# 任务描述

A5，A13的过去半小时数据的前10%和后10%的数据作为ceiling和floor

# 类的设计

## UpperLow

|  |
| --- |
| def computeUpLow(self):  若self.data\_queue中已经存满数据，则根据self.upp\_pro和self.low\_pro计算ceiling和floor  若self.data\_queue中有数据但未满，则根据当前数据排序计算ceiling和floor |

Self.upp\_pro ceiling的比例

Self.low\_pro floor的比例s

## UpperLowControl

|  |
| --- |
| def \_\_init\_\_(self, flux, work\_period, interval, upp\_pro, low\_por, computeKind):  对computeKind形参解析，根据解析后的参数，调用不同的getReadyFor函数  初始化各种类对象 |
| def getData(self,cur\_ts):  根据self.computeArg1和self.computeArg2，获取不同的数据 |
| def getPingJiaContract(self,cur\_ts):  获取当前平价合约的Latest值 |

computeKind 参数，比如“A5 0”代表当月，1代表下月，2代表下季，3代表隔季

# 输入

|  |  |  |  |
| --- | --- | --- | --- |
| 键值 | 数值 | Hash键 | 单位 |
| MDLD:cur\_ts:A13:[IC01-IH04] | -3840、-3838、-3850、46244.9、46246.9 | B、S、L、C、R | 十\*点 |
| MDLD:cur\_ts:A5:[1908M02600-2003M03400] | 18、34、17、28130 | LATEST、BP1、SP1、Pe | 十\*点 |

# 输出

|  |  |  |  |
| --- | --- | --- | --- |
| 键值 | 数值 | Hash键 | 单位 |
| MDLD:cur\_ts:A13:[IC01-IH04] | -626.0 |  | 十\*点 |
| MDLD:cur\_ts:A5:[PM|NM|NS|SS]:PJ | -109.0 | A30mUpper  A30mLow | 十\*点 |

# 工作时间设置须知：

时间要多设置一秒

[["09:30:01","11:30:01"],["13:00:01","15:00:01"]]

因为 计算的是前一秒的数据

# 添加数据项须知：

首先

解析computeKind形参，获得参数，根据获得的参数，调用不同的准备函数sgetReadyFor(), getReadyFor函数要对self.writeFlux\_key赋值

之后

更改getData函数，根据参数写新的getDataFor函数

# 程序代码

RealTimeData/UpperLower2VVV.py