

Contents

1	Classes	2
1.1	imaginary – complex numbers and its functions	2
1.1.1	ComplexField – field of complex numbers	4
1.1.1.1	createElement – create Imaginary object	5
1.1.1.2	getCharacteristic – get characteristic	5
1.1.1.3	issubring – subring test	5
1.1.1.4	issuperring – superring test	5
1.1.2	Complex – a complex number	6
1.1.2.1	getRing – get ring object	7
1.1.2.2	arg – argument of complex	7
1.1.2.3	conjugate – complex conjugate	7
1.1.2.4	copy – copied number	7
1.1.2.5	inverse – complex inverse	7
1.1.3	ExponentialPowerSeries – exponential power series	8
1.1.4	AbsoluteError – absolute error	8
1.1.5	RelativeError – relative error	8
1.1.6	exp(function) – exponential value	8
1.1.7	expi(function) – imaginary exponential value	8
1.1.8	log(function) – logarithm	8
1.1.9	sin(function) – sine function	8
1.1.10	cos(function) – cosine function	8
1.1.11	tan(function) – tangent function	8
1.1.12	sinh(function) – hyperbolic sine function	8
1.1.13	cosh(function) – hyperbolic cosine function	8
1.1.14	tanh(function) – hyperbolic tangent function	8
1.1.15	atanh(function) – hyperbolic arc tangent function	9
1.1.16	sqrt(function) – square root	9

Chapter 1

Classes

1.1 `imaginary` – complex numbers and its functions

The module `imaginary` provides complex numbers. The functions provided are mainly corresponding to the `cmath` standard module.

- **Classes**
 - `ComplexField`
 - `Complex`
 - †`ExponentialPowerSeries`
 - †`AbsoluteError`
 - †`RelativeError`
- **Functions**
 - `exp`
 - `expi`
 - `log`
 - `sin`
 - `cos`
 - `tan`
 - `sinh`
 - `cosh`
 - `tanh`
 - `atanh`
 - `sqrt`

This module also provides following constants:

- e** :
This constant is obsolete (Ver 1.1.0).
- pi** :
This constant is obsolete (Ver 1.1.0).
- j** :
j is the imaginary unit.
- theComplexField** :
theComplexField is the instance of **ComplexField**.

1.1.1 ComplexField – field of complex numbers

The class is for the field of complex numbers. The class has the single instance **theComplexField**.

This class is a subclass of **Field**.

Initialize (Constructor)

ComplexField() \rightarrow *ComplexField*

Create an instance of ComplexField. You may not want to create an instance, since there is already **theComplexField**.

Attribute

zero :
It expresses The additive unit 0. (read only)

one :
It expresses The multiplicative unit 1. (read only)

Operations

operator	explanation
in	membership test; return whether an element is in or not.
repr	return representation string.
str	return string.

Methods

1.1.1.1 createElement – create Imaginary object

`createElement(self, seed: integer) → Integer`

Return a Complex object with `seed`.

`seed` must be complex or numbers having embedding to complex.

1.1.1.2 getCharacteristic – get characteristic

`getCharacteristic(self) → integer`

Return the characteristic, zero.

1.1.1.3 issubring – subring test

`issubring(self, aRing: Ring) → bool`

Report whether another ring contains the complex field as subring.

1.1.1.4 issuperring – superring test

`issuperring(self, aRing: Ring) → bool`

Report whether the complex field contains another ring as subring.

1.1.2 Complex – a complex number

Complex is a class of complex number. Each instance has a coupled numbers; real and imaginary part of the number.

This class is a subclass of **FieldElement**.

All implemented operators in this class are delegated to complex type.

Initialize (Constructor)

Complex(re: *number* im: *number*=0) \rightarrow *Imaginary*

Create a complex number.

re can be either real or complex number. If **re** is real and **im** is not given, then its imaginary part is zero.

Attribute

real :

It expresses the real part of complex number.

imag :

It expresses the imaginary part of complex number.

Methods

1.1.2.1 `getRing` – get ring object

`getRing(self)` → *ComplexField*

Return the complex field instance.

1.1.2.2 `arg` – argument of complex

`arg(self)` → *radian*

Return the angle between the x-axis and the number in the Gaussian plane.
radian must be Float.

1.1.2.3 `conjugate` – complex conjugate

`conjugate(self)` → *Complex*

Return the complex conjugate of the number.

1.1.2.4 `copy` – copied number

`copy(self)` → *Complex*

Return the copy of the number itself.

1.1.2.5 `inverse` – complex inverse

`inverse(self)` → *Complex*

Return the inverse of the number.

If the number is zero, `ZeroDivisionError` is raised.

1.1.3 ExponentialPowerSeries – exponential power series

This class is obsolete (Ver 1.1.0).

1.1.4 AbsoluteError – absolute error

This class is obsolete (Ver 1.1.0).

1.1.5 RelativeError – relative error

This class is obsolete (Ver 1.1.0).

1.1.6 exp(function) – exponential value

This function is obsolete (Ver 1.1.0).

1.1.7 expi(function) – imaginary exponential value

This function is obsolete (Ver 1.1.0).

1.1.8 log(function) – logarithm

This function is obsolete (Ver 1.1.0).

1.1.9 sin(function) – sine function

This function is obsolete (Ver 1.1.0).

1.1.10 cos(function) – cosine function

This function is obsolete (Ver 1.1.0).

1.1.11 tan(function) – tangent function

This function is obsolete (Ver 1.1.0).

1.1.12 sinh(function) – hyperbolic sine function

This function is obsolete (Ver 1.1.0).

1.1.13 cosh(function) – hyperbolic cosine function

This function is obsolete (Ver 1.1.0).

1.1.14 tanh(function) – hyperbolic tangent function

This function is obsolete (Ver 1.1.0).

1.1.15 atanh(function) – hyperbolic arc tangent function

This function is obsolete (Ver 1.1.0).

1.1.16 sqrt(function) – square root

This function is obsolete (Ver 1.1.0).