Expanded types and conversion

Complex numbers are a concept introduced into mathematics to reason about values like “square root of -1”. The following Wikipedia resources give a good introduction to complex numbers:

http://en.wikipedia.org/wiki/Complex_number

Mathematical value types are implemented in Eiffel using *expanded* classes. Expanded classes are classes that implement value, not reference semantics. Expanded values are copied on assignment. They are always automatically initialized.

The goal of the exercise is to create a class `COMPLEX` that implements complex numbers in a comfortable way. The following operations should be supported:

- Accessing the real and imaginary parts of the complex numbers.
- Common operations on complex numbers: addition, subtraction, negation, multiplication, division.
- It should be possible to compute the square root from a complex number.
- It should be possible to have tuples of `DOUBLE` values automagically converted into `COMPLEX` numbers (see example text below).
- Printing complex numbers should result in a well-readable form.

As an example, the following code:

```eiffel
local
  c: COMPLEX
do
  c := [-1.0,0.0]
  print (c.square_root)
  print ("%N")
  c := [2.0,-1.0]
  c := c + [2.0,5.0]
  print (c.absolute)
  print ("%N")
  print (c / [2.0,0.0])
  print ("%N")
end
```

This should produce the following output:

```
[0+1i]
5
[4+-3i]
```