객체지향 시험 3

1. 첨부한 “inheritanceExample.zip” 들어 있는 프로그램과 유사한 문제가 출제될 예정이니 이 프로그램을 숙지하고 간단한 응용문제를 생각해보기 바람.
2. 퀴즈 3번과, 사진관리 프로그램, 퀴즈 1.2문제, inheritancecontainment.zip에 있는 프로그램을 참조해서 다음 문제를 풀어라. 각 단계별 결과를 볼 수 있는 실행코드가 주어진 파일 폴더에 들어 있다.

*제공된 코드에 수정할 부분이 표시되어 있다. 예를 들어서 A번의 ii 항목과 관련되는 부분은 “//?????[문제 3-2-x] “ 시작되고 구현할 내용에 대한 간단한 설명이 있다.*

“exam3-2-B” 폴더에 들어 있는 코드는 *A번과 B번에서 함께 사용할 수 있다. 다만 A번에서는 EventType class를 사용한지 않으니 이 파일만 include 시키지 않으면 된다.*

* 1. 제공된 소스폴더 “exam3-2-B”에는 이미 배포한 프로그램과 동일한(약간 보완) 실행 가능한 파일이 들어있다. 이 프로그램을 아래와 같이 변경하여 완성하라. *코드에서 A에 해당하는 부분은 “//?????[문제 3-2-A” 검색하면 된다. A번 문제를 완성한 후에도 샘플코드에 포함된 B번 문제 지시문(“//?????[문제 3-2-B”]를 지우지 않아야 한다.* 
     1. Template list class를 unsorted에서 sorted list로 바꾸어라. 이를 위해서 Add(), Delete(), Retrieve() 함수를 sorted list를 지원하도록 바꾼다. 이때 비교연산자를 overloading해서 써야 하는데 list에 저장할 record class(이 문제에서는 PhotoType)의 비교연산자가 overloading되어야 한다.
     2. Delete()함수의 exception 처리(list full일때와 레코드가 존재하지 않을 때)를 std:exception class를 base class로 사용하는 derived class를 정의하여 try, throw, catch를 이용하여 구현하라. Empty list인 경우 “List is Empty” 항목이 없을 때는 하나의 catch 문으로 "Key not found in the List"를 출력하도록 작성.
     3. GetNext()함수와 기능은 동일 하지만 다음 항목의 포인터를 리턴하는 T\* GetNextPtr()를 구현하라.
     4. Retrieve()와 동일한 기능을 하는 찾은 항목의 포인터를 리턴하는 RetrievePtr()를 구현
     5. Application class의 DisplayList()함수와 FindPhto()함수를 ii 와 iii에서 작성한 함수를 이용하여 다시 작성한다.
     6. \*\* A에서 작성한 코드는 별개의 폴더에 저장하여 제출한다.
  2. 문제 A에서 생성된 코드의 event list를 첨부하여 다음 기능을 수행한다.

*A번을 위해서 include에서 제외시켰던 “envent.h”를 복원하고 A번에서 작성하였던 코드에 “//?????[문제 3-2-B” 에서 요구한 내용을 추가한다.*

* + 1. 작업 1(PhotoFromKB())와 작업 6(ReadAllFromFile())에서 event list를 채울 수 있도록 다음 함수 구현하라
       1. Application::AddPhotoToList(item): 입력된 사진정보를 master list와 eventList에 추가
       2. EventType::AddPhotoKey() : event의 nameList에 새 사진이름(키)을 추가
       3. PhotoFromKB()와 ReadAllFromFile()에 위 수정사항 반영.
    2. 작업 8( event list를 이용하여 입력된 이벤트에 소속된 모든 사진정보 화면에 출력)을 위한 다음 함수를 구현한다.

Appication::DisplayByEventUsingEL(): 이벤트명을 받아서 event list에서 event를 찾고, 존재하면 DisplayDetail()를 호출하여 소속된 사진의 자세한 정보를 출력EventType::DisplayDetail() : master list를 전달받아서 해당 이벤트에서 찍은 모든 사진의 자세한 정보를 검색하여 화면에 출력.

* + 1. 작업 9(사진명과 이벤트 명을 AND로 사진을 검색)을 위한 다음 함수를 작성한다.

**Appication:: SearchByPhotoNameNEvent():**사진명과 event명을 받아서 eventList에서 항목을 찾은 다음에 EventType의 멤버함수(FindByNameKey)를 이벤트에 저장된 사진 중에서 일치하는 사진정보를 pointer로 받아서 화면에 출력.

**EventType::FindByNameKey()**: nameList에서 사진명을 찾아서 존재하면 전달받은 master list에서 해당 항목을 찾아서 포인터를 리턴한다.

* + 1. 작업 10(이벤트와 Contents내의 단어가 동시에 일치하는 사진 검색)을 위한 다음 함수를 구현한다.

**Application::SearchByEventNContents():** 이벤트명과 내용관련 키를 입력받아서 이벤트가 event list에 존재하면 EventType 멤버함수(FindByContentsKey)를 호출하여 이벤트에 소속된 사진 중에서 내용관련 키와 일치하는 첫번째 사진을 출력**FindByContentsKey(List<PhotoType> &master, string key)**: 현 이벤트에 소속된 사진 중에서 입력된 내용과 일치하는 첫번째 사진을 찾아서 그 사진 record의 포인터를 리턴. contents에서 일치하는 키를 차지 위해서는 PhotoType 함수(FindKeyFromContents)가 필요하다.

**PhotoType::FindKeyFromContents(string key):** 사진의 contents에 key가 포함되면 true를 그렇지 않으면 false를 리턴한다.

\*\*\* B번 문제까지 완성한 사람은 A번과 B번의 내용이 포함된 결과만 제출한다.

* 1. 문제 B의 class구조를 Application class와 멤버변수와 함수 중에서 공용으로 쓰일 수 있는 것들을 분리하여 Base class로 선언하고 EventType와 Application class를 derived class로 선언함으로써 derived class들이 master list를 직접 access할수 있게 바꾼다. *폴더 “exam3-2-C”에 있는 부분 code는 “exam3-2-B”의 code와 동일하다. “exam3-2-C”에는 C에서 구현할 부분이 표시되어 있다. 따라서 B에서 작성한 코드를 지시한대로 고치면 된다.*
     1. Application의 master list와 이와 관련되면서 여러 derived class에서 공통으로 사용할 수 있는 멤버함수들을 분리하여 base class를 작성한다.
     2. Based class를 상속받아서 Application class와 EventType class를 derived class를 구현한다. 필요한 멤버함수들도 함께 수정한다. \*\* base class의 static 변수는 사용하기 전에 메모리가 할당(선언)되어야 한다.
     3. 작업 8을 위해서 B에서 구현한 함수들을 derived class의 장점을 살릴 수 있도록 다시 구현한다.
     4. 작업 9과 10을 위해서 B에서 구현한 함수들을 derived class의 장점을 살릴 수 있도록 수정한다.

\*\*\* C 번의 답은 A와 B와 구분해서 제출한다.