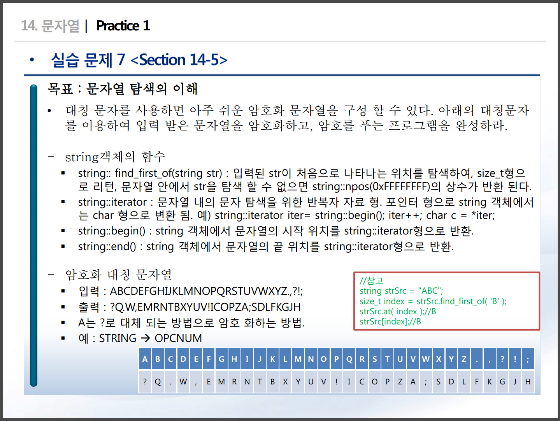
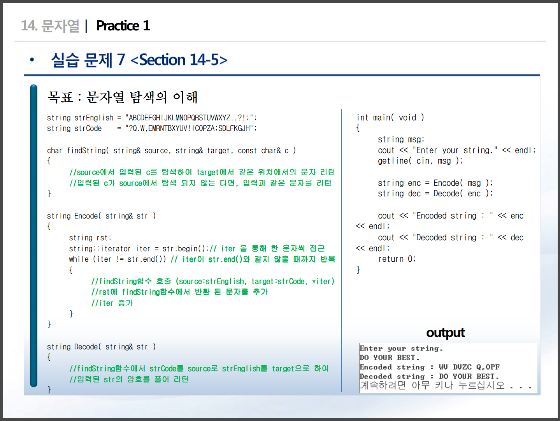
**객체지향 프로그래밍 2017년 봄학기 4차시험**

1. **애견샵에서 동물들을 관리하는 프로그램을 구현하려고 한다. 다음 조건을 참고하여 클래스를 구성하고 테스트 하시오. [20점]**
   1. 클래스는 전체 동물을 관리하는 petManager, 동물클래스인 Pet, Pet을 상속받고 개를 나타내는 Dog, Pet을 상속받고 고양이를 나타내는 Cat으로 구성된다.
   2. 클래스: Pet (Abstract 클래스)
      1. 변수: Pet의 종류(Dog와 Cat)에 대한 데이터를 저장할 변수 (열거형을 사용)
      2. 변수: 이름 (스트링 혹은 char 포인터)
      3. 변수: 연령, 성별, 체중, 건강상태, 크기를 저장할 변수
         1. 체중은 실수형(소수점 이하 1자리까지만 지정)
         2. 건강상태, 크기는 상, 중, 하로 구분해야 함(enum, string 중 선택하여 구현)
         3. 성별은 male, female로 구분 (enum, string 중 선택하여 구현)
      4. 함수: 모든 정보(데이터)를 출력하는 함수 pInfo( )
      5. 함수: 생성자, 소멸자, operator=( )
         1. Operator=()의 경우 객체의 타입이 같은 경우에만 동작해야 한다. 타입이 다른 객체가 인자로 사용될 경우 이것을 구분하여 적절한 메시지를 출력한 후 강제 종료 되도록 try, catch 문으로 예외처리를 한다.
      6. 함수: 분양 비용을 계산해 주는 함수. 아래의 기본 분양비용 방법을 적절한 위치에서 구현하라.
         1. Dog는 기본 분양 비용: 200000.
            1. 건강상태에 따라 기본 비용에 상은 30%, 중은 20%, 하는 10% 추가
            2. 크기에 따라 기본 비용에 상은 25%, 중은 20%, 하는 15% 추가
            3. 성별이 female일 경우 ii까지 계산한 비용의 15%가 추가
            4. Cat은 기본 분양 비용이 180000 이고 여타 비용 계산 방식은 Dog와 동일
      7. \*\*\*본 클래스에서 구현되지 못하는 함수들은 순수가상함수로 선언하고, 이를 아래에서 구현함.
   3. 클래스: Dog
      1. 필요한 함수 구현
   4. 클래스: Cat
      1. 필요한 함수 구현
   5. 클래스: petManager
      1. 이 클래스는 이 샵에서 분양 예정인 동물들의 리스트를 관리한다.
      2. 변수: Pet \* 형식 혹은 STL의 vector<Pet> 형식. 관리 동물들의 리스트를 저장한다. 이 변수 하나로 Dog 과 Cat 을 모두 관리.
      3. 함수: Pet 을 매개변수로 받아 이를 위 리스트에 추가해 주는 함수
      4. 함수: 모든 동물의 분양 비용 합을 계산해 주는 함수. 이 함수에서는 각각의 동물의 분양 비용 계산 함수를 돌아가며 호출하여 총 합을 계산함
      5. 함수: 모든 동물들의 정보를 출력하는 함수.
   6. 필요한 경우 추가로 함수를 작성한다.
   7. Main 에서는 Dog, Cat 을 최소 5마리 이상 생성하여 petManager에 추가하고, petManger를 이용하여 총 비용계산, 정보 출력 등을 수행하라.
   8. 그리고, operator= 도 적절히 테스트하라.
2. **중세 전투 게임에 사용할 클래스들을 잘 정의하여 아래 main이 오른쪽과 같이 출력 결과를 낼 수 있도록 구현하시오. [20점]**
   1. 클래스 Unit은 abstract class로 유닛을 표현하는데 기본적으로 필요한 체력과 이름, 공격 수단에 대한 데이터와 이를 처리하기 위한 함수들을 가진다. 그 중 일부는 순수 가상함수로 하여 자식에서 구현하도록 한다. 구체적으로 다음과 같은 멤버를 가진다.
      1. Protected 멤버
         1. 정수형 현재 체력을 저장할 변수;
         2. 공격 시 출력할 문자열을 저장할 변수;
         3. 이름을 저장할 변수
      2. Public 멤버
         1. 적절한 생성자 및 소멸자
         2. HP에 대한 수정 및 반환 함수.
      3. Public에 속하는 순수 가상 함수
         1. Unit \*를 매개변수로 받는 공격 함수
         2. 이름을 반환하는 함수
         3. 자신의 정보를 출력하는 함수
   2. 클래스 SwordMan 과 Mage는 거의 유사한 구조를 가지고 Unit을 public으로 상속 받아 순수 가상함수들을 정의한다.
      1. 초기화 할 때 공격 문장을 각각 칼과 마법으로 구분하고, 초기 체력을 소드맨은 200, 메이지는 50으로 한다.
      2. 이름을 반환하는 함수는 이름 앞에 각각 자기 직업을 붙여서 반환한다.
      3. 정보 출력 함수도 각각의 직업 부분을 다르게 하여 출력한다.
      4. 공격 함수 사용 시 소드맨은 전달 받은 객체에 5점의 피해를 주고, 메이지는 10점의 피해를 준다.
   3. 상속 구조인 유닛 표현 클래스의 부모 포인터를 이용하여 게임 안의 한 케릭터를 객체로 만드는데 실제로 사용할 Character 클래스를 구현하라. Character는 template을 사용하며 멤버는 다음과 같다.
      1. Protected 멤버
         1. generic type의 포인터 타입인 데이터를 저장할 변수;
      2. public 멤버
         1. 데이터 입력 및 반환 함수
         2. 적절한 생성자 및 소멸자
         3. battle() generic type의 포인터을 매개변수로 전달 받아 데이터와 매개변수 서로에 대해 공격 함수를 호출하는 함수.
         4. printInfo() 데이터의 정보 출력 함수를 호출하는 함수.

|  |  |
| --- | --- |
| main.cpp | **출력 결과** |
| #include "Character.h"  #include "SwordMan.h"  #include "Mage.h"  int main()  {  Character <Unit> u1, u2;  u1.setData(new SwordMan("Sam"));  u2.setData(new Mage("Mike"));  u1.printInfo();  u2.printInfo();  u1.battle(u2.getData());  u1.printInfo();  u2.printInfo();  return 0;  } | 이름 Sam  직업 소드맨  HP 200  이름 Mike  직업 메이지  HP 50  소드맨 Sam이 메이지 Mike에게 칼로 공격!  5점 피해!  메이지 Mike이 소드맨 Sam에게 마법으로 공격!  10점 피해!  이름 Sam  직업 소드맨  HP 190  이름 Mike  직업 메이지  HP 45 |

1. **위의 Character를 상속 받은 AceCharacter를 잘 정의하여 아래 main이 오른쪽과 같이 출력 결과를 낼 수 있도록 구현하시오. [20점]**
   1. AceCharacter는 기존의 Character에 파일로부터 입력 받아 레벨업을 하는 기능을 가진 클래스이다. 다음과 같은 멤버를 가진다.
      1. static 함수 buildData() 이 함수에서는 “data.bin”파일에 binary.로 0부터 90까지 10씩 증가하는 int 데이터를 저장한다.
      2. levelUp() 함수 이 함수에서는 “data.bin”의 시작 위치부터 int를 기준으로 1~10번째 랜덤한 위치로 이동하여 데이터를 읽어 데이터의 HP를 증가시킨다.
   2. Unit, SwordMan, Mage 클래스에 Unit\*를 전달 받아 같은 직업인지 여부를 bool로 반환하는 함수를 각각 구현한다.
   3. 위 함수를 이용하여 AceCharacter에서 battle()함수를 오버라이딩 하여 입력 받은 매개변수가 같은 직업일 경우 전투를 하지 않도록 한다.
   4. AceCharacter에 generic type 포인터 타입에 대한 형변환 연산자를 구현한다. 이 함수는 데이터 반환 함수의 호출 결과를 반환한다.

|  |  |
| --- | --- |
| main.cpp | **출력 결과** |
| #include "AceCharacter.h"  #include "SwordMan.h"  #include "Mage.h"  int main()  {  AceCharacter <Unit>::buildData();  AceCharacter <Unit> u1, u2, u3;  u1.setData(new SwordMan("Sam"));  u2.setData(new Mage("Mike"));  u3.setData(new Mage("John"));  u1.levelUp();  u1.battle(u2);  u1.printInfo();  u2.printInfo();  u2.battle(u3);  return 0;  } | 소드맨 Sam이 메이지 Mike에게 칼로 공격!  5점 피해!  메이지 Mike이 소드맨 Sam에게 마법으로 공격!  10점 피해!  이름 Sam  직업 소드맨  HP 210  이름 Mike  직업 메이지  HP 45 |

1. **암호화 프로그램. [20점]**
   1. String 관련 실습시간에 치환 테이블을 이용한 암호화 방법을 구현하였다. 다음 항목들을 추가로 구현하여 이 프로그램을 확장하라.
   2. 문자열을 사용자에게 받는 대신에 텍스트 파일을 읽어 파일내의 텍스트를 모두 암호화 할 수 있게 만들어라. (main에서는 파일 이름을 사용자에게 입력 받아 이 파일의 내용을 암호화)
   3. 텍스트 파일에는 실습 문제와는 달리 공백문자와 리턴 문자가 포함될 수 있다. 암호화 시 이는 치환을 하지 않고 그대로 두는 구문을 추가하라.
   4. 암호화된 문자열을 모두encrypted.txt에 파일 출력하라 (텍스트 출력).
   5. 그 다음, encrypted.txt 파일을 읽어, decoding 함수로 암호를 풀어 원본 문자열을 복구하라. 그리고 이를 화면에 출력하라.
   6. 추가로, 복구된 원본 문자열을 줄 별로 (리턴 문자를 기준으로)로 분리하고, 줄 수만큼의 크기를 가지는 string 배열을 만들어 각각 저장하라.
   7. 저장된 스트링 배열을 알파벳 순으로 정렬하고, 이를 화면에 출력하라.
   8. 추가로, 만약 원본 문자열에 소문자가 존재한다면 이를 대문자로 치환하여 암호화 하는 과정을 예외처리로 추가하라. (힌트: findString함수에서 try, catch 구문 사용). 암호를 푸는 과정에서 따로 소문자로 복원할 필요 없음.
2. **KHUKMIN 은행에서 새로운 은행계좌 및 고객 관리 시스템을 개발하려고 한다. 이에 대한 클래스 디자인을 수행하라. [20점]**
   1. 컴파일/실행되는 프로그램을 만들 필요는 없고, 클래스 껍데기 코드만 (헤더파일) 구현.
   2. 고객, 계좌, 은행직원, 사내 게시판, 온라인 어카운트, 은행 앱, Transaction, 은행 지점, 지점당 만족도, 등등의 관련 클래스들에 대해, 이 클래스들이 어떻게 연결되어 있어야 하고 (상속, 집합, 함수호출 등) 어떤 함수 및 데이터 들이 필요한 지에 대한 디자인임. (위 클래스 외에 자유롭게 추가 가능, 예, 고객은 여자고객, 남자고객을 자식으로, 계좌는 펀드계좌, 일반계좌, 주식계좌 등을 자식으로, 직원 인사관리 클래스를 추가하기 등등 확장 가능)
   3. 생각할 수 있는 모든 기능을 다 제공하게 아주 넓고 좀 추상적으로 디자인을 수행해도 되고, 특정 기능을 아주 디테일하게 디자인 해도 됨. 후자에서는 시스템이 제공하는 기능에 대한 정의를 명확하게 해야 함.
   4. 각각의 디자인의 타당성에 대한 설명을 코멘트로 추가하여야 함.