شرکت مشاور سرمایه گذاری
ارزش پرداز آریان
تحت نظارت سازمان بورس و اوراق بهادار

QuantoRythm

مستندات استفاده از سامانه QuantoRythm برای
«مسابقه معاملات الگوریتمی بازار سرمایه – ۱۵ دی تا ۱۷ بهمن ۱۳۹۷»

ویرایش ۱- ۱۰ دی ۱۳۹۷

شما با استفاده از سامانه QuantoRythm می‌توانید کدهای نوشته شده خود را وارد سامانه نمایید تا به صورت مستمر و مداوم اجرا شود.
کدهای اصلی در متد DecidePrimitive نوشته می‌شوند که توابع قابل استفاده و متغیرهای در دسترس در ادامه توضیح داده شده است. همچنین سایر توابع قابل استفاده نیز در همین مستند توضیح داده شده است.
محتوای کلاس‌های استفاد شده نیز در همین مستند توضیح داده شده است.

1. `protected override object DecidePrimitive(AlgorithmInputEvents inputType, out Exception exception)`

تصمیم‌گیری برای خرید و فروش در این متد انجام می‌گیرد که به صورت منظم و مستمر اجرا می‌شود

2. `protected override void CustomActionPrimitive(string actionName, UserInfo userInfo, out Exception exception)`

اجرا دستورات سفارشی کاربر

3. `protected override void StartPrimitive(string userName, string userFirstName, string userLastName, string OriginName)`

در هنگام شروع الگوریتم یک بار اجرا می‌شود

4. `protected override void StopPrimitive(string userName, string userFirstName, string userLastName, string OriginName)`

در هنگام توقف الگوریتم اجرا می‌شود

5. `MarketStatus LastMarketStatus`

وضعیت بازار (باز/بسته/پیش‌گشایش)

6. `bool isDeciding`

نشان دهنده مشغول بودن الگوریتم به تصمیم‌گیری مرحله قبل

7. `bool isEnabled`

نشان‌دهنده فعال بودن الگوریتم

8. `DateTime CurrentTime`

زمان جاری

9. `OmsClientData GetOmsData(string accountCode)`

دریافت اطلاعات محتوای حساب کارگزاری

10. `CustomerBrokerInfo GetAccount(string accountCode)`

دریافت اطلاعات حساب

11. `TseTmcMarketData GetMarketData(string instrumentId)`

دریافت داده بازار

12. `List<Candle> GetCandles(string instrumentId, TimeFrameEnum timeFrame, int period, DateTime date)`

دریافت داده‌های شمعی معاملات

13. `void SendMyOrderNow(int price, int qty, string ISINCode, CustomerBrokerInfo myCbi, OrderSide sd)`

ارسال سفارش خرید و فروش

14. `void SendNotificationImmediatly(string title, string text, LogLevel logLevel)`

ارسال پیغام فوری

15. `void SendNotificationPatiently(string messageCategory, string title,string text, LogLevel logLevel = LogLevel.Info,int minimumMessageVerbosityTimeSeconds = 600)`

ارسال پیغام غیرفوری

16. `void UpdateParameterValueByTitleEn(string titleEn, object value)`

ذخیره پارامترهای تغییریافته

نمونه کد

```
//الگوریتم اصلی در قسمت DecidePrimitive نوشته می‌شود که این قسمت از کد به صورت منظم و مستمر اجرا خواهد شد


//در اینجا می‌توانید متغیرهایی که لازم دارید تعریف کنید.
private int myValue = 0;

//این متد وقتی که الگوریتم شما شروع شود صدا زده می‌شود
//شما به اطلاعات کاربری که فرمان شروع را صادر کرده است دسترسی دارید
protected override void StartPrimitive(string userName, string userFirstName, string
userLastName, string OriginName)
{
    //  از این متد برای ارسال پیغام به داشبورد مانیتورینگ استفاده کنید
    //  در این حالت به جای ارسال پیغام‌های مکرر، تنها یک پیغام در بازه‌ای که ده دقیقه‌ای ارسال می‌شود
    this.SendNotificationPatiently("MessageCategory", "MessageTitle", "MessageText");

    //  اگر پارامتری برای الگوریتم در نظر گرفته اید و آن را در بخش تنظیمات تعریف کرده اید،
    //  با استفاده از این متد می‌توانید آخرین مقدار ذخیره شده آن را بازیابی کنید.
    //  نوع پارامتری که ذکر می‌کنید باید با نوع تعریف شده در تنظیمات برابر باشد
    this.myValue = this.GetParameterObjectValueByTitleEn<int>("myParameter");
}

//این تابع وقتی الگوریتم شما متوقف شود صدا زده می‌شود
//شما به اطلاعات کاربری که فرمان توقف را صادر کرده است دسترسی دارید
protected override void StopPrimitive(string userName, string userFirstName, string
userLastName, string OriginName)
{

    //  با استفاده از این متد می‌توانید مقدار پارامترها را ذخیره کنید.
    //  پارامترهای مهم را بعد از تغییر ذخیره کنید.
    this.UpdateParameterValueByTitleEn("myParameter", this.myValue);
}

//این متد با زدن دکمه‌های اختصاصی ای که برای الگوریتم در بخش تنظیمات ایجاد کرده‌اید فراخوانی می‌شود.
//نام دکمه‌ای که فشار داده شده است بر اساس نامی  که در بخش تنظیمات در نظر گرفته‌اید
//در اینجا در دسترس است
protected override void CustomActionPrimitive(string actionName, UserInfo userInfo, out
Exception exception)
{
    exception = null;
}

//این متد در به طور مستمر و منظم وقتی که الگوریتم شما در حال اجرا باشد صدا زده می‌شود.
//تا زمانی که اجرای قبلی تمام نشود، برای بار بعدی صدا زده نخواهد شد.
protected override object DecidePrimitive(AlgorithmInputEvents inputType, out Exception
exception)
{
    //  اگر خطایی را گزارش می‌کنید در این متغیر قرار دهید.
    //  در غیر اینصورت آن را به همین شکل رها کنید
    exception = null;
```

```
        //      برای دریافت اطلاعات وضعیت حساب کارگزاری) پورتفوی، سفارشات، مانده‌ها (از این متد استفاده کنید.
    var omsData = this.GetOmsData("123_12");

        //      برای دریافت اطلاعات مانده حساب کارگزاری از این متد استفاده کنید.
    var myAccount = this.GetAccount("123_12");

        //      برای دریافت اطلاعات آخرین وضعیت معاملات یک نماد از این متد استفاده نمایید
    var marketdata = this.GetMarketData("IRO3ZOBZ0001");

    var change = marketdata.PClosing / marketdata.PriceYesterday;

        //      برای بررسی اینکه آیا برای این نماد در این حساب کارگزای
        //      سفارش ارسال نشده‌ای وجود دارد که هنوز تکلیف آن مشخص نشده است
    var hasPendingOrder = omsData.HasPendingOrderFor("IRO3ZOBZ0001");

        //      برای بررسی اینکه آیا برای این نماد در این حساب کارگزاری
        //      سفارش باز وجود دارد؟
    var hasOpenorder = omsData.HasOpenOrderFor("IRO3ZOBZ0001");

        //      حجم معاملات انجام شده برای این نماد
        //      در این حساب کارگزاری
    var totalExecuted = omsData.SumExecutedTradeFor("IRO3ZOBZ0001");

        //      برای دسترسی به بهترین سفارشات خرید و فروش بازار
                                                    // Qoutes[0~2]
                                                //BestBuyers      خریداران
                                                //BestSellers     فروشندگان
                                                // Price     قیمت Quantity / تعداد Volume / حجم
    var buyerPrice = marketdata.BestBuyers.Qoutes[0].Price;
    var dummy2 = marketdata.BestBuyers.Qoutes[1].Quantity;
    var dummy3 = marketdata.BestSellers.Qoutes[2].Volume;

    var myBuyPrice = buyerPrice - 1;

    var qtyForFirstAndSecondRow = marketdata.BestSellers.Qoutes[0].Quantity +
marketdata.BestSellers.Qoutes[1].Quantity;
                                                    // TradesVol:     حجم معاملات بازار در این نماد
    var totalTrades = marketdata.TradesVol;

        //      دریافت داده‌های شمعی برای یک نماد
    var candles = this.GetCandles("IRO3ZOBZ0001", TimeFrameEnum.D01, 1, DateTime.Now);

        //      اندیکاتورهای تکنیکال

    int outBegIdx;
    int outNbElement;

        //      محاسبات بر مبنای قیمت پایانی هر شمع
    var data = candles.Select(x => (double)x.Close).ToArray();

    var output = new double[data.Length];

        //      میانگین متحرک ساده
```

```
    var TIresult = TI.MovingAverage(2, 4, data, 2, TI.MAType.Sma, out outBegIdx, out
outNbElement, output);

    var sma = (int)output[0];

                                                    //      سایر اندیکاتورها، همانند کتابخانه TA-LIB و با همان ساختار پیاده سازی شده اند.

                                                    //      مانده قابل برداشت از حساب کارگزاری
    var amount = Math.Round(omsData.CreditInfo.Withdrawable * 0.8);

    int qty = (int)Math.Round(amount / myBuyPrice);

                                                    //      بررسی شرط مورد نظر برای خرید یا فروش
    if (!hasOpenorder && !hasPendingOrder && totalExecuted < 2500 &&
        marketdata.PClosing > 1180 &&
        totalTrades > 1500000000 && qtyForFirstAndSecondRow > 320000000 && sma == 1800)
    {
                //      ارسال سفارش فروش برای نماد مورد نظر به حجم و قیمت تعیین شده و از طریق کارگزاری معرفی شده
        this.SendMyOrderNow(myBuyPrice, qty, "IRO3ZOBZ0001", myAccount, OrderSide.Sell);
    }

    return null;}

                                                    //متدهای دیگر مورد نیاز خود را نیز می‌تواند بنویسید
private int myCalulations(int input)
{
    return input * 2;
}
```

کلاس‌های استفاد شده

```
OmsClientData:
        DateTime RequestTime
        List<PortfolioItem> Portfolio
        List<Order> OpenOrders
        List<Order> PendingOrders
        List<Order> Orders
        List<Trade> Trades
        CreditInfo CreditInfo
        List<CustomerAccount> Accounting
        List<Order> InternalOrders
        List<Order> ExternalOrders
        LoginStatus loginStatus

PortfolioItem:
        SymbolInfo Symbol               // کلاس اطلاعات نماد
        long Quantity                   // مانده

SymbolInfo:
        string Symbol                   //نماد
        string Name                     // نام
        string ISINCode                 // ISIN Code
        string InsCode                  // InsCode (TseTmc)

Order:
        string InstrumentId
        DateTime OrderDateTime
        int Price                       // قیمت
        string OrderIdOms               // شناسه سفارش
        string OrderIdInternal          // شناسه داخلی سفارش
        OrderValidity OrderValidity// سفارش اعتبار: معاملات جلسه/حذف و انجام/تاریخ تا معتبر/لغو تا معتبر/روز
        string OrderValidityDate
        OrderSide OrderSide             // نوع سفارش:خرید/فروش
        int Quantity                    // حجم سفارش
        int QuantityMinimum
        int QuantitiyDisplayed
        int QuantityExecuted            // حجم معامله شده از سفارش
        OrderStatus OrderStatus         // سفارش وضعیت: حذف سیستم از سفارش /شده انجام کامل صورت به سفارش
        معاملات سیستم در ثبت/شده
        OrderType OrderType

CreditInfo:
        long Withdrawable               // مبلغ قابل برداشت
        long BalanceBlocked             // مبلغ بلوکه شده
        long TotalBalance               // جمع کل

Trade:
        string TradeOmsId               //شناسه معامله
        string TradeInternalId          //شناسه داخلی معامله
        int Vol                 //حجم معامله
        DateTime TradeTime              //زمان معامله
        string InstrumentId             //شناسه نماد
```

```
OrderSide OrderSide          //نوع معامله (خرید/فروش)
int Price                    //قیمت معامله
string OrderOmsId            //شناسه سفارش
```