

## 1.2 어느 정도 규모 있는 첫 C++ 프로그램

2024년 1월 7일 일요일    오후 8:59

클래스, 익셉션, 스트림, 벡터, 네임스페이스, 레퍼런스를 비롯한 다양한 기능을 활용하여 좀 더 완전하고 실전에서 쓸만한 C++ 프로그램으로 만들어본다.

### 1.2.1 직원 정보 관리 시스템

회사 직원의 정보를 관리하는 프로그램

- 직원 채용과 해고
- 직원 승진과 좌천
- 과거와 현재의 직원 모두 조회
- 현재 직원 모두 조회
- 과거 직원 모두 조회

### 1.2.2 Employee 클래스

Employee 클래스는 직원 한 명에 대한 모든 정보를 관리한다. 정보는 메서드로 조회하거나 변경한다.

콘솔에 정보를 출력하는 기능도 제공한다. 직원의 급여와 고용 상태 정보를 수정하는 메서드도 있다.

#### 1. Employee.cppm (.ixx)

- 1) Employee.cppm 모듈 인터페이스 파일은 Employee 클래스를 정의한다.

#### 2. Employee.cpp

- 1) 모듈 구현
- 2) 게터와 세터가 정의되어 있다면 같은 클래스 안이라도 가급적 이를 활용하는 것이 스타일 측면에서 바람직하다.
- 3) 코드가 아무리 단순하더라도 데이터 멤버를 public으로 선언한 것보다 게터와 세터를 구현하는 것이 낫다.
  - 예를 들어 setSalary() 메서드에서 경계값 검사를 하는 경우를 생각해보자 게터와 세터가 있다면 중단점을 설정해서 급여를 조회하거나 설정하는 시점의 값을 확인할 수 있기 때문에 디버깅하기 훨씬 쉽다.
  - 클래스에서 데이터를 저장하는 방식을 변경할 때 다른 코드는 건드릴 필요 없이 게터와 세터만 수정하면 된다.

#### 3. EmployeeTest.cpp

- 1) 작성한 클래스마다 테스트하는 코드도 함께 작성하면 좋다.

### 1.2.3 Database 클래스

Database 클래스는 표준 라이브러리에서 제공하는 std::vector 클래스를 이용하여 employee 객체를 저장한다.

#### 1. Database.cppm

- 1) 성과 이름을 입력하여 신입 직원에 대한 정보를 쉽게 추가하는 기능을 제공 (신

입 직원에 대한 레퍼런스를 리턴)

2) `getEmployee()` 메서드를 호출해서 직원 객체에 대한 레퍼런스를 구할 수 있다.

- 직원 번호로 조회
- 성과 이름으로 조회

3) 모든 직원에 대한 중앙 저장소이기 때문에 전체 직원 정보, 현재 재직 중인 직원 정보, 퇴사한 직원 정보를 출력하는 메서드를 제공한다.

4) `private` 데이터 멤버 정의

## 2. Database.cpp

1) `addEmployee()` 메서드는 새로운 `Employee` 객체를 생성해서 정보를 알맞게 채우고, 이를 `vector`에 추가한다.

2) `m_nextEmployeeNumber` 데이터 멤버는 한 번 사용하면 값을 하나 증가시켜서 다음 직원에 대해서는 새 번호를 받게 만든다.

3) `getEmployee()` 메서드는 범위 기반 `for` 문을 사용해서 `m_employees`에 있는 모든 직원에 대해 루프를 돌면서 메서드에 전달된 정보와 일치하는 `Employee`가 있는지 확인한다. 이 과정에서 일치하는 직원이 없으면 익셉션을 던진다.

## 3. DatabaseTest.cpp

### 1.2.4 사용자 인터페이스

사용자가 직원 데이터베이스를 쉽게 사용할 수 있도록 메뉴 기반으로 사용자 인터페이스를 구현한다.

- `main()` 함수는 화면에 메뉴를 출력하고, 여기서 선택한 동작을 수행하는 과정을 무한히 반복한다.
- 각 동작은 대부분 별도의 함수로 정의한다.
- 직원 정보를 화면에 출력하는 것과 같은 간단한 동작은 `case`에 직접 구현한다.

1. `displayMenu()` 함수는 메뉴를 화면에 출력하고, 사용자로부터 입력을 받는다.

사용자가 이상한 행동을 하지 않으며, 숫자를 요청하면 숫자를 입력한다고 가정했다.

2. `dohire()` 함수는 사요자로부터 신입 직원의 이름을 입력받아서 데이터베이스에 그 직원에 대한 정보를 추가하도록 요청한다.

3. `doFire()` 와 `doPromote()`는 둘 다 직원 번호를 이용하여 데이터베이스에서 그 직원에 대한 정보를 조회해서 `Employee` 객체의 공용 메서드를 사용하여 항목을 적절히 변경한다.