

## 033: Prelude における Num クラスとインスタンスを完成させる

↑ up

- issued: 2020-04-15
- 分類: B 機能追加
- status: Closed (2021-10-31)

### 調査ログ

2020-05-09 (Sat)

058 を閉じるためにも、div を実装しよう。

まず、class (Num a, Ord a) => Read a where ... と書いた時点でエラー。Rename.renClassDecls で super class が2つ以上に対応していなかった。

しかし、いまは、まだ Int と Integer しかないので、Enum のみのサブクラスとして Integral を実装するしかないな。

Double, Float も実装。

2021-10-27 (Wed)

状況整理しよう。

参照するのは、Haskell 2010 language report の Standard Prelude<sup>\*1</sup> で、bunny においては、lib/Prelude.hs に実装が書かれる。

Num クラスは現状以下のように書かれていて、仕様通り：

```
class (Eq a, Show a) => Num a where
  (+), (-), (*) :: a -> a -> a
  negate      :: a -> a
  abs, signum :: a -> a
  fromInteger :: Integer -> a
  -- Minimal complete definition:
  -- All, except negate or (-)
  x - y    = x + negate y
  negate x = 0 - x
```

Real, Integral, Fractional, Floating も仕様通りにかかっている。

RealFrac は、round の定義が仕様とすこし違う書き方になっている。なにか問題を回避したのかな（覚え

---

\*1 <https://www.haskell.org/onlinereport/haskell2010/haskellch9.html#x16-1710009>

てない)

```
class (Real a, Fractional a) => RealFrac a where
  properFraction  :: (Integral b) => a -> (b, a)
  truncate, round :: (Integral b) => a -> b
  ceiling, floor  :: (Integral b) => a -> b
  -- Minimal complete definition: properFraction
  truncate x = m where (m, _) = properFraction x
  round x = let (n, r) = properFraction x
              m        = if r < 0 then n - 1 else n + 1
              s = signum (abs r - 0.5)
            in if (s == -1) || (s == 0 && even n)
               then n
               else m

  ceiling x = if r > 0 then n + 1 else n
             where (n, r) = properFraction x
  floor x = if r < 0 then n - 1 else n
           where (n, r) = properFraction x
```

Numeric Functions はそろっている。

RealFloat クラスを定義して、Double, Float をそのインスタンスにすれば、ひとまず完成かな。(Ratio は別チケット 108 なので)

2021-10-31 (Sun)

RealFloat のインスタンス Float, Double の定義を完了。これの疎通試験を sample333.hs としてテストに追加。

この issue はこれでクローズ。