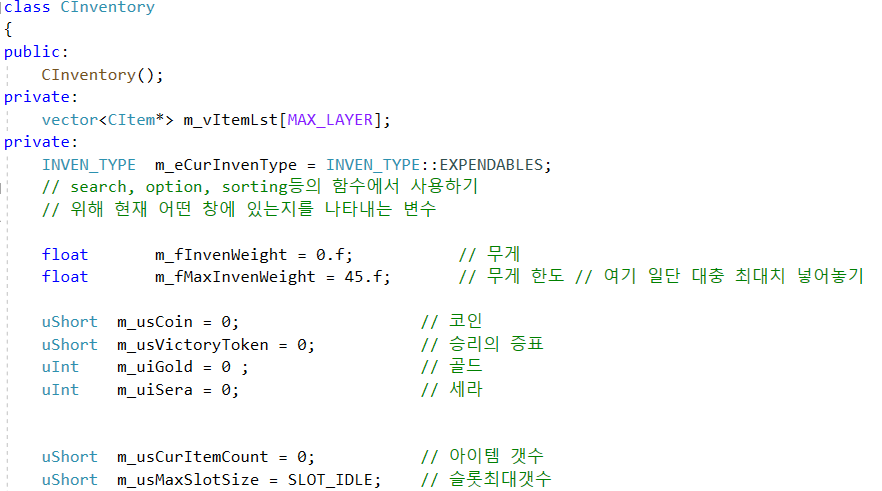
- 아이템 및 인벤토리 내용은 특징적인 부분만 포함되면 됩니다.

- 구현부에는 인벤토리에 아이템 추가/삭제하는 기능만 포함하세요

- 설계 이유를 간단하게 기술하세요

**1. 인벤토리 클래스**

텍스트이(가) 표시된 사진

자동 생성된 설명



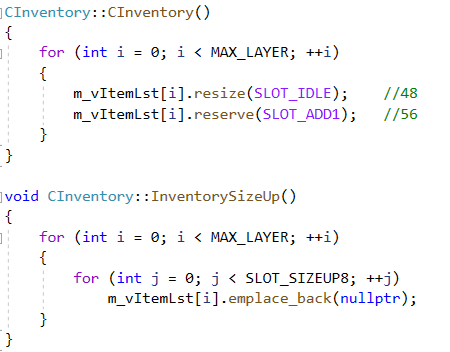
장비, 소모품과 같은 창의 개수는 고정적이라고 생각하여

배열로 사용하였고 INVEN\_TYPE이라는 enum class의 데이터를 멤버변수로(m\_CurInvenType) 사용하여 삭제 시에 현재 창을 기준으로 아이템을 검색하여 삭제할 수 있도록 구현하였습니다.

벡터를 사용한 이유는 index로 접근하여 slot의 번호만으로 처리할 수 있기 때문에 효율적이고

슬롯의 개수가 가변적이기 때문에 사용하였습니다.

**2. 인벤토리 생성**



인벤토리 생성과 동시에 장비/소모품 아이템 리스트를 resize, reserve해 주는데

그 이유는 vector는 일정 크기의 배열을 할당한 뒤 push\_back을 계속 하여 size가 capacity를

넘어가려 하면 2배 크기의 배열을 만들어 이동하는 식으로 구현 되어 있기 때문에 기본 슬롯

개수인 48만큼 resize하여 사용할 원소의 개수를 세팅하였고

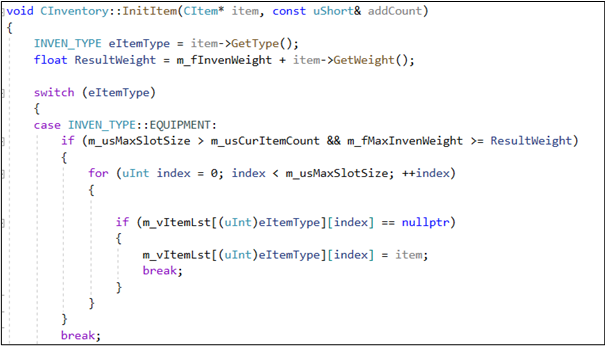
추가되는 슬롯의 최대치가 56이므로 56만큼 reserve하여 아이템이 추가적으로 들어왔을 때 스스로

공간을 늘리지 않게 하여 성능을 더 향상시켰습니다.

슬롯이 56으로 추가 되었을 때 56만큼 resize해준다면 nullptr로 초기화 되기 때문에 emplace\_back으로

하나하나 추가해줬습니다.

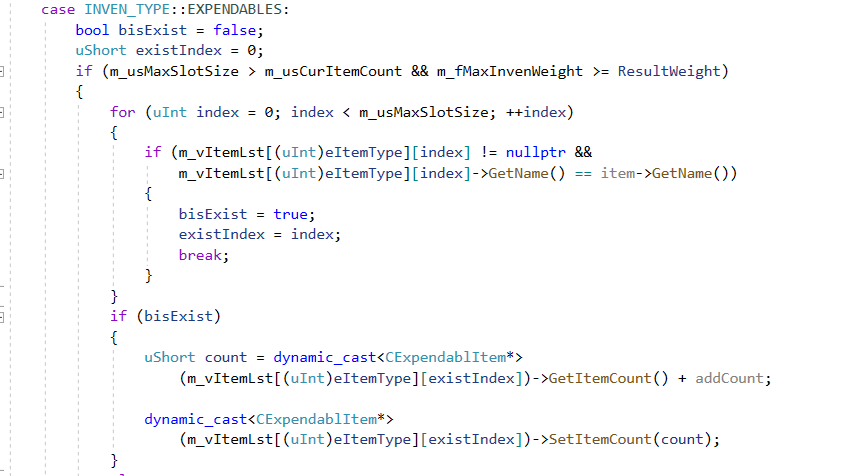
**3. 아이템 추가**

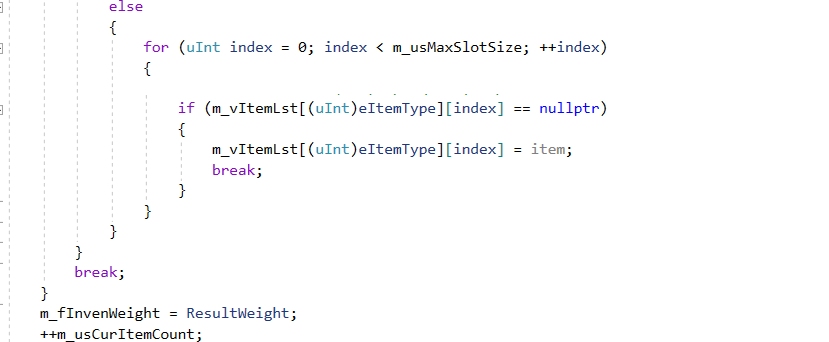


아이템을 추가하는 InitItem 함수이며 간단하게 장비와 소모품 추가만 구현하였습니다.

장비 추가의 경우 인벤토리의 아이템 개수가 최대치보다 작고 무게 최대치가 무게가 들어올 아이템과 기존 무게의 합산보다 크거나 같다는 조건을 통해 추가 가능 여부를 판단 하였고 추가 가능시에는

m\_vItemLst를 순회하며 빈 인덱스가 있을 경우 아이템을 추가하도록 구현하였습니다.





소모품의 경우 개수만 추가해야 할 경우가 있기 때문에 1차로 아이템을 순회하며 현재 있는 아이템인지

에 대한 검사를 한 후 존재한다면 인자로 받은 addCount 변수만큼 해당 아이템의 개수를 더해 주는

방식입니다.

만약 존재하지 않는다면 장비 아이템과 마찬가지고 아이템 리스트를 순회하며 빈공간에 넣어주게 됩

니다.

**4. 아이템 삭제**

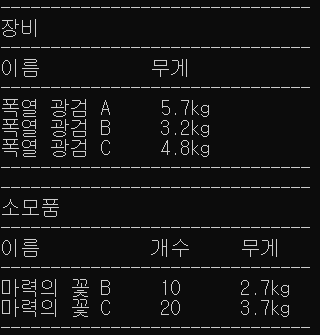
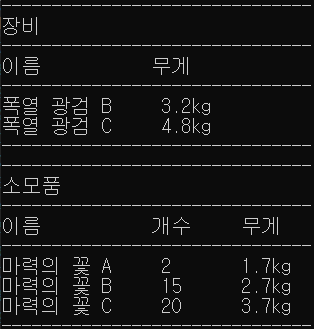
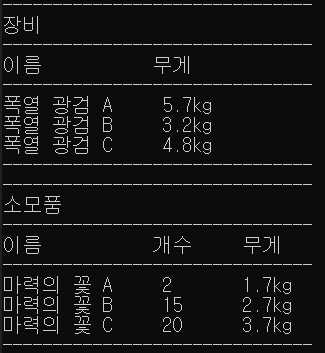


선택한 인덱스 번호와 삭제할 개수를 인자로 받게 되는데 현재 열어놓은 창을 판단하는 m\_eCurInvenType 변수를 사용하여 원하는 아이템 리스트를 사용할 수 있습니다.

장비 같은 경우 따로 개수가 없기 때문에 소모품에서만 현재 개수 – 삭제 개수 가 0일때와 아닌 경우를

판단하여 삭제 혹은 아이템의 개수를 줄여주는 방식으로 구현하였습니다.

**5. 실행 결과**



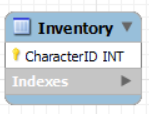
<아이템 삽입> <폭열 광검A 삭제> <마력의 꽃A,B 삭제>

나. 인벤토리 DB 테이블을 설계하세요.

- MySQL을 사용하세요

- 작성한 테이블 설계에 대한 이유를 간단하게 기술하세요

인벤토리를 소유하고 있는 캐릭터ID는 Null값을 허용하지 않으며 중복되지 않는 Unique한 식별번호여야 하고 음수인 경우가 없다고 생각하여 데이터 옵션을 설정하였습니다.

게임 상에서 인벤토리에 있는 골드, 세라, 코인, 승리의 증표 같은 경우는 인벤토리 내에서만 사용하는 데이터가 아니라고 생각했습니다.

예를 들어 골드는 경매장, 세라는 세라샵에서 사용되는 데이터 이기 때문에 캐릭터에 포함되어야 한다고 생각했습니다.

그리고 처음에는 아이템의 정보가 포함되어야 한다고 생각하여 내구도, 강화, 재련, 슬롯의 개수 등을 넣을 것을 고려하였지만 인벤토리 내의 아이템테이블에 포함되어야 하는 데이터라고 생각해서 제외하게 되었습니다.

**2. 간단한 던파 경매장 서버 검색기능을 구현하세요**

- C++로 작성하세요. 경매장 서버 모든 검색기능이 포함될 필요는 없으며 DB사용하지 않습니다.

- 메인 함수에서 여러 아이템을 메모리에 올리고 다음 두가지 기능 구현하세요

1. 이름으로 아이템 검색, 레벨은 50~60레벨, 레어리티는 유니크

2. 카테고리별로 아이템 검색

- 메모리 사용 및 검색속도를 고려하세요

- 구현에 대한 이유를 간단히 기술하세요.

(구현이 힘들 경우 던파 경매장 서버 구현 원리에 대해 생각하시는 대로 설명해 주셔도 됩니다)

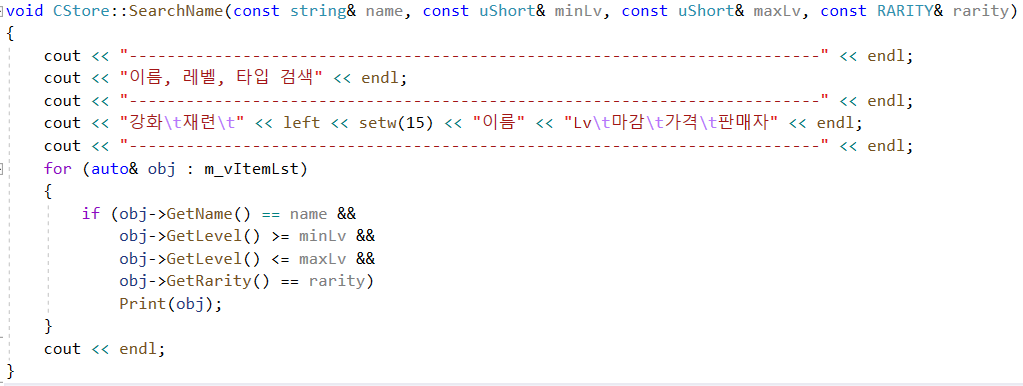
1. **경매장 클래스**

**텍스트이(가) 표시된 사진

자동 생성된 설명**경매장의 아이템을 vector에 저장하여 사용하였습니다.

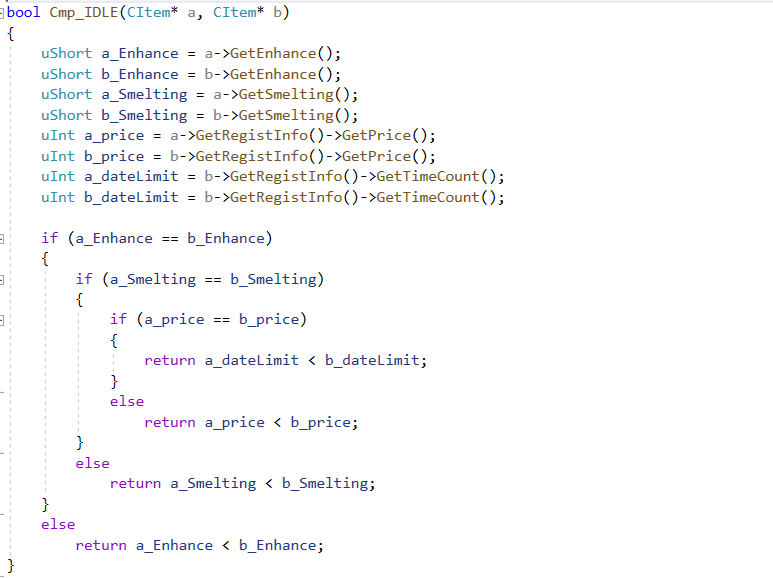
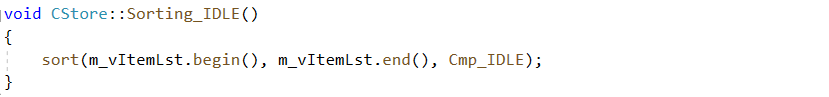
그 이유는 가변적이고 아이템 구매 시 index로 접근하여 사용할 수 있기 때문입니다.

1. **검색기능**

****

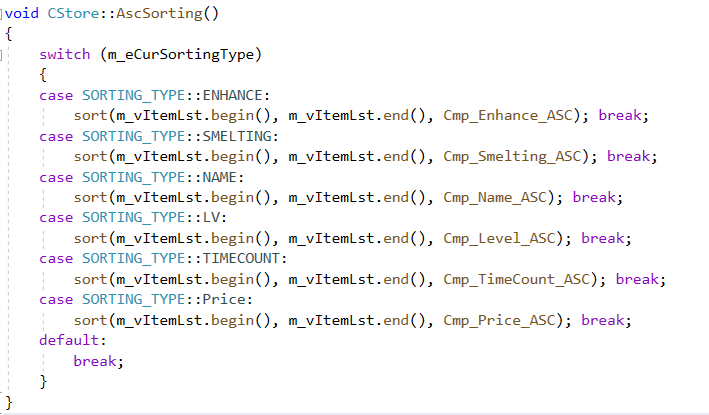
Search 관련 기능들은 아이템리스트를 순회하며 인자로 들어온 조건에 맞게 출력할 수 있도록 구현하였습니다.

1. **정렬기능**



경매장의 기본 정렬을 위해 Cmp\_IDLE 함수를 구현하여 강화, 재련, 가격, 남은 시간을 기준으로 정렬하여

사용하였습니다.



원하는 데이터에 따라 정렬할 수 도록 내림, 오름차순을 판단하는 변수와 현재 정렬되어 있는 판단하는

Enum class 변수를 통해 정렬하였습니다.

1. **실행 결과**

텍스트, 모니터, 점수판, 화면이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명

**3. 설문**

**# 아래 항목 중 다룰 수 있거나 다뤄본 적이 있는 항목을 기재하세요.**

1. **OS:** Windows

**나. 프로그래밍 언어:**

**C++, C**

**다. 프로그래밍 & 기타 도구 (IDE / 툴/ 프로파일러 / 소스형상관리):** 비주얼 스튜디오

**라. DBMS (NoSQL 포함):**

MariaDB, MsSQL

**마. 수강했던 개발관련 전공 과목:**

C 프로그래밍, C++ 프로그래밍, 윈도우프로그래밍, 자료구조

STL, 게임소프트웨어공학, 데이터베이스, 알고리즘,

게임서버프로그래밍, 네트워크 게임 프로그래밍

컴퓨터 그래픽스

**바. 최근 관심있는 기술:**

**블록체인**