

2016 C++ 프로그래밍 실습과제 5

※ 기존 과제들에 대한 클래스 구현.

5.1 실습과제 4에서 구현하던 복소수 구조체와 관련 함수를 클래스로 다시 구현하시오.

- (1) UML 클래스 다이어그램을 그리시오.
 - 클래스이름, 데이터 멤버, 멤버함수 표시
 - 접근 권한 표시 (+, -, #)
- (2) 복소수 클래스를 3가지 방법으로 구현하고 테스트하시오.
 - 구조체 형식을 클래스 형태로만 변경한 클래스: **ComplexV2** (복소수 v.2)
 - ☞ **ComplexV2.h, ComplexV2.cpp**
 - 멤버 함수의 이름을 단순화한 클래스: **ComplexV3** (복소수 v.3)
 - ☞ **ComplexV3.h, ComplexV3.cpp**
 - 모든 멤버 함수를 **inline** 함수로 구현한 클래스: **ComplexV4** (복소수 v.4)
 - ☞ **ComplexV4.h**
 - 강의시간에 설명한 코드를 참고하여 구현할 것
 - **main()** 함수에서 구현한 클래스들을 한꺼번에 테스트할 것

5.2 실습과제 4에서 구현하던 이차방정식 관련 함수를 클래스로 다시 구현하시오.

- (1) UML 클래스 다이어그램을 그리시오.
 - 클래스 이름
 - ☞ 예) **MyQuadratic**
 - 데이터 멤버
 - ☞ 다항식의 계수: **a, b, c** // $f(x) = ax^2 + bx + c$
 - 멤버 함수
 - ☞ **read()**: 이차방정식을 읽어들이는 함수
 - ☞ **print()**: 이차방정식을 출력하는 함수
 - ☞ **add()**: 두 개의 이차 방정식을 더하는 함수
 - ☞ **solve()**: 이차방정식을 풀어 해를 구하는 함수
- (2) UML 클래스 다이어그램으로 설계한 이차방정식 클래스를 구현하고 테스트하시오.
 - 위에서 구현한 복소수 클래스를 사용하여 구현함.
 - **Complex** 클래스의 여러 가지 구현 방법 중에서 하나의 방식으로 구현하면 됨.
 - ☞ **MyQuadFn.h, [MyQuadFn.cpp]**
 - **read, print, add**는 복소수에서와 같은 의미임
 - ☞ 기존 과제에서 이차방정식의 덧셈 함수 **add()** 추가
 - ☞ **Complex** 클래스에서 각 함수의 원형을 참고하여 설계하고 구현할 것 (매개

변수와 반환형)

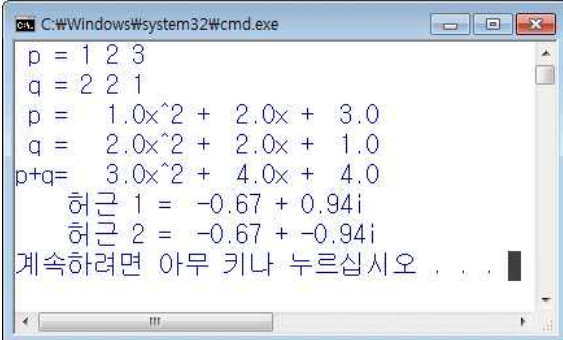
- 이차 방정식의 근을 구하는 `solve()`
- 이차방정식의 근을 출력하는 `printSolution()` 함수는 일반 함수로 처리하라.
(`inline`으로 처리)

☞ 왜 일반함수로 처리를 할까?

☞ `printSolution()`을 멤버함수로 처리할 수 있는 방법은?

(3) 다음과 같은 테스트 프로그램을 작성하고 동작을 테스트하시오.

```
void main() {  
    QuadFn p, q, r;  
    p.read( " p = " );  
    q.read( " q = " );  
    r.add( p, q );  
  
    p.print( " p = " );  
    q.print( " q = " );  
    r.print( "p+q= " );  
  
    int n;  
    Complex r1, r2;  
    n = r.solve( r1, r2 );  
    printResult( n, r1, r2);  
}
```



```
C:\Windows\system32\cmd.exe  
p = 1 2 3  
q = 2 2 1  
p = 1.0x^2 + 2.0x + 3.0  
q = 2.0x^2 + 2.0x + 1.0  
p+q= 3.0x^2 + 4.0x + 4.0  
    허근 1 = -0.67 + 0.94i  
    허근 2 = -0.67 + -0.94i  
계속하려면 아무 키나 누르십시오 . . .
```

5.3 실습과제 2와 3에서 구현하였던 구구단 게임과 랭킹 관련 코드도 이와 같은 절차로 클래스를 만들고 테스트해보시오. (선택)