

컴퓨터 프로그래밍 2
-hw02-

학번 : 201602038
이름 : 이 미 진

1. 함수 설명

1-1 AppIO

- 1) Boolean AppIO_in_solvingIsRequested() : 이차방정식 풀이를 할 것인지 종료할 것인지 묻는 메시지 출력
- 2) void AppIO_in_quadEquation(float* p_c0, float* p_c1, float* p_c2) : 계수를 각각 입력받아서 돌려주는 함수
- 3) void AppIO_out_msg_startSolvingQuadEquation() : <<<이차방정식 풀이>>> 메시지 출력
- 4) void AppIO_out_msg_endSolvingQuadEquation() : <<<이차방정식 풀이를 종료합니다>>> 메시지 출력
- 5) void AppIO_out_msg_secondOrderTermCoefficientIsZero() : 이차항의 계수가 0임을 알리는 메시지 출력
- 6) void AppIO_out_msg_determinantIsNegative(float aDeterminant) : 판별식의 값이 음수임을 알리는 메시지를 출력하고 판별식의 값을 보여준다.
- 7) void AppIO_out_quadEquation (float c0, float c1, float c2) : 이차방정식을 보여줌 단, 계수의 값이 0인 항은 보이지 않게 하고 모든 계수가 0이면 0=0 을 출력
- 8) void AppIO_out_solution (float root1, float root2) : 방정식의 해를 보여주는 함수

1-2 QuadEquationProblem

- 1) Boolean QuadEquationProblem_secondOrderTermCoefficientIsZero (QuadEquationProblem aProblem) : 이차항이 0인지 아닌지 상태를 보여주는 함수
- 2) Boolean QuadEquationProblem_determinantIsNegative (QuadEquationProblem aProblem) : 판별식이 0인지 아닌지 상태를 보여주는 함수
- 3) float QuadEquationProblem_determinant (QuadEquationProblem aProblem) : 판별식의 값을 얻는 함수
- 4) Solution QuadEquationProblem_solve (QuadEquationProblem aProblem) : 방정식의 해를 구한다. 결과는 해(Solution) 객체로 만들어 돌려준다.
- 5) Solution QuadEquationProblem_solution (QuadEquationProblem aProblem) : 소유하고 있는 정보로부터 객체의 상태를 얻는다.

2. 전체 코드

1) main.c

```
//
//  main.c
//
//
//  Created by stu2017s10 on 2017. 3. 14..
//
//

#include <stdio.h>
#include <math.h>

#include "AppIO.h"
#include "Common.h"
#include "QuadEquationProblem.h"

int main() {
    QuadEquationProblem    qeProblem; // 객체 qeProblem 생성
    Boolean                solvingIsRequested; // 객체 solvingIsRequested 생성
    QuadEquation            equation; // 객체 equation 생성
    Solution                solution; // 객체 solution 생성
    //객체 생성 : 문제 풀이 과정에서 필요한 객체의 속성 정보를 소유한다.

    AppIO_out_msg_startSolvingQuadEquation(); // 이차방정식 풀이 시작
    메시지 출력
    solvingIsRequested = AppIO_in_solvingIsRequested(); // 이차방정식
    풀이 여부를 묻고 사용자로부터 응답을 입력받는다.

    while ( solvingIsRequested ) {
        AppIO_in_quadEquation (&(equation._c0), &(equation._c1),
        &(equation._c2)); // 이차방정식의 계수를 입력받는다.
        AppIO_out_quadEquation (equation._c0, equation._c1,
        equation._c2); // 이차방정식을 보여준다.

        //QuadEquationProblem_ 을 붙인 함수는 QuadEquationProblem 객체
        와 직결된 함수
        qeProblem._equation = equation; // main()에서 객체의 속성값을 직
        접 변경

        if( QuadEquationProblem_secondOrderTermCoefficientIsZero(qeProblem
        ) ) { // 이차항의 계수가 0 이면
            AppIO_out_msg_secondOrderTermCoefficientIsZero(); //
            이차항의 계수가 0이어서 이차방정식이 아니라는 메시지 출력
        }
        else {
```

```

if( QuadEquationProblem_determinantIsNegative(qeProblem) ) {    //
판별식의 값이 음수이면
    float determinant =
QuadEquationProblem_determinant (qeProblem) ;    //판별식의 값과 판별식의
값이 음수라는 메시지를 출력
        AppIO_out_msg_determinantIsNegative(determinant);
    }
    else {
        qeProblem._solution =
QuadEquationProblem_solve(qeProblem); // main()에서 객체의 속성값을 직접
변경 , 이차방정식을 푼다.
        solution = QuadEquationProblem_solution(qeProblem);
        AppIO_out_solution (solution._root1, solution._root2);
// 이차방정식의 해를 출력
    }
}
solvingIsRequested = AppIO_in_solvingIsRequested(); // 이차
방정식 풀이 여부를 묻고 사용자로부터 응답을 입력받는다.
}
AppIO_out_msg_endSolvingQuadEquation(); // 이차방정식 풀이 종료 메
시지 출력
return 0;
}

```

2) Common.h

```

//
// Common.h
//
//
// Created by stu2017s10 on 2017. 3. 14..
//
//

#ifndef Common_h
#define Common_h

#include <math.h>

#define EPSILON 0.000001 // 상수에 이름을 부여하기 위한 매크로 선언
#define FloatValueIsZero(NUMBER) fabsf(NUMBER) < EPSILON // 반
복 사용되는 코드를 위한 매크로 선언

typedef enum {FALSE, TRUE} Boolean; // Boolean 타입을 define 해주고
FALSE, TRUE 값을 가짐. FALSE와 TRUE의 순서는 바뀌면 안됨.

#endif /* Common_h */

```

3) AppI0.c

```
//
// AppI0.c
//
//
// Created by stu2017s10 on 2017. 3. 14..
//
//

#include "AppI0.h"
#include "Common.h"
#include "QuadEquationProblem.h"

//입력 관련 함수
Boolean AppI0_in_solvingIsRequested() { // 이차방정식 풀이를 할 것인지 종
료할 것인지 묻는 메시지 출력
    char answer;
    printf("방정식을 풀려면 y, 풀이를 종료하려면 아무 키나 입력하십시오.");
    fpurge(stdin); // 이전에 입력되어 임시 보관중인 모든 입력 값을 없앤다.
    answer = getchar(); // 키보드로부터 키 한개를 입력받는다.
    return ( answer == 'y'); // 입력 받은 것이 y이면 TURE,그 외의 값은
FALSE 반환
}

void AppI0_in_quadEquation( float* p_c0, float* p_c1, float*
p_c2 ) { // 계수를 각각 입력받아서 돌려준다.

    printf("2차항의 계수를 입력하십시오 : ");
    scanf("%f" ,p_c2); // 2차항의 계수

    printf("1차항의 계수를 입력하십시오 : ");
    scanf("%f" ,p_c1); // 1차항의 계수

    printf("상수항의 계수를 입력하십시오 : ");
    scanf("%f" ,p_c0); // 상수항의 계수
}

//출력 관련 함수
void AppI0_out_msg_startSolvingQuadEquation() { // <<<이차방정식 풀이
>>> 메시지 출력
    printf("<<<이차방정식 풀이>>>\n");
}

void AppI0_out_msg_endSolvingQuadEquation() { // <<<이차방정식 풀이를
종료합니다>>> 메시지 출력
    printf("<<<이차방정식 풀이를 종료합니다>>>\n");
}
```

```

void AppIO_out_msg_secondOrderTermCoefficientIsZero() { // 이차항의
계수가 0임을 알리는 메시지 출력
    printf(">이차항의 계수가 0이어서, 이차방정식이 아닙니다.\n");
}
void AppIO_out_msg_determinantIsNegative(float aDeterminant) { //
판별식의 값이 음수임을 알리는 메시지를 출력하고 판별식의 값을 보여준다.
    printf(">판별식의 값이 음수여서 해가 존재하지 않습니다.\n");
    printf("-판별식의 값 : %0.1f \n ",aDeterminant);
}

void AppIO_out_quadEquation (float c0, float c1, float c2) { //
이차방정식을 보여줌 단, 계수의 값이 0인 항은 보이지 않게 하고 모든 계수가 0이면 0=0
을 출력
    Boolean aNonZeroTermExists = FALSE;
    printf(">주어진 방정식 : ");
    //매크로 사용 , 0인지 아닌지 검사
    if( ! (FloatValueIsZero(c2)) ) { // 2차항의 계수가 0이 아니라면
        aNonZeroTermExists = TRUE;
        printf("(%0.1f)x * x" , c2);
    }
    if( ! (FloatValueIsZero(c1)) ) { // 1차항의 계수가 0이 아니라면
        if( aNonZeroTermExists )
            printf(" + ");
        aNonZeroTermExists = TRUE;
        printf("(%0.1f) x" ,c1);
    }
    if( ! (FloatValueIsZero(c0)) ) { // 상수항의 계수가 0이 아니라면
        if( aNonZeroTermExists )
            printf(" + ");
        aNonZeroTermExists = TRUE;
        printf("(%0.1f)" ,c0);
    }
    if( !aNonZeroTermExists ) { // 모든 계수가 0이면
        printf("0");
    }
    printf(" = 0\n");
}

void AppIO_out_solution (float root1, float root2) { // 방정식의
해를 보여준다.
    printf(">방정식의 해는 다음과 같습니다.\n");
    printf("x1 = %0.2f \n" ,root1);
    printf("x2 = %0.2f \n" ,root2);
}

```

4) AppIO.h

```
//
// AppIO.h
//
// Created by stu2017s10 on 2017. 3. 14..
//
//

#ifndef AppIO_h
#define AppIO_h

#include "Common.h"
#include <stdio.h>

//입력 관련 함수
Boolean AppIO_in_solvingIsRequested(void); // 이차방정식 풀이를 할 것인지
종료할 것인지 묻는 메시지 출력
void AppIO_in_quadEquation( float* p_c0, float* p_c1, float*
p_c2 ); // 계수를 각각 입력받아서 돌려준다.

//출력 관련 함수
void AppIO_out_msg_startSolvingQuadEquation(void); // <<<이차방정식
풀이>>> 메시지 출력
void AppIO_out_msg_endSolvingQuadEquation(void); // <<<이차방정식 풀
이를 종료합니다>>> 메시지 출력
void AppIO_out_msg_secondOrderTermCoefficientIsZero(void); // 이차항
의 계수가 0임을 알리는 메시지 출력
void AppIO_out_msg_determinantIsNegative(float aDeterminant); //
판별식의 값이 음수임을 알리는 메시지를 출력하고 판별식의 값을 보여준다.
void AppIO_out_quadEquation (float c0, float c1, float c2); //
이차방정식을 보여줌 단, 계수의 값이 0인 항은 보이지 않게 하고 모든 계수가 0이면 0=0
을 출력
void AppIO_out_solution (float root1, float root2); // 방정식의 해
를 보여준다.

#endif /* AppIO_h */
```

5) QuadEquationProblem.c

```
//
// QuadEquationProblem.c
//
// Created by stu2017s10 on 2017. 3. 14..
//
```

```

//

#include "QuadEquationProblem.h"
#include "AppIO.h"

// 이차방정식 관련 함수

// 객체의 상태를 보여주는 함수 ( 객체의 속성 정보 값이 바뀌지 않음)
Boolean QuadEquationProblem_secondOrderTermCoefficientIsZero
(QuadEquationProblem aProblem) {
    return FloatValueIsZero (aProblem._equation._c2);
}

Boolean QuadEquationProblem_determinantIsNegative
(QuadEquationProblem aProblem) { // 판별식이 0인지 아닌지 상태를 보여주는 함수
    if ( ( (aProblem._equation._c1 *
aProblem._equation._c1)-4*aProblem._equation._c2*aProblem._equation._c0 ) < 0) {
        return TRUE;
    }
    else return FALSE;
}

float QuadEquationProblem_determinant (QuadEquationProblem
aProblem) { // 판별식의 값을 얻는 함수
    // 계산을 하여 객체의 상태를 얻는다.
    float determinant = (aProblem._equation._c1 *
aProblem._equation._c1)-4*aProblem._equation._c2*aProblem._equation._c0; // 판별식의 값 계산, 판별식의 값 b*b-4*a*c
    return determinant;
}

//객체의 상태를 바꾸는 함수 ( 객체의 속성 정보 값이 바뀜 )
Solution QuadEquationProblem_solve (QuadEquationProblem aProblem)
{ // 방정식의 해를 구한다. 결과는 해(Solution) 객체로 만들어 돌려준다.
    Solution solution;

    float determinant = QuadEquationProblem_determinant
(aProblem);
    //객체에게 일을 시키는 쪽 ( main()에서만 사용하는 함수지만 이렇게 객체 자신도
    필요하면 사용!)
    float sqrtDeterminant = sqrt(determinant); // <math.h> 반드시
include
    solution._root1 = (-aProblem._equation._c1 +
sqrtDeterminant) / (2.0 * aProblem._equation._c2);
    solution._root2 = (-aProblem._equation._c1 -
sqrtDeterminant) / (2.0 * aProblem._equation._c2);
}

```



```

        return solution;
    }

    Solution QuadEquationProblem_solution (QuadEquationProblem
aProblem) {
    // 소유하고 있는 정보로부터 객체의 상태를 얻는다.
    return aProblem._solution;
}

```

6) QuadEquationProblem.h

```

//
//  QuadEquationProblem.h
//
//
//  Created by stu2017s10 on 2017. 3. 14..
//
//

#ifndef QuadEquationProblem_h
#define QuadEquationProblem_h

#include "Common.h"

typedef struct {
    float _c0; // 상수항의 계수
    float _c1; // 1차항의 계수
    float _c2; // 2차항의 계수
} QuadEquation;

typedef struct {
    float _root1; // 방정식의 해 1
    float _root2; // 방정식의 해 2
} Solution;

typedef struct {
    QuadEquation _equation;
    Solution _solution;
} QuadEquationProblem; // 이차방정식 문제

Boolean QuadEquationProblem_secondOrderTermCoefficinetIsZero
(QuadEquationProblem aProblem);
Boolean QuadEquationProblem_determinantIsNegative
(QuadEquationProblem aProblem);
Solution QuadEquationProblem_solve (QuadEquationProblem aProblem);

```

```

Solution QuadEquationProblem_solution (QuadEquationProblem
aProblem);
float QuadEquationProblem_determinant (QuadEquationProblem
aProblem);
void QuadEquationProblem_setEquation (QuadEquationProblem,
QuadEquationProblem equation);

#endif /* QuadEquationProblem_h */

```

3. 종합 설명

1. QuadEquationProblem의 객체 qeProblem 생성
2. Boolean 타입의 객체 solvingIsRequested 생성
3. QuadEquation의 객체 equation 생성
4. Solution의 객체 solution 생성
5. AppIO_out_msg_startSolvingQuadEquation() 함수로 이차방정식 풀이 시작 메시지를 출력한다.
6. solvingIsRequested = AppIO_in_solvingIsRequested() 함수로 이차방정식 풀이 여부를 묻고 사용자로부터 응답을 받는다.
7. 입력받은 값이 y일 경우 while문으로 이차방정식 풀이를 반복한다.
8. AppIO_in_quadEquation (&(equation._c0), &(equation._c1), &(equation._c2)) 함수로 이차방정식의 이차항, 일차항, 상수항의 계수를 입력받는다.
9. AppIO_out_quadEquation (equation._c0, equation._c1, equation._c2) 함수로 이차방정식을 보여준다.
10. QuadEquationProblem_secondOrderTermCoefficientIsZero(qeProblem) 함수로 입력받은 이차항의 계수가 0이면 AppIO_out_msg_secondOrderTermCoefficientIsZero() 로 이차항의 계수가 0이어서 이차방정식이 아니라는 메시지를 출력한다.
11. 입력받은 이차항의 계수가 0이 아니고 QuadEquationProblem_determinantIsNegative(qeProblem) 함수로 판별식의 값이 음수이면 판별식의 값이 음수라는 메시지와 판별식의 값을 출력한다.
12. 이차항의 계수가 0이 아니고 판별식의 값이 음수가 아닐 때, qeProblem._solution = QuadEquationProblem_solve(qeProblem) 함수로 이차방정식을 풀고, AppIO_out_solution (solution._root1, solution._root2) 으로 이차방정식의 해를 출력한다.
13. 이차방정식 풀이가 끝나면 다시 solvingIsRequested = AppIO_in_solvingIsRequested() 으로 다시 이차방정식 풀이 여부를 묻고 사용자로부터 응답을 입력받는다.
14. y가 아닌 값이 입력되면 AppIO_out_msg_endSolvingQuadEquation()으로 이차방정식 풀이 종료 메시지를 출력하고 반복을 종료한다.

4. 실행 결과

```
<<<이차방정식 풀이>>>
방정식을 풀려면 y, 풀이를 종료하려면 아무 키나 입력하시오.y
2차항의 계수를 입력하시오 : 1.0
1차항의 계수를 입력하시오 : 1.0
상수항의 계수를 입력하시오 : 1.0
>주어진 방정식 :  $(1.0)x * x + (1.0)x + (1.0) = 0$ 
>판별식의 값이 음수여서 해가 존재하지 않습니다.
-판별식의 값 : -3.0
방정식을 풀려면 y, 풀이를 종료하려면 아무 키나 입력하시오.y
2차항의 계수를 입력하시오 : 0.0
1차항의 계수를 입력하시오 : 0.0
상수항의 계수를 입력하시오 : 0.0
>주어진 방정식 :  $0 = 0$ 
이차항의 계수가 0이어서, 이차방정식이 아닙니다.
방정식을 풀려면 y, 풀이를 종료하려면 아무 키나 입력하시오.y
2차항의 계수를 입력하시오 : 4.0
1차항의 계수를 입력하시오 : 2.0
상수항의 계수를 입력하시오 : 1.0
>주어진 방정식 :  $(4.0)x * x + (2.0)x + (1.0) = 0$ 
>판별식의 값이 음수여서 해가 존재하지 않습니다.
-판별식의 값 : -12.0
방정식을 풀려면 y, 풀이를 종료하려면 아무 키나 입력하시오.y
2차항의 계수를 입력하시오 : 4.0
1차항의 계수를 입력하시오 : 2.0
상수항의 계수를 입력하시오 : 0.0
>주어진 방정식 :  $(4.0)x * x + (2.0)x = 0$ 
>방정식의 해는 다음과 같습니다.
x1 = 0.00
x2 = -0.50
방정식을 풀려면 y, 풀이를 종료하려면 아무 키나 입력하시오.n
<<<이차방정식 풀이를 종료합니다>>>
Program ended with exit code: 0
```