MyFloat

13307130178 王政和

Quick Start

```
➜  ~ #use comma for statistic
➜  ~ python MyFloat.py 5.0 3
0_0000100_0_1000000,8.000000,8.000000,0.000000
0_0000010_0_1000000,2.000000,2.000000,0.000000
0_0000100_0_1111000,15.000000,15.000000,0.000000
0_0000001_0_1101010,1.656250,1.666667,0.006250
```

Usage

```
➜  ~ python
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from MyFloat import *
>>> MyFloat(23.33)
0_0000101_0_1011101
>>> a = MyFloat(0.3)
>>> b = MyFloat(0.02)
>>> a.decode()
38.0
>>> a + b
1_0000001_0_1010001
>>> (a*b).decode()
0.75
>>> a.m
'1001100'
>>> a.e
'0000001'
>>> b.mv()
81
>>> b.ev()
-5
```

Illustration

Summary

- `MyFloat` 为核心类，从一个 `float` 构造出一个 `16-bit float`
- 重载了四则运算符
- 乘法采用了 `Trounding` 策略

- 规格化、舍入、报错等操作均按照讲义要求实现

Details

- scale(m, e)
  - 接受带符号的尾数和阶，返回一个 `MyFloat`
- parseFloat(float)
  - 把 `float` 转换为 `MyFloat`
- main(x, y)
  - 计算xy的四则运算并比较结果，输出
- fmtPrint(c, res)
  - 美化输出，为了方便统计采用 `csv` 格式
- 其他函数功能见各函数的 `__doc__`

Test

- 测试脚本为 `randomtest.py`

```
➜  ~ time python randomtest.py
cur error 0
cur error 1
cur error 2
python randomtest.py  6.73s user 1.54s system 77% cpu 10.631 total
```

- 跑了1000组随机计算，均为[-500,500)浮点数，有3次 `Error` ，这里忽略
- 之后用 `Excel` 统计了一下，各运算平均误差如下：

+	-	*	/
0.018126836	0.014546672	0.014221392	0.008597601

- 结果比较满意