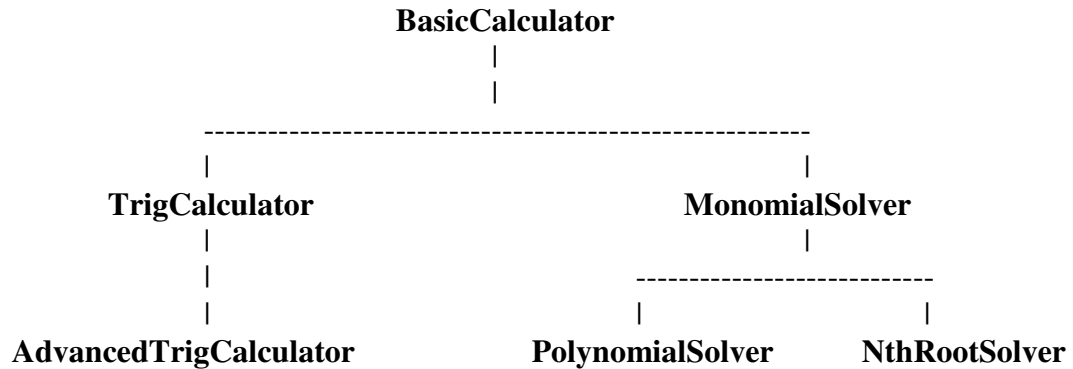


Due Date : Thursday, 24th April 2008 by 11PM (To be submitted online via SPARK)

Project 5: Implementing a Hierarchy

Implement a hierarchy of different types of calculators. The grandparent class, or the highest class in the hierarchy, will be `BasicCalculator`. Its subclass will be `MonomialSolver` and `TrigCalculator`. The children of `MonomialSolver` are `PolynomialSolver` and `NthRootSolver`. The child of `TrigCalculator` is `AdvancedTrigCalculator`. This hierarchy is shown below.



Do not worry about buttons for the numbers and functionality: you will not be using graphics for this project.

All the calculators, and the methods that they should have, are described below.

BasicCalculator: This is the simplest form of a calculator found: a four function calculator. It should have the following five public methods. You are welcome to use methods in the `Math` class for this part of the project.

- add:** This method takes two parameters of type `double` and returns their sum.
- subtract:** This method takes two parameters of type `double` and returns the difference of the second subtracted from the first.
- multiply:** This method takes two parameters of type `double` and returns their product.
- divide:** This method takes two parameters of type `double` and returns the difference of the first divided by the second.
- power:** This method takes two parameters of type `double` and returns the result of raising the first parameter to the power of the second.

MonomialSolver: In general, engineering is using the tools available to create or improve things, but MonomialSolver really emphasizes that. Using only inherited methods, this class should solve equations of the form $a*x + b = c$. Besides inherited methods, this class needs only one more public method to solve these equations.

solveEquation: This method takes four parameters, all of which are integers. In the equation above, a, b, and c are all parameters; a fourth integer, n, will also be a parameter. In this class, n will be ignored; when this method is overridden in subclasses, you will use n. Remember to use inherited methods and not to directly call methods in the Math class, nor to use mathematical operators explicitly. X should be returned by the method.

PolynomialSolver: Override the solveEquation method to solve an n^{th} -degree polynomial of the form $(ax + b)^n = c$. You do not need to implement any new methods for this class, but you should use inherited methods instead of methods in the Math class or mathematical operators, just as you did with MonomialSolver.

NthRootSolver: Override the solveEquation method to solve an nth-root equation of the form $(ax + b)^{1/n} = c$. You do not need to implement any new methods for this class.

TrigCalculator: This is really an abstract idea for a calculator, so the class should also be abstract. Just as engineers like to augment the size of their tool kits, this class should expand on the functionality of its parent. Add the following public methods with the following functionality. ***Again, you may use the methods in the Math class when implementing these methods.***

sine: Takes a double as a parameter; this double represents an angle in radians. Returns the sine of the parameter.

cosine: Takes a double representing a number of radians as a parameter. Returns the cosine of the angle.

AdvancedTrigCalculator: Expand on the available methods from BasicTrig class to perform some functionality that is commonly used in the advanced trigonometric calculations.

tangent: Returns the tangent of a double passed as a parameter; this double represents an angle in radians.

secant: Returns the secant of a double passed as a parameter; this double represents an angle in radians.

cosecant: Returns the cosecant of a double passed as a parameter.

The following equations are given as a reference for calculating tangent, secant and cosecant of an angle A.

$$\text{Tangent}(A) = \text{sine}(A) / \text{cosine}(A)$$

$$\text{Secant}(A) = 1 / \text{cosine}(A)$$

$$\text{Cosecant}(A) = 1/\text{sine}(A)$$

The Driver Class:

Write a driver for the class that uses an instance of a PolynomialSolver, an NthRootSolver, MonomialSolver, TrigCalculator and AdvancedTrigCalculator. Your driver should use some methods from the BasicCalculator class to show that it works correctly, Additionally, use the chart below to how to use each of the classes. In addition to the following examples, call your methods three more times and explain in the comments how you know you got the correct answers.

Class	Use
PolynomialSolver	Solve the equation $(6x - 40)^7 = 953$. Solve this equation by hand to verify that your PolynomialSolver is giving the correct answer.
NthRootSolver	Solve the equation $(12x + 96)^{0.2} = 3$. Solve the equation by hand to verify the answer that your NthRootSolver gives.
MonomialSolver	Solve the equation $(10x + 25) = 50$. Solve the equation by hand to verify the answer that your MonomialSolver gives.
TrigCalculator	Calculate $\sin(1.05)$, $\cos(3.14)$. None of these should give you an error.
AdvancedTrigCalculator	Finally, calculate $\text{cosecant}(0.65)$, $\text{tangent}(0.3)$, and $\text{secant}(10)$. None of these should yield an error.