

```

if (sm.GetNiji(3, 10, 7, &x1, &x2) == true)
{
    cout << "x1 = " << x1 << "\n";
    cout << "x2 = " << x2 << "\n";
}
else
{
    cout << "해가 없다.\n";
}

return 0;
}

```

```

x1 = -1
x2 = -2.33333

```

List 4-2의 실행 결과

2차 방정식 해의 공식은 중학교에서 배운 것들입니다. 아주 오래 전에 배운 것이라 기억이 희미하실 분들을 위해 공식을 옮겨 놓았습니다. 이 공식의 루트 안의 계산이 SimpleMath 클래스의 멤버 함수 TempCalc()로 이루어져 있습니다. TempCalc()의 반환값이 음수라면 허수 해가 됩니다.

2차 방정식의 일반형

$$ax^2 + bx + c = 0$$

해의 공식

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

그림 4-7

2차 방정식의 해 공식

클래스를 정의할 때 멤버를 캡슐화할 것인가, 아닌가에 대해 결정해야 합니다. 어떤 이유로 캡슐화를 실행하는가의 문제는 객체 지향 프로그래밍을 어떻게 인식하고 있는가에 의해 정해집니다. '캡슐화 따위는 귀찮다!'고 생각한다면 모든 멤버를 public:으로 공개해 주십시오. 그렇다고 프로그램이 움직이지 않는 일은 없을 것입니다. 여하튼 '모든 멤버가 public:로는 곤란하다!'라는 문제에 부딪치게 된다면 당연히 자연스럽게 캡슐화를 활용할 수 있게 됩니다.