Introducing Zoom

The object oriented modeling language for Mathematics

Prepared for

Project Zove

Monday, 2 March 2009

Version 0.1

Prepared by

Team Zove

Revision and Signoff Sheet

Change Record

Date	Author	Version	Change reference
2009-02-12	Sam Huang	0.1	Initial draft for review/discussion

Reviewers

Name	Version approved	Position	Date

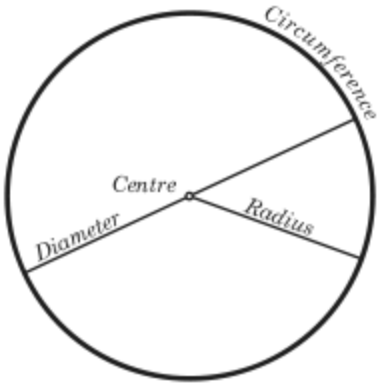
Table of Contents

1 Introduction

Zove, mean for 'solve', is a software designed to solve mathematical problems automatically.

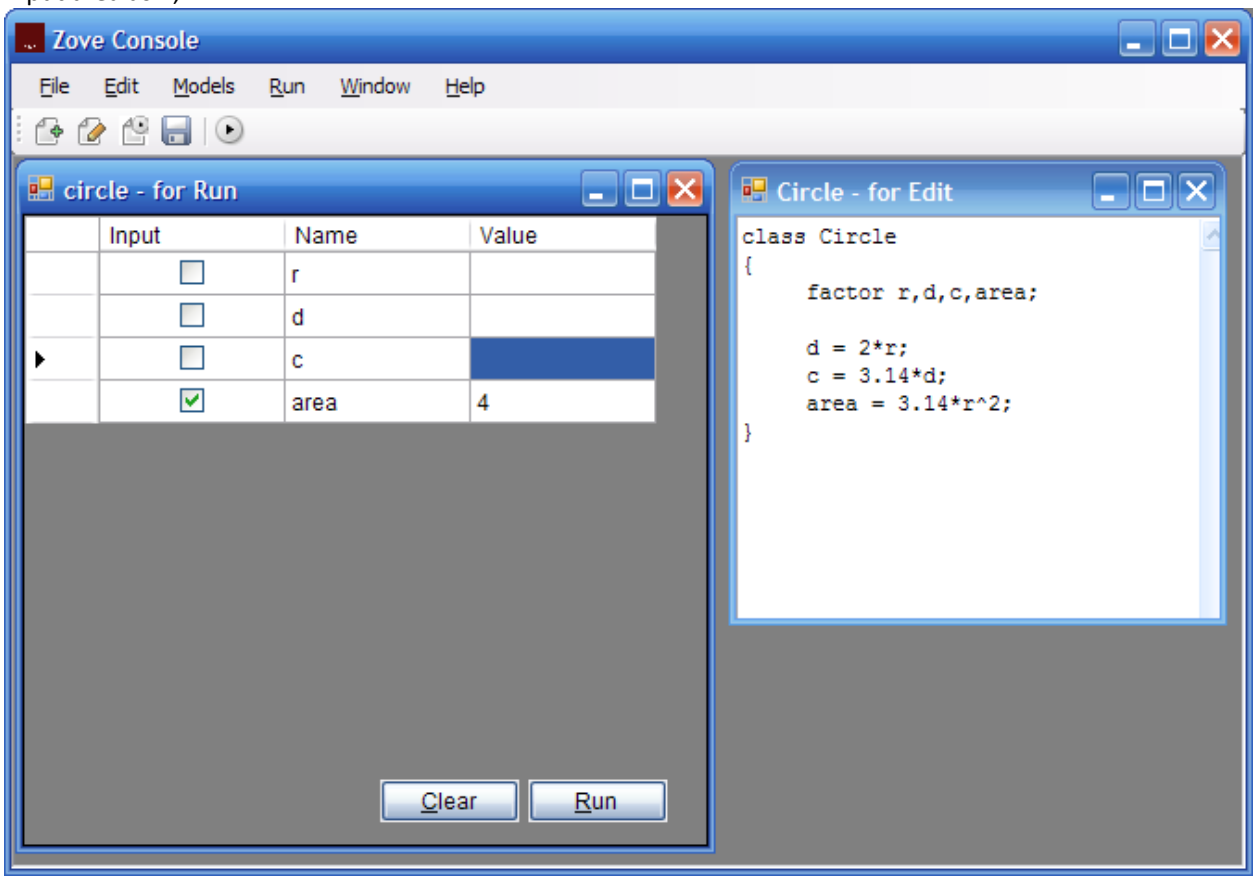
Zoom, acronym for Zove Object Oriented Modeling, is a language building models in mathematical equations and later used by the Zove software.

Example 1. Circle
The area of the circle is known as 4, to find the rest of the properties.



The definition of a circle is shown in right, the solve window's on left.

1. Input area as 4;



2. Run the model, the rest is found.

2 Modeling in equations

As shown from the last example, modeling is pretty straightforward in Zoom: using formulas. The formula used is nothing different than those from text books.

Data type

In the last example of Cube, its edge, area and volume are defined as a 'factor'. Factor is a data type for numbers.

2.2 Equations

Either sides of the equation mark can be mathematical expressions. This means they can be written as

a + b/3 = c – d*5;

And more, a sequence of equations can be written in a line:

a + b/3 = c – d*5 = e^2 + f;

2.3 Operators

Besides +, -, * and / as familiar, power is represented by ^, and log is '. Such as

The square of a: a^2

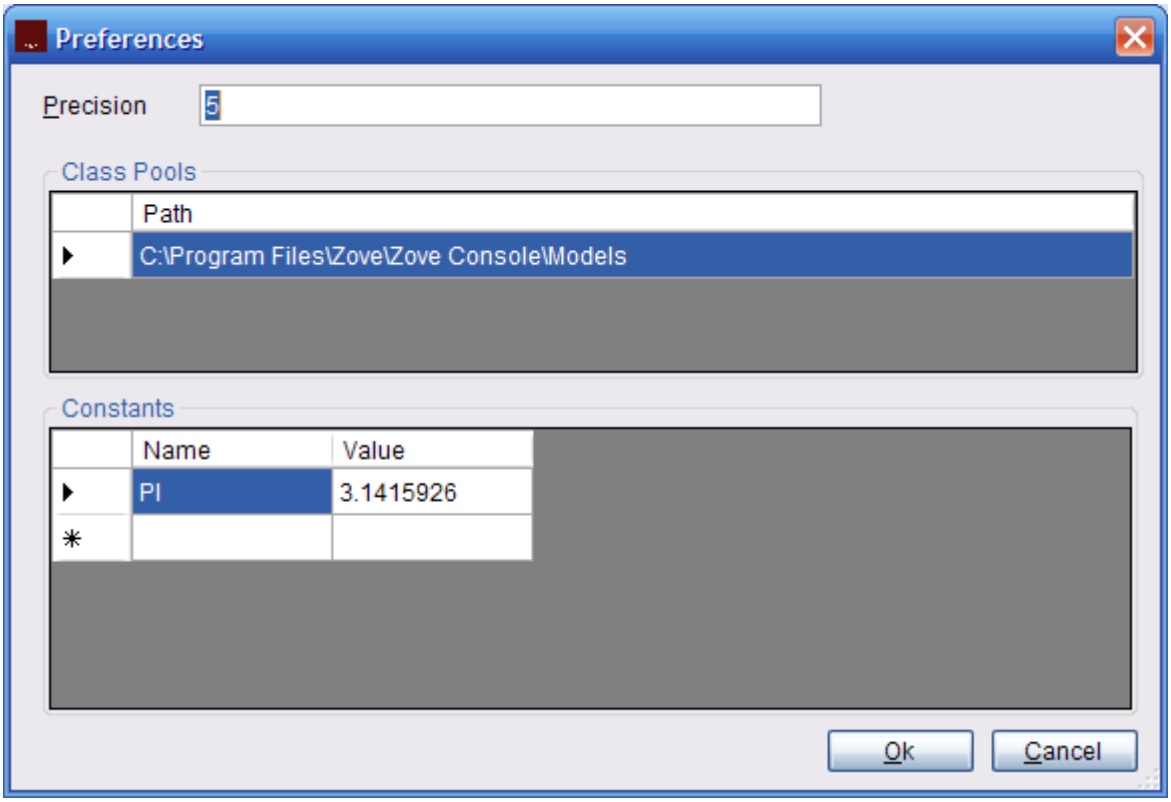
The square root of a: a'2

2.4 Functions

The built in functions are listed in the Appendix I, covering common mathematical functions like trigonometric, floor, ceiling, etc.

2.5 Constants

Constants can be defined in the Preferences dialog, via menu Edit > Preferences



Constants can be used nothing different as a factor.

```
class Triangle
{
  factor A, B, C;

  A = PI- B - C;
}
```

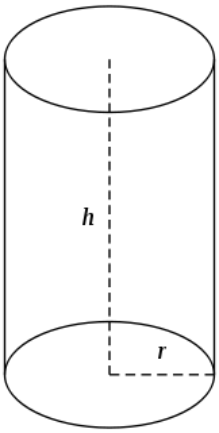
3 Object Oriented Modeling

3.1 What is Object Orientation?

Class is a mean to encapsulate the definitions of an object, so that it could be reused in other places.

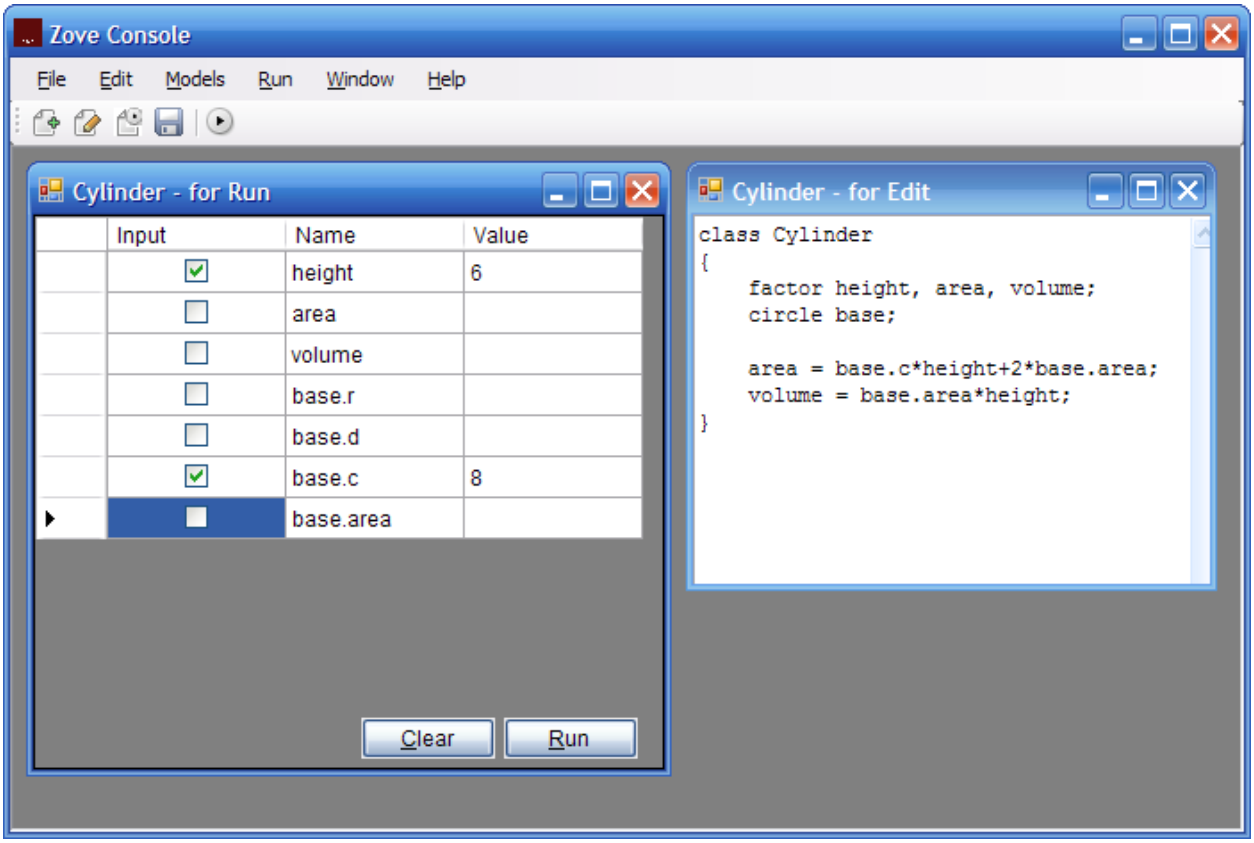
3.2 Encapsulation

A cylinder is an object whose base is a circle.

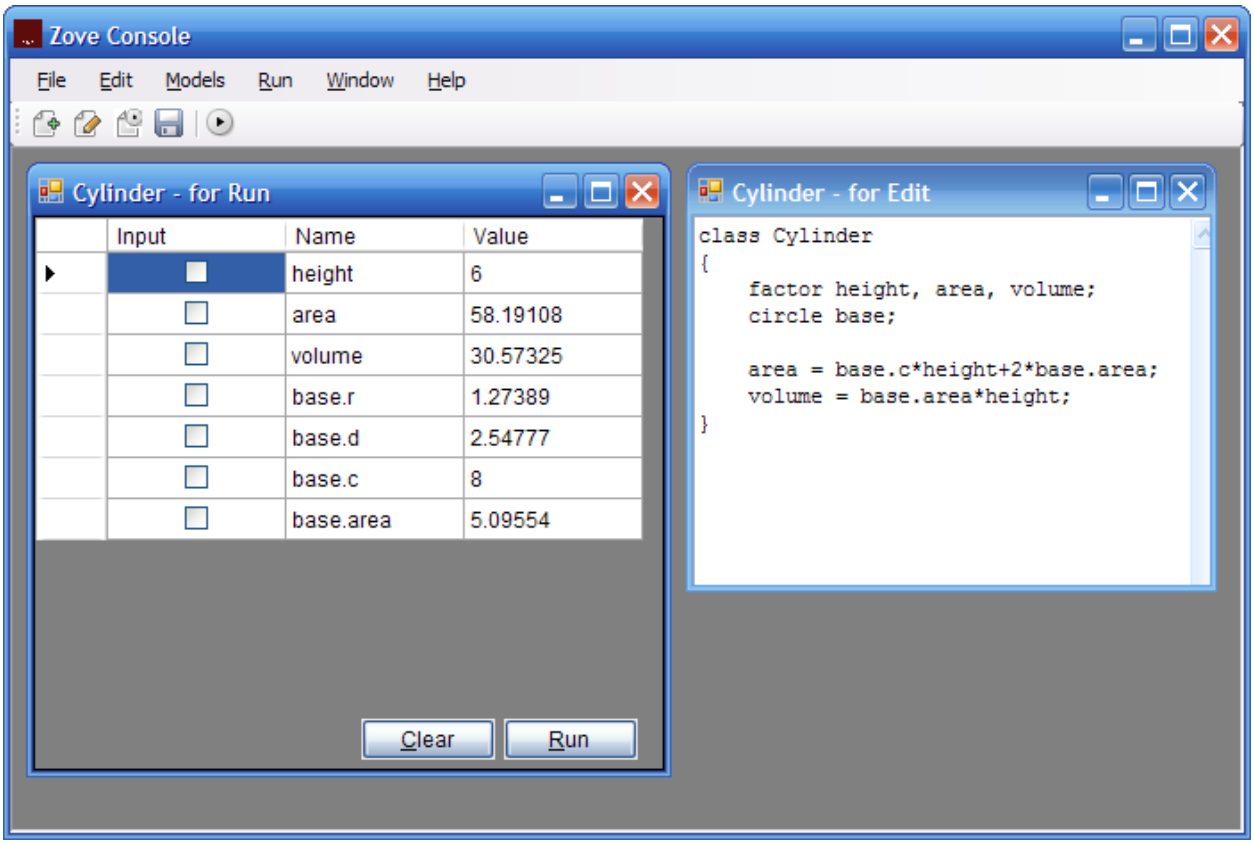



```
class Cylinder
{
    factor height, area, volume;
    circle base;

    area = base.c*height+2*base.area;
    volume = base.area*height;
}
```

The side of the cylinder is actually a rectangle, given the height and the circumstance of the bottom circle as input.



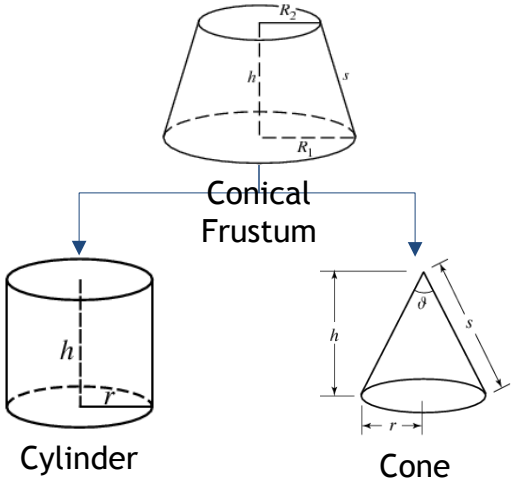
Press Run, the rest is calculated.



3.3 Inheritance

In another perspective, a cylinder can be viewed as a conical frustum whose bottom radius is the same as the top radius; and similarly, a cone can be viewed as a conical frustum whose top radius is zero.

So they have an inheritance relationship as



The conical frustum is defined as

```
class ConicalFrustum
{
      factor slantAngle, height, volume, slantArea, slantHeight, area;
      circle top, bottom;
      RightTriangle t;

      slantAngle = t.B;
      slantArea = PI*slantHeight*(top.r+bottom.r);
      slantHeight = t.c;
      height = t.b;
      t.a = bottom.r - top.r;
      area = top.area + bottom.area + slantArea;

      volume = PI*height*(top.r^2+top.r*bottom.r+bottom.r^2)/3;
}
```

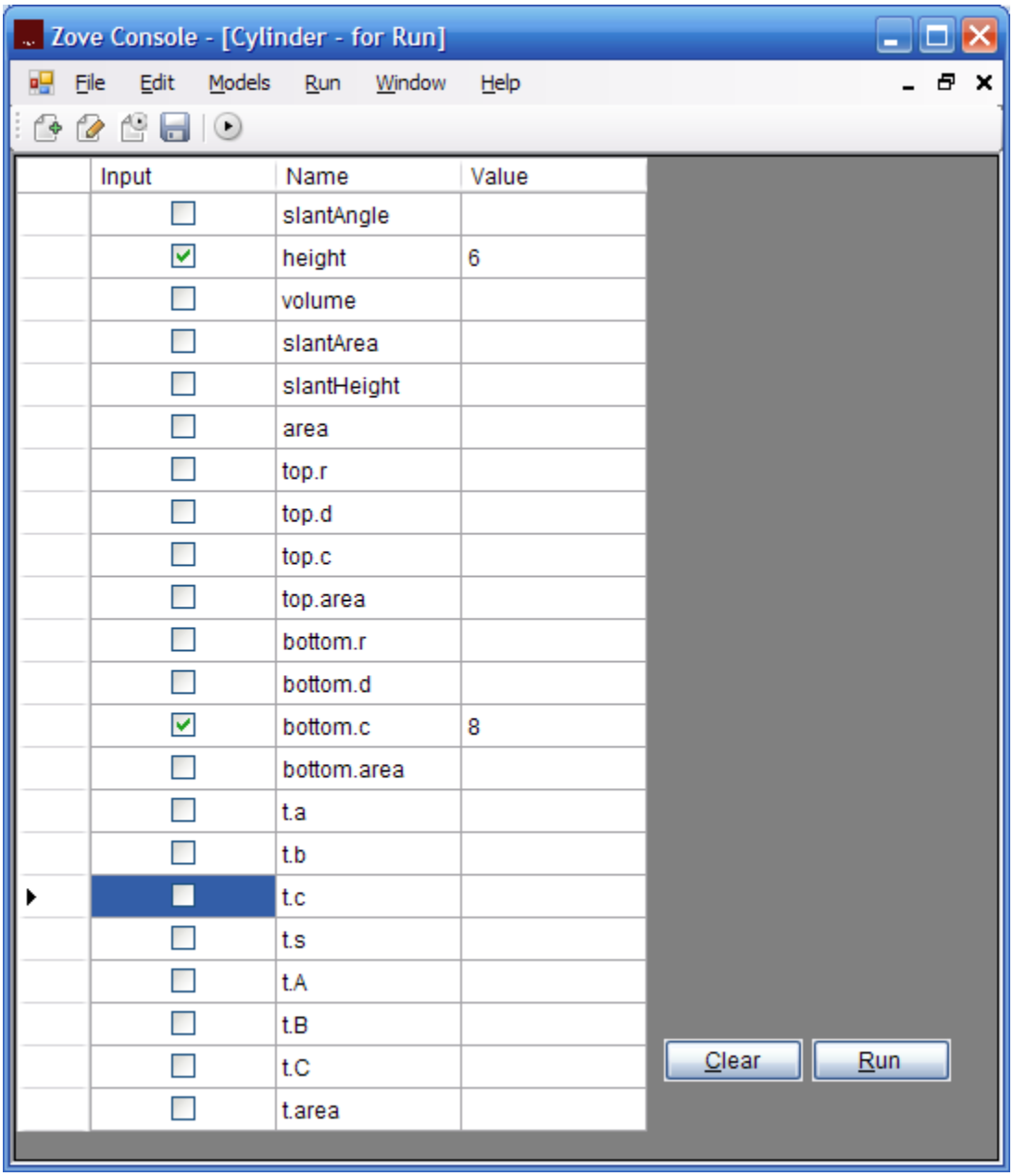
For cylinder, it is

```
class Cylinder extends ConicalFrustum
{
      top.r = bottom.r;
}
```

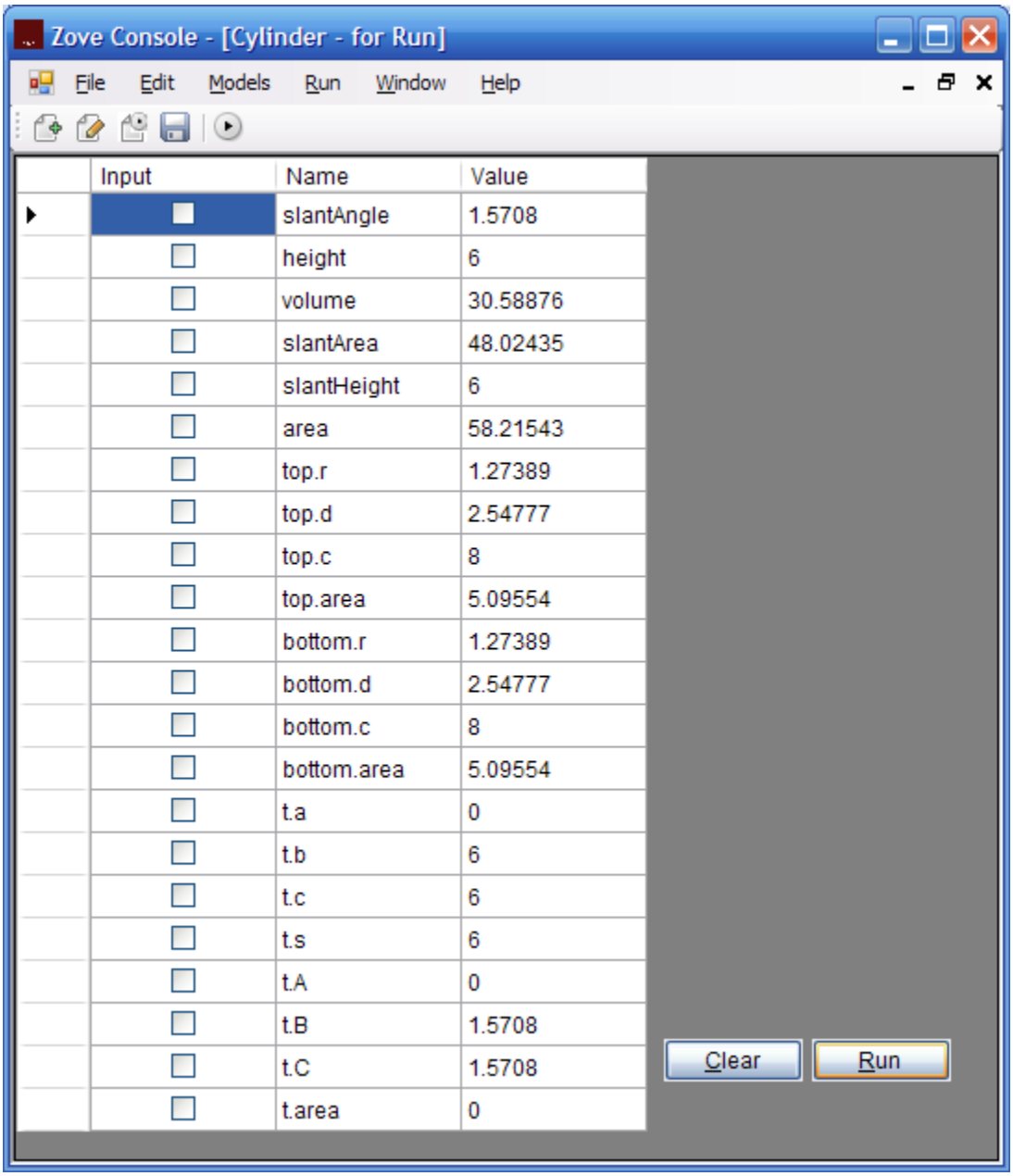
For cone, it is

```
class Cone extends ConicalFrustum
{
    top.r = 0;
}
```

Using the same input from Example 2.1 to test the cylinder defined in this section:



The rest are computed



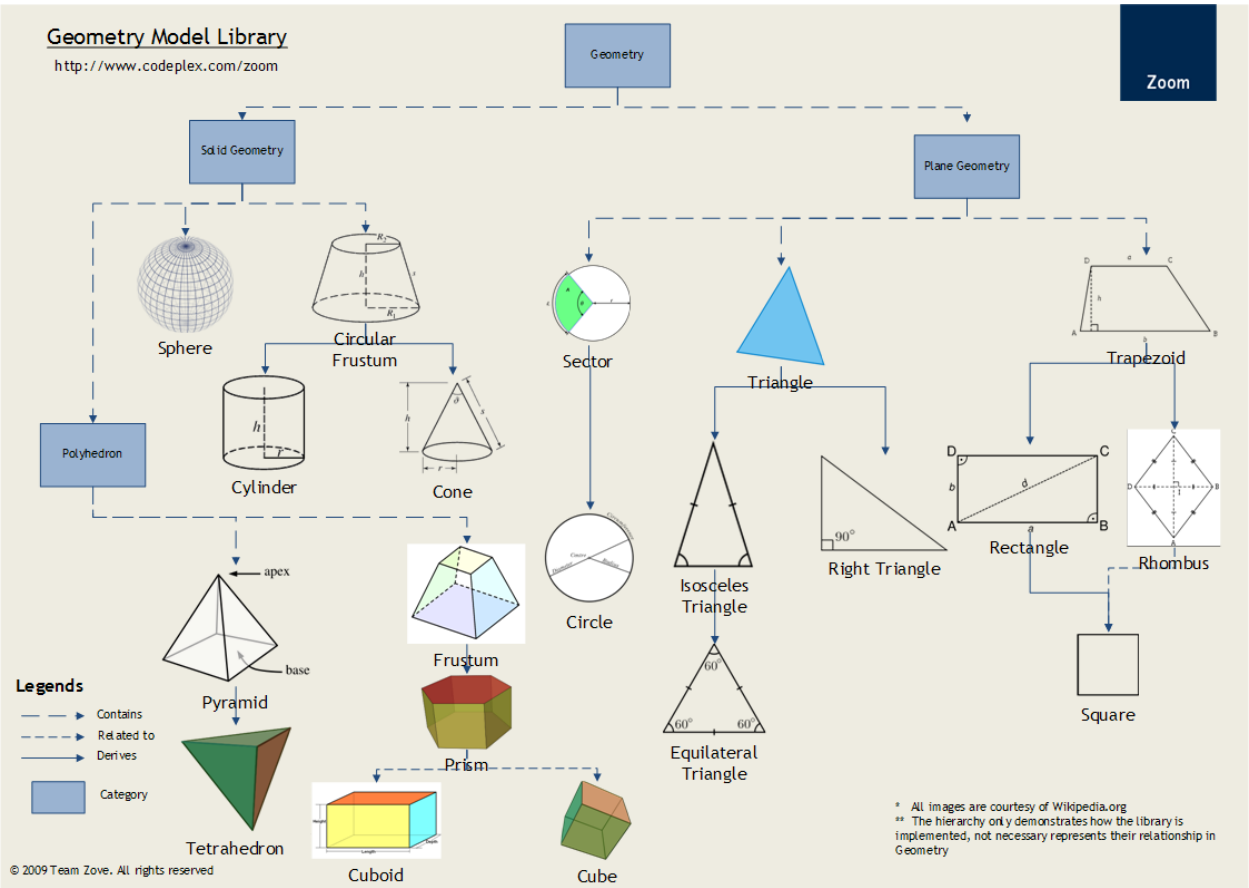
4 The Basic Classes

4.1 Overview

Zove is shipped with a collection of classes of geometry in Zoom, helps the users to quickly build up their applications for real world problems as well as examples of building models in Zoom.

The collection is composed of the common objects in plane and solid geometry, such as circle, triangle and cone. Via the Zove Console menu Model > Plane or Model > Solid they can be tested or edited.

Below is a hierarchy of the library, the classes are subjected to change as the project is rapid growing.



4.2 Triangle

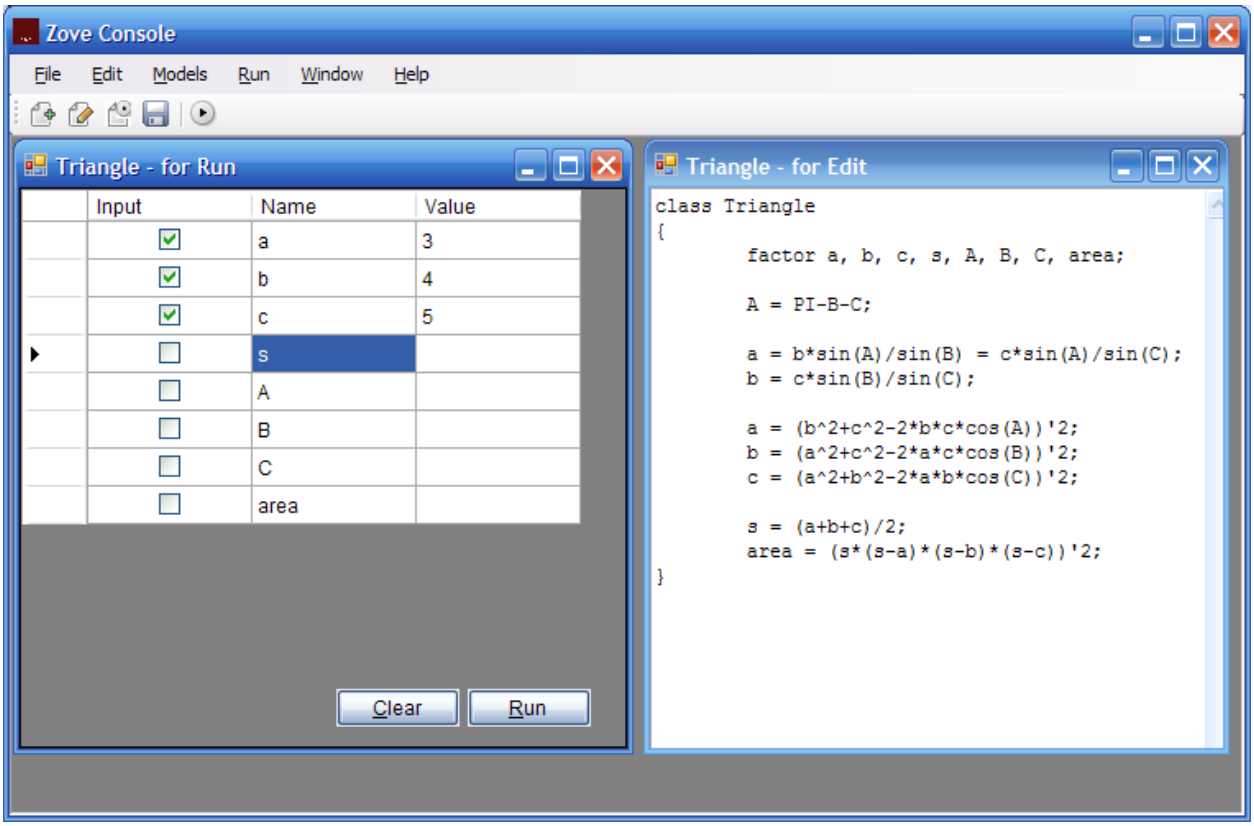
Triangle is very useful in building up other shapes, and is widely used throughout the library.

It is defined as

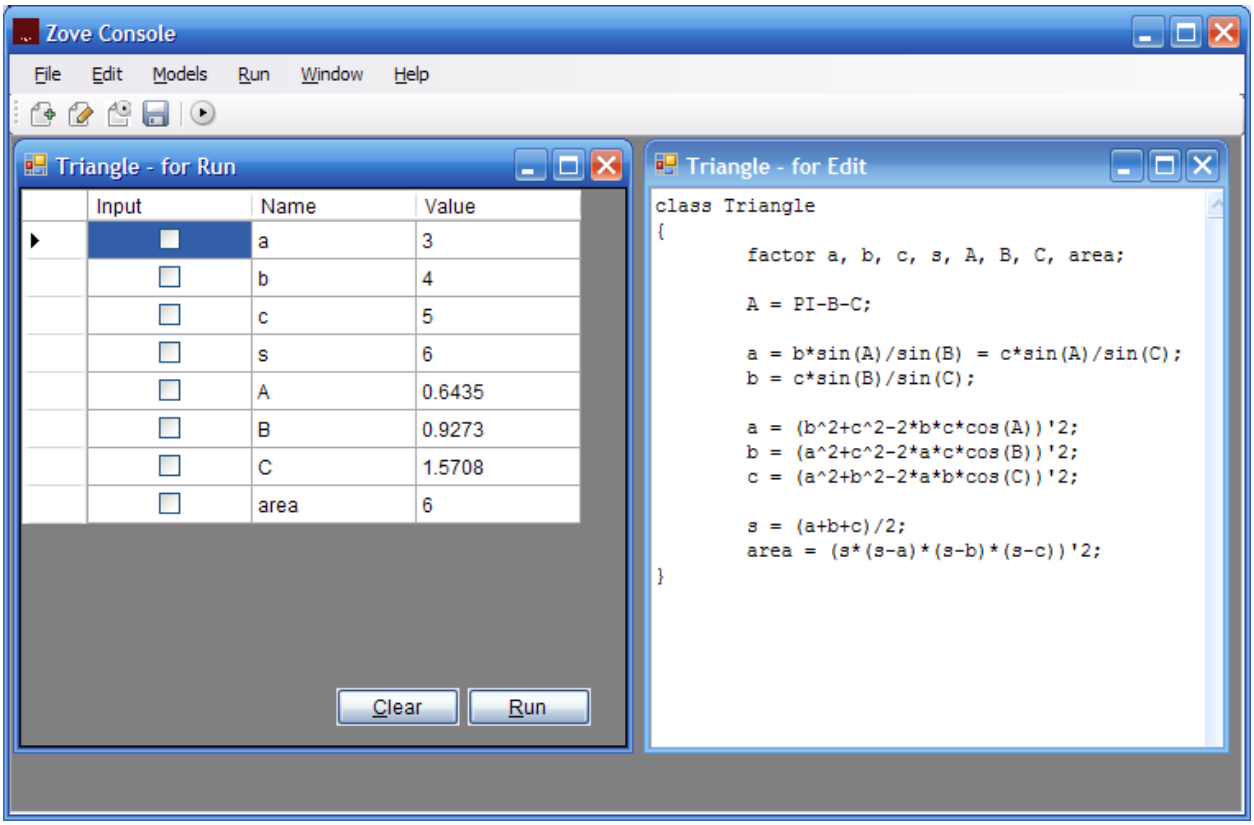
```
class Triangle
{
      factor a, b, c, s, A, B, C, area;

      A = PI-B-C;

      a = b*sin(A)/sin(B) = c*sin(A)/sin(C);
      b = c*sin(B)/sin(C);

      a = (b^2+c^2-2*b*c*cos(A))'2;
      b = (a^2+c^2-2*a*c*cos(B))'2;
      c = (a^2+b^2-2*a*b*cos(C))'2;

      s = (a+b+c)/2;
      area = (s*(s-a)*(s-b)*(s-c))'2;
}
```

The first equation denotes the sum of three angles is 180 degree, and PI in radian. The following two equations represent Law of sines[link], while the next three equations represent Law of cosines[link]. The last two equations defines the area's relationship with the three edges with Heron's formula[link].

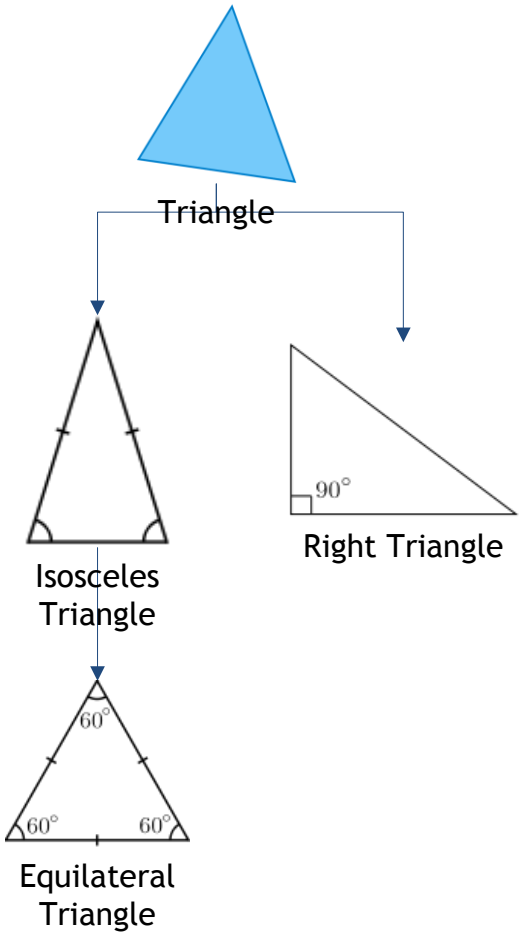
Example 4.2 Triangle
Given the length of the three edges, finds the rest values of the triangle.



The rest are found as

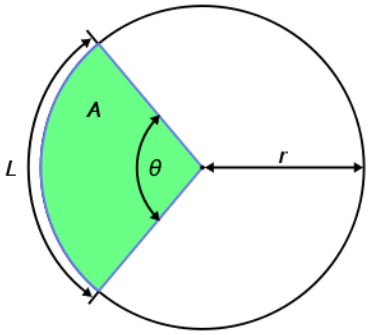


Its derivatives includes isosceles triangle, right triangle and equilateral triangle, they are also shipped in the class library, as the relationship depicted as



4.3 Circle & Sector

A circular sector is a portion of a circle, it has all the attributes similar to a circle.

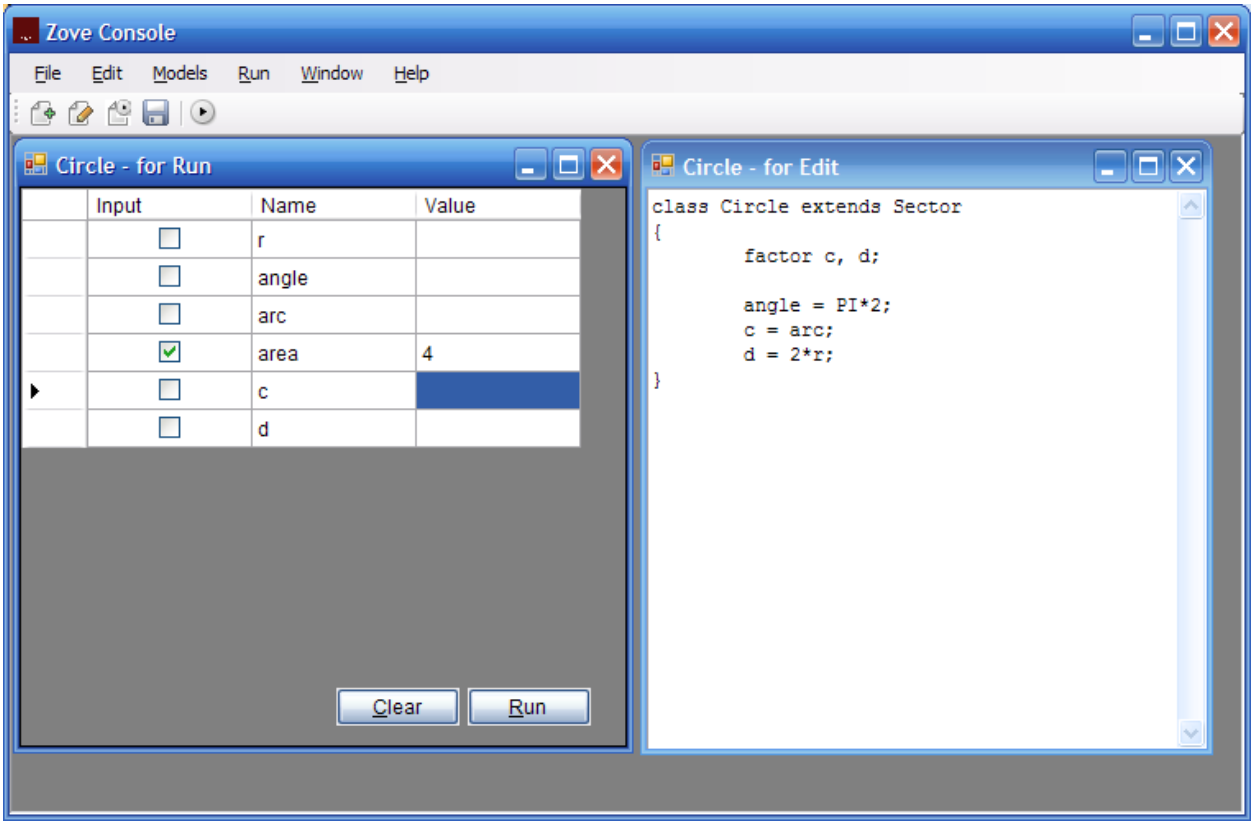



```
class Sector
{
      factor r, angle, arc, area;

      angle = arc/r;
      arc = angle*r;
      area = angle*r^2/2;
}
```

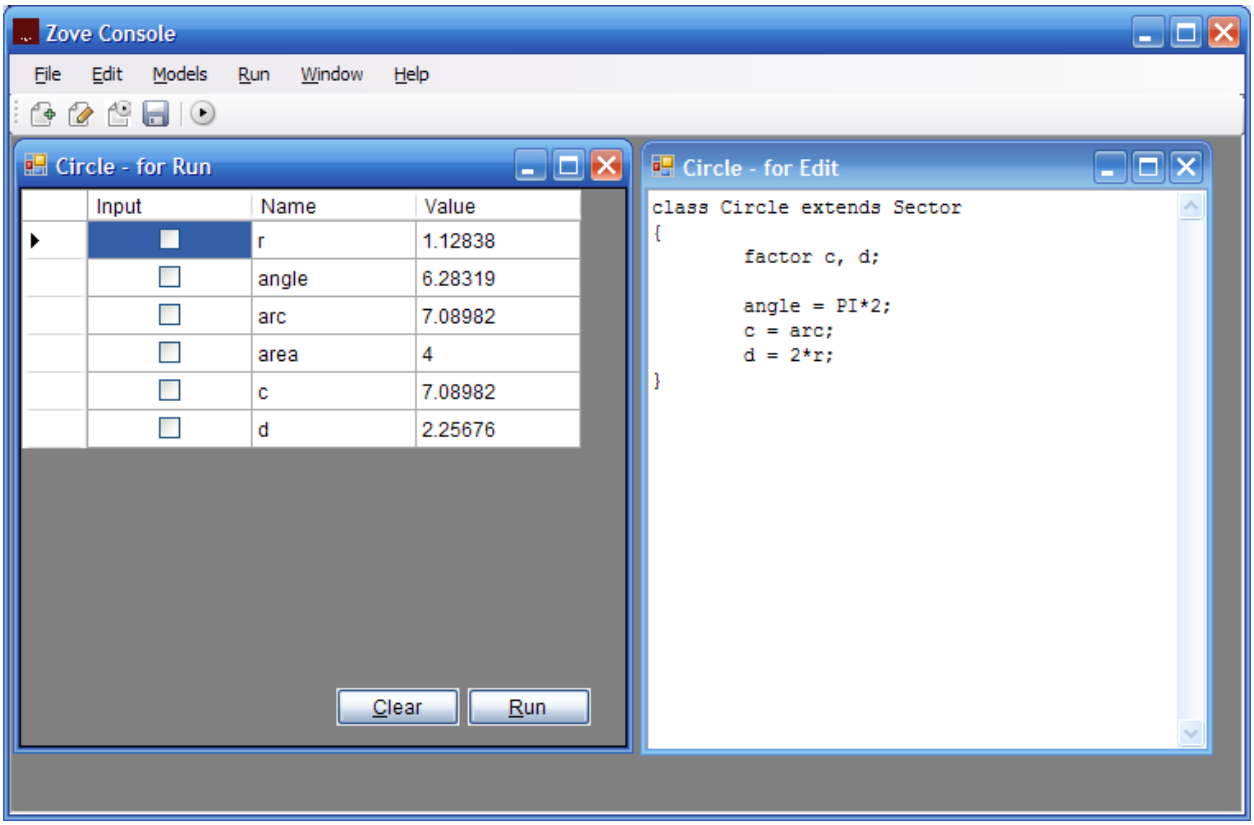
After the Sector is defined, circle can be altered as

```
class Circle extends Sector
{
	factor c, d;

	angle = PI*2;
	c = arc;
	d = 2*r;
}
```

Given area as the same as Example 1.1



The rest is found as



4.4 More…

Users can find out more by have a glimpse on the sources of the library.

5 Finale

The software ships a GUI program and a set of models ready to be used. During its rapid development, the release would be frequently presented in the official web site. The development team is eager to hear your feedbacks, any comments or suggestions help bettering software might finally ease your work and life, as the project sincerely hopes.

6 Appendix

6.1 Built-in Functions

Name	Description	Example
Sine	Sine	sin(3.1415926)
Cosine	Cosine	cos(3.1415926)
Floor	Floor	floor(3.141592)
Ceiling	Ceiling	ceil(3.1415926)
Max	Max	max(3, 5)

Min	Min	min(3, 5)

6.2 Preference Settings

Name	
Precision	The precision resulted numbers hold, default is 5
Class pool	The directory housing Zoom sources, default is the Model folder under the installation folder of Zove Console
Constant	The constants used in all equations

7 References

1. Object-oriented Programming http://en.wikipedia.org/wiki/Object-oriented_programming
2. Circle http://en.wikipedia.org/wiki/Circle
3. Circular Sector http://en.wikipedia.org/wiki/Circular_sector
4. Triangle http://en.wikipedia.org/wiki/Triangle
5. Heron's Formula http://mathworld.wolfram.com/ConicalFrustum.html
6. Trigonometry http://en.wikipedia.org/wiki/Trigonometry
7. Conical Frustum http://mathworld.wolfram.com/ConicalFrustum.html
8. Cone http://en.wikipedia.org/wiki/Cone_(geometry)
9. Cylinder http://en.wikipedia.org/wiki/Cylinder_(geometry)