

Solutions for Exercises Unit 1

This is a non-graded exercise which should be posted to your learning journal.

Write a program that uses the following subroutine, from Subsection 8.3.3 of the textbook, to solve equations specified by the user.

```
/**
 * Returns the larger of the two roots of the quadratic equation
 *  $Ax^2 + Bx + C = 0$ , provided it has any roots. If  $A == 0$  or
 * if the discriminant,  $B^2 - 4AC$ , is negative, then an exception
 * of type IllegalArgumentException is thrown.
 */
static public double root( double A, double B, double C )
    throws IllegalArgumentException {
    if (A == 0) {
        throw new IllegalArgumentException("A can't be zero.");
    }
    else {
        double disc = B*B - 4*A*C;
        if (disc < 0)
            throw new IllegalArgumentException("Discriminant < zero.");
        return (-B + Math.sqrt(disc)) / (2*A);
    }
}
```

Your program should allow the user to specify values for A, B, and C. It should call the subroutine to compute a solution of the equation. If no error occurs, it should print the root. However, if an error occurs, your program should catch that error and print an error message. After processing one equation, the program should ask whether the user wants to enter another equation. The program should continue until the user answers no.

Discussion:

This is really just a fairly easy exercise in using exceptions. The `root()` subroutine must be called in a try statement. There must be a catch clause to handle the `IllegalArgumentException` that might be thrown by the routine. The catch clause can simply print an error message:

```
try {
    solution = root(A,B,C);
    TextIO.putln("A solution of the equation is " + solution);
}
catch (IllegalArgumentException e) {
    TextIO.putln("Sorry, I can't find a solution.");
    TextIO.putln(e.getMessage());
}
```

Note that I've put the output statement that prints the solution inside the try statement. If an `IllegalArgumentException` is thrown by the subroutine, then this output statement will not be executed since the computer will jump immediately to the catch clause. You have to be careful about things like this. It wouldn't do to have the output statement after the `try..catch` statement since then the computer would still try to execute the output statement after handling an *IllegalArgumentException*. There are other ways to deal with this problem. For example, since the `try..catch` statement occurs in a while loop, we could put a `continue` statement in the catch clause to abort further processing when an exception occurs:

```
try {
    solution = root(A,B,C);
}
catch (IllegalArgumentException e) {
    TextIO.putln("Sorry, I can't find a solution.");
    TextIO.putln(e.getMessage());
    continue; // jump back to start of the while loop.
}
```

```
/* We only get to this point if NO exception was thrown.
   So we know that the solution was computed successfully. */
```

```
TextIO.putln("A solution of the equation is " + solution);
```

The rest of the main program is pretty standard. A complete solution is shown below.

The Solution:

```
/**
 * This program will compute and print one of the solutions
 * to an equation of the form  $A \cdot X^2 + B \cdot X + C = 0$ , where
 * A, B, and C are numbers entered by the user. It depends
 * on the non-standard class TextIO for doing input/output.
 */
public class Quadratic {

    public static void main(String[] args) {

        double A, B, C; // Coefficients in the equation.
        double solution; // The solution computed for the equation.
        boolean goAgain; // Set to true if the user wants to
                        // solve another equation.

        TextIO.putln("This program will print a solution of an equation");
        TextIO.putln("of the form  $A \cdot X^2 + B \cdot X + C = 0$ , where A, B, and");
        TextIO.putln("C are values that you enter.");

        do {

            /* Get the coefficients from the user. */

            TextIO.putln();
            TextIO.putln("Enter values for A, B, and C:");
            TextIO.put("A = ");
            A = TextIO.getlnDouble();
            TextIO.put("B = ");
            B = TextIO.getlnDouble();
            TextIO.put("C = ");
            C = TextIO.getlnDouble();
            TextIO.putln();

            /* Print the solution, or an error message, if
               there is no solution. */

            try {
                solution = root(A,B,C);
                TextIO.putln("A solution of the equation is " + solution);
            }
            catch (IllegalArgumentException e) {
                TextIO.putln("Sorry, I can't find a solution.");
                TextIO.putln(e.getMessage());
            }

            /* Find out whether the user wants to go again. */

            TextIO.putln();
            TextIO.put("Do you want to solve another equation? ");
            goAgain = TextIO.getlnBoolean();

        } while (goAgain);

    } // end main

    /**
     * Returns the larger of the two roots of the quadratic equation
```

```
* A*x*x + B*x + C = 0, provided it has any roots. If A == 0 or
* if the discriminant, B*B - 4*A*C, is negative, then an exception
* of type IllegalArgumentException is thrown.
*/
static public double root( double A, double B, double C )
    throws IllegalArgumentException {
    if (A == 0) {
        throw new IllegalArgumentException("A can't be zero.");
    }
    else {
        double disc = B*B - 4*A*C;
        if (disc < 0)
            throw new IllegalArgumentException("Discriminant < zero.");
        return (-B + Math.sqrt(disc)) / (2*A);
    }
}

} // end class
```

Last modified: Wednesday, 13 May 2020, 12:43 PM