Contents

Chapter 1

Classes

1.1 rational – 整数と有理数

rational モジュールはクラス Rational, クラス Integer, クラス RationalField, そして クラス IntegerRing として整数と有理数を提供.

- **Classes**
  - **Integer**
  - **IntegerRing**
  - **Rational**
  - **RationalField**

  このモジュールはまた以下のコンテンツを提供する:

theIntegerRing :
`theIntegerRing` は有理整数環を表す. **IntegerRing** のインスタンス.

theRationalField :
`theRationalField` は有理数体を表す. **RationalField** のインスタンス.

1.1.1 Integer – 整数

Integer は整数のクラス. 'int' と 'long' は除算において有理数を返さないので, 新しいクラスを作成する必要があった.

このクラスは **CommutativeRingElement** と long のサブクラス.

Initialize (Constructor)

Integer(`integer`: ***integer***) → ***Integer***

Integer オブジェクトを構成. もし引数が省略されたら, 値は 0 となる.

Methods

1.1.1.1 getRing – ring オブジェクトを得る

getRing(`self`) → ***IntegerRing***

IntegerRing オブジェクトを返す.

1.1.1.2 actAdditive – 2 進の加法鎖の加法

actAdditive(`self`, `other`: ***integer***) → ***Integer***

other に加法的に作用, すなわち, n は `other` の n 回の加算に拡大される. 結果としては以下と同じ:

```
return sum([+other for _ in range(self)])
```

しかし, ここでは 2 進の加法鎖を使う.

1.1.1.3 actMultiplicative – 2 進の加法鎖の乗法

actMultiplicative(`self`, `other`: ***integer***) → ***Integer***

other に乗法的に作用する, すなわち, n は `other` の n 回の乗算に拡大される. 結果としては以下と同じ:

```
return reduce(lambda x,y: x*y, [+other for _ in range(self)])
```

しかし, ここでは 2 進の加法鎖を使う.

1.1.2 IntegerRing – 整数環

有理整数環に対するクラス.
　このクラスは CommutativeRing のサブクラス.

Initialize (Constructor)

IntegerRing() → *IntegerRing*

　IntegerRing のインスタンスを作成. すでに theIntegerRing があるので, インスタンスを作成する必要がないかもしれない.

Attributes

zero :
　　加法の単位元 0 を表す. (読み込み専用)

one :
　　乗法の単位元 1 を表す. (読みこみ専用)

Operations

operator	explanation
`x in Z`	元が含まれているどうか返す.
`repr(Z)`	repr 文字列を返す.
`str(Z)`	str 文字列を返す.

Methods

1.1.2.1 createElement – Integer オブジェクトを作成

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

`seed` に対する Integer オブジェクトを作成.
`seed` は int 型, long 型 または rational.Integer でなければならない.

1.1.2.2 gcd – 最大公約数

gcd(`self`, `n`: *integer*, `m`: *integer*) → *Integer*

与えられた二つの整数の最大公約数を返す.

1.1.2.3 extgcd – 拡張 GCD

extgcd(`self`, `n`: *integer*, `m`: *integer*) → *Integer*

タプル (u, v, d) を返す; これらは与えられた二つの整数 `n` と `m` の最大公約数 d と, $d = \mathtt{n}u + \mathtt{m}v$ となる u, v.

1.1.2.4 lcm – 最小公倍数

lcm(`self`, `n`: *integer*, `m`: *integer*) → *Integer*

与えられた二つの整数の最小公倍数を返す. もし両方とも 0 なら, エラーが起こる.

1.1.2.5 getQuotientField – 有理数体オブジェクトを得る

getQuotientField(`self`) → *RationalField*

有理数体 (RationalField) を返す.

1.1.2.6 issubring – 部分環かどうか判定

issubring(`self`, `other`: Ring) → *bool*

もう一方の環が部分環として整数環を含んでいるか報告.

もし other も整数環なら, 出力は True.　その他の場合もう一方の整数環の issuperring メソッドのおける実装に依存.

1.1.2.7　issuperring – 含んでいるかどうか判定

issuperring(`self`, `other`: Ring) → *bool*

整数環がもう一方の環を部分間として含んでいるか報告.
もし other も整数環なら, 出力は True.　その他の場合もう一方の整数環の issubring メソッドのおける実装に依存.

1.1.3 Rational – 有理数

有理数のクラス.

Initialize (Constructor)

Rational(`numerator`: ***numbers***, `denominator`: ***numbers***=**1**)
→ ***Integer***

有理数は以下から構成:

- 整数,
- float
- Rational.

もし toRational メソッドがあれば, 他のオブジェクトを変換することができる. さもなくば TypeError が起こる.

Methods

1.1.3.1 getRing – ring オブジェクトを得る

getRing(`self`) → ***RationalField***

RationalField オブジェクトを返す.

1.1.3.2 decimalString – 小数を表す

decimalString(`self`, `N`: ***integer***) → ***string***

小数第 `N` 桁とした文字列を返す.

1.1.3.3 expand – 連分数による表現

expand(`self`, `base`: ***integer***, `limit`: ***integer***) → ***string***

もし `base` が自然数なら, 分母が `base` の高々`limit` 乗である最も近い有理数を返す.
さもなくば (すなわち, `base`=0), 分母が高々`limit` である最も近い有理数を返す.
`base` は負の整数であってはならない.

1.1.4 RationalField – 有理数体

RationalField は有理数体のクラス. このクラスは **theRationalField** という唯一のインスタンスを持つ.

このクラスは **QuotientField** のサブクラス.

Initialize (Constructor)

RationalField() → *RationalField*

RationalField のインスタンスを作成. すでに theRationalField があるので, インスタンスを作成する必要はないかもしれない.

Attributes

zero :
加法の単位元 0 を表す, すなわち Rational(0, 1). (読み込み専用)

one :
乗法の単位元 1 を表す, すなわち Rational(1, 1). (読み込み専用)

Operations

operator	explanation
`x in Q`	元が含まれているかどうか返す.
`str(Q)`	str 文字列を返す.

Methods

1.1.4.1 createElement – Rational オブジェクトを返す

createElement(`self`, `numerator`: *integer or* Rational, `denominator`: *integer*=1)
→ *Rational*

Rational オブジェクトを作成.

1.1.4.2 classNumber – 類数を得る

classNumber(`self`) → *integer*

有理数体の類数は 1 なので, 1 を返す.

1.1.4.3 getQuotientField – 有理数体オブジェクトを返す

getQuotientField(`self`) → *RationalField*

有理数体インスタンスを返す.

1.1.4.4 issubring – 部分環かどうか判定

issubring(`self`, `other`: Ring) → *bool*

もう一方の環が部分環として有理数体を含んでいるか報告.
もし other もまた有理数体なら, 出力は True. 他の場合もう一方の issuperring メソッドにおける実装に依存.

1.1.4.5 issuperring – 含んでいるかどうか判定

issuperring(`self`, `other`: Ring) → *bool*

有理数体がもう一方の環を部分環としてを含んでいるか報告.
もし other もまた有理数体なら, 出力は True. 他の場合もう一方の issubring メソッドにおける実装に依存.

Bibliography