```java
// com/github/sahasatvik/math/MathParser.java

package com.github.sahasatvik.math;

/**
 * MathParser contains methods for solving simple operations involving
 * arithmetic operators and functions.
 * These methods can be accessed by subclasses of MathParser.
 *
 *      @author         Satvik Saha
 *      @version        1.0, 16/10/2016
 *      @since          1.0
 */

public class MathParser {

        /**
         * Checks whether the String passed to it can be parsed as a number.
         *
         *      @param  str                     the String to be tested
         *      @return                         true if the String can be parsed as a number
         *      @since  1.0
         */

        protected static boolean isNumber (String str) {
                try {
                        /* Return true only if parseDouble(String) doesn't complain! */
                        Double.parseDouble(str);
                        return true;
                } catch (Exception e) {
                        return false;
                }
        }

        /**
         * Calculates the factorial of a number.
         *
         *      @param  x                       the number whose factorial is to be calculated
         *      @return                         the factorial of the number passed
         *      @since  1.0
         */

        protected static double factorial (double x) {
                /* Special cases! */
                if (x < 2)
```

```
                return 1;
        double n = 1;
        while (x > 0)
                n *= x--;
        return n;
}

/**
 * Solves and returns the result of a simple binary expression.
 * Only the following operators are supported : <pre>{@code
 *      ^       -       power
 *      /       -       division
 *      *       -       multiplication
 *      +       -       addition
 *      -       -       subtraction}</pre>
 *
 *      @param  a                       the operand on the left
 *      @param  op                      the opearator
 *      @param  b                       the operand on the right
 *      @return                         the result on evaluating the expression
 *      @since  1.0
 */

protected static double solveBinaryOperation (double a, String op, double b) {
        double result = 0.0;
        /*
         * Match the operator against a list of supported ones, then
         * perform the appropriate operation.
         */
        if (op.equals("^")) {
                result = Math.pow(a, b);
        } else if (op.equals("%")) {
                result = a % b;
        } else if (op.equals("/")) {
                result = a / b;
        } else if (op.equals("*")) {
                result = a * b;
        } else if (op.equals("+")) {
                result = a + b;
        } else if (op.equals("-")) {
                result = a - b;
        }
        return result;
}
```

```
/**
 * Solves and returns the result of an expression involving a function
 * with only one operand.
 * Only the following function names are supported : <pre>{@code
 *      abs     -       absolute value
 *      fct     -       factorial
 *      exp     -       exponentiation
 *      log     -       logarithm (base 'e')
 *      rad     -       convert degrees to radians
 *      deg     -       convert radians to degrees
 *      sin     \
 *      cos     |
 *      tan      \      standard trigonometric
 *      sec      /      functions
 *      csc     |
 *      ctn     /}</pre>
 *
 *      @param  func                    the function name
 *      @param  x                       the operand
 *      @return                         the result on evaluating the expression
 *      @throws com.github.sahasatvik.math.FunctionNotFoundException
 *                                      thrown when func is not recognized
 *      @since  1.0
 */

protected static double solveUnaryFunction (String func, double x)
                                        throws FunctionNotFoundException {
        double result = 0.0;
        /*
         * Math the function name against supported ones, then
         * perform the appropriate operation.
         */
        if (func.equals("sin")) {
                result = Math.sin(x);
        } else if (func.equals("cos")) {
                result = Math.cos(x);
        } else if (func.equals("tan")) {
                result = Math.tan(x);
        } else if (func.equals("csc")) {
                result = 1.0/Math.sin(x);
        } else if (func.equals("sec")) {
                result = 1.0/Math.cos(x);
        } else if (func.equals("ctn")) {
                result = 1.0/Math.tan(x);
        } else if (func.equals("rad")) {
```

```
                result = Math.toRadians(x);
            } else if (func.equals("deg")) {
                result = Math.toDegrees(x);
            } else if (func.equals("fct")) {
                result = factorial(x);
            } else if (func.equals("abs")) {
                result = Math.abs(x);
            } else if (func.equals("exp")) {
                result = Math.exp(x);
            } else if (func.equals("log")) {
                result = Math.log(x);
            } else {
                /*
                 * Throw an Exception if the function name does not
                 * match any supported one.
                 */
                throw new FunctionNotFoundException(func + "[]");
            }
            return result;
        }
}
```