

Contents

1	Classes	2
1.1	vector – ベクトルオブジェクトとその計算	2
1.1.1	Vector – ベクトルクラス	3
1.1.1.1	copy – 自身のコピー	5
1.1.1.2	set – 他の compo を設定	5
1.1.1.3	indexOfNoneZero – 0 でない最初の位置	5
1.1.1.4	toMatrix – Matrix オブジェクトに変換	5
1.1.2	innerProduct(function) – 内積	7

Chapter 1

Classes

1.1 vector – ベクトルオブジェクトとその計算

- Classes
 - **Vector**
- Functions
 - **innerProduct**

このモジュールはある例外クラスを提供する。

VectorSizeError : ベクトルのサイズが正しくないことを報告. (主に二つのベクトルの演算において.)

1.1.1 Vector – ベクトルクラス

Vector はベクトルに対するクラス.

Initialize (Constructor)

Vector(*compo*: *list*) → *Vector*

compo から新しいベクトルオブジェクトを作成. *compo* は整数または **RingElement** のインスタンスである要素のリストでなければならない.

Attributes

compo :
ベクトルの成分を表す.

Operations

数学の世界での標準の通り, インデックスは 1 が最初だということに注意.

operator	explanation
$u+v$	ベクトルの和.
$u-v$	ベクトルの差.
$A*v$	行列とベクトルの積.
$a*v$	ベクトルのスカラー倍.
$v//a$	スカラー除算.
$v\%n$	compo の要素の n での剰余.
$-v$	各要素の符号を変える.
$u==v$	等しいかどうか.
$u!=v$	等しくないかどうか.
$v[i]$	ベクトルの i 番目の成分を返す.
$v[i] = c$	ベクトルの i 番目の成分を c に置き換える.
$\text{len}(v)$	compo の長さを返す.
$\text{repr}(v)$	compo の repr 文字列を返す.
$\text{str}(v)$	compo の string 文字列を返す.

Examples

```
>>> A = vector.Vector([1, 2])
>>> A
Vector([1, 2])
>>> A.compo
[1, 2]
```

```
>>> B = vector.Vector([2, 1])
>>> A + B
Vector([3, 3])
>>> A % 2
Vector([1, 0])
>>> A[1]
1
>>> len(B)
2
```

Methods

1.1.1.1 copy – 自身のコピー

`copy(self) → Vector`

`self` のコピーを返す.

1.1.1.2 set – 他の `compo` を設定

`set(self, compo: list) → (None)`

`self` の **compo** を新しい `compo` で置き換える.

1.1.1.3 indexOfNoneZero – 0 でない最初の位置

`indexOfNoneZero(self) → integer`

`self.compo` の 0 でない成分の最初のインデックスを返す.

† **compo** の全ての成分が 0 の場合, `ValueError` が起こる.

1.1.1.4 toMatrix – Matrix オブジェクトに変換

`toMatrix(self, as_column: bool=False) → Matrix`

createMatrix 関数を使い **Matrix** オブジェクトを返す.

もし `as_column` が `True` なら, `self` を縦ベクトルとみなした行列を返す. さもなくば, `self` を横ベクトルとみなした行列を返す.

Examples

```
>>> A = vector.Vector([0, 4, 5])
>>> A.indexOfNoneZero()
2
>>> print A.toMatrix()
0 4 5
>>> print A.toMatrix()
```

0
4
5

1.1.2 innerProduct(function) – 内積

`innerProduct(bra: Vector, ket: Vector) → RingElement`

bra と ket の内積を返す.

この関数は複素数体上の元に対するエルミート内積もサポートする.

† 返される値は成分の型に依存することに注意.

Examples

```
>>> A = vector.Vector([1, 2, 3])
>>> B = vector.Vector([2, 1, 0])
>>> vector.innerProduct(A, B)
4
>>> C = vector.Vector([1+1j, 2+2j, 3+3j])
>>> vector.innerProduct(C, C)
(28+0j)
```