**package** algorithms;

// CS 225A, 9-2-08

/\* Lab 1 Assignment: Use Eclipse to create and run the Quadratic class

\* given below. A Quadratic is a quadratic expression in a single variable

\* with real (double precision) coefficients. The basic Quadratic class

\* has a constructor, a toString method, and a method "roots" for finding

\* the real roots of the quadratic expression (if any).

\*/

**public** **class** Quadratic {

// Data fields

// A Quadratic object "knows" its three coefficients.

**private** **double** a, b, c;

// Constructor

// Input: The desired coefficients d, e, f, with d != 0

// Result: Creates the Quadratic object corresponding to d\*x^2 + e\*x + f

**public** Quadratic(**double** d, **double** e, **double** f)

{

a = d;

b = e;

c = f;

}

// Converts this Quadratic to its String representation

// Input: void

// Returns: "ax^2 + bx + c"

**public** String toString()

{

**return** a + "x^2 + " + b + "x + " + c;

}

// Finds the real roots of this Quadratic

// Input: void

// Returns: a 1-by-2 array containing the real roots of this Quadratic function

// Note: If there are no real roots, both entries of the returned array will

// have the value Float.Nan (not a number)

**public** **double**[] roots()

{

**double**[] r = **new** **double**[2];

**double** d = b\*b - 4\*a\*c; // discriminant

**if**(d<0)

r[0] = r[1] = Float.*NaN*;

**else**

{

r[0] = (-b-Math.*sqrt*(d)/(2\*a));

r[1] = (-b+Math.*sqrt*(d)/(2\*a));

}

**return** r;

}

// Main method to exercise the Quadratic class

**public** **static** **void** main(String[] args) {

Quadratic f = **new** Quadratic(1.0, 0.0, 5.0);

Quadratic g = **new** Quadratic(0.005, 10.8, 5.2);

System.*out*.println("f = " + f.toString());

System.*out*.println("g = " + g.toString());

**double**[] rootsF = f.roots();

**double**[] rootsG = g.roots();

System.*out*.print("The real roots of " + f.toString() + " are ");

System.*out*.println(rootsF[0] + " and " + rootsF[1]);

System.*out*.print("The real roots of " + g.toString() + " are ");

System.*out*.println(rootsG[0] + " and " + rootsG[1]);

}

}