

カブロボ作成ルール/仕様 Java プログラマー用補足資料

本ドキュメントはカブロボルール、カブロボ作成ルールを補足するものです。特に、カブロボSDKを使用してJava言語でプログラミングする時のルール及び仕様について補足解説します。

1. 売買注文

カブロボの注文思考時間

カブロボは前場の前(8時~9時)と後場の前(11時半~12時半)の2回、それぞれ前場・後場の注文に向けて思考する時間が与えられます。思考時間はそれぞれとも3分が上限となっております。具体的には、ロボットクラス(AbstractRobot 抽象クラスを実装したクラス)の order メソッドが実行されます。各場合とも、その時点までのデータが取得可能となっております。前場前では前日までのデータ(海外指標データを含む)が取得できます。後場前では、前場前のデータに加えて前場中のデータも含まれることとなります。実際には、それぞれのカブロボの起動時間は異なりますが、カブロボに提供するデータと時間は全て同じです。例えば、前場の前では、8時から9時の間のいずれの時間にカブロボは起動しますが、取得できるデータは8時までのデータであり、8時に起動したことをカブロボには通知します。

カブロボの翌営業日に向けた思考時間(スクリーニング等)

カブロボは翌営業日の注文に向けて、スクリーニング処理をすることで銘柄の絞込みを行なうことができます。スクリーニング処理は、後場後(16時~24時)の最大8分の間に行ないます。スクリーニング後の銘柄を監視銘柄と呼び、翌営業日の注文では、この監視銘柄の中から注文することができます。具体的には、ロボットクラス(AbstractRobot 抽象クラスを実装したクラス)の screening メソッドが実行されます(order メソッドは呼びません)。よって、order メソッドだけではなく、別に screening メソッドも実装する必要があります。その時点までのデータが取得可能ですが、その日の海外指標データに関しては、翌日にならないと取得できないため、含まれません。翌営業日の最初の注文時から取得できます。上記のカブロボの注文時と同じように、それぞれのカブロボに提供するデータおよび時間は同じであり、この場合、16時までのデータを提供し、16時にスクリーニングが開始されたことをカブロボに通知します。

注文量

注文量は、買いの場合は正の数量を指定し、売りの場合は負の数量を指定してください。ポートフォリオにおいても、現物保有株は正の数量、信用売り銘柄は負の数量で扱われます。

コーポレートアクション

指定値(デフォルトは 20%)以上の含み損がある保有銘柄や、上場廃止・管理ポスト・整理ポストなどのマイナス要素があるコーポレートアクションが予定された銘柄などをシステム側が強制反対売買する際には、下記に述べる CustomerTradeId が 9999 に設定されます。また、上場廃止・管理ポスト・整理ポストなどのマイナス要素があるコーポレートアクションが予定された銘柄の取り扱い中止日は Stock#getDisableDate()のメソッドで取得できます。なお、該当銘柄の新規注文はできません。

銘柄組入比率

新規注文における銘柄組入比率上限値を設定することができます。デフォルトでは 10%になっており、robot-config.xml により、設定できるようになっています。

注文付加情報

注文をする際にユーザーの自由な ID をメモ代わりに振ることができます。OrderManager の setLastOrderCustomerId(Integer id)メソッドを利用することで、注文リストの最後の注文に ID を付与することができます。これは OrderHistory 中の CustomerTradeId と同一になります。また、setLastOrderReason(String reason)メソッドを利用する事で、注文リストの最後の注文に注文理由を付与する

ことができます。OrderHistoryManager から過去の注文履歴を参照し、OrderHistory#getReason()で取得できます。

条件付特別注文

OrderManager#setLastOrderSpecialControl() メソッドで、注文リストの最後の注文に高度な発注条件(トリガー機能と注文価格修正指示機能)を付加することができます。Portfolio オブジェクトから行った反対売買の注文に対しても使用することができます。

新規買い指値注文以外では、注文価格修正指示を付加すると条件付加前の注文価格は無視されます。新規の買い指値注文では、取引余力が増えずに済む場合のみ、注文価格修正が成功します。

新規買い指値注文のみ

修正後価格 > 条件付加前の注文価格 ⇒ 注文失効 (約定しません)

修正後価格 ≤ 条件付加前の注文価格 ⇒ 発注成功 (約定判定が行われます)

※ 条件付特別注文を使用する場合、robot-config.xml 及び robot-profile.xml の<special-order>タグの設定が必要です。設定方法につきましては robot-config.html をご覧ください。

取引 ID

新規注文取引毎に TradeID という識別 ID が振られます。1取引とは、ある新規注文をしてそれが全て反対売買を終えるまでのことを指します。そして、同一銘柄であっても、取引が異なればそれらがまとめられることはなく、別取引として扱われます。反対売買は、保有しているポートフォリオの TradeID を指定することでできます。Portfolio クラスから直接反対売買を行うと ID 自体を意識する必要がありません。

未約定注文

後場前において前場に注文をして約定したものは OrderHistory から取得できますが、約定しなかったものについては NotExecutedOrder から取得することになります。後場時に約定をしなかったものはシステム自動キャンセルとなるため、スクリーニング時には OrderHistory から取得できます。

2. キャンセルについて

反対売買時において、前場に出した注文をキャンセル後ならば、後場に条件を変えて出すことができます。損切りや利益確定の際にご利用できます。

3. 時間制限

制限時間をオーバーして途中で終了した場合、(AbstractRobot クラスの order や screening メソッドを返せずに終了した場合)、それらの情報は保持されません。カブロボの動作中は、注文内容や MemoManager クラスで保存したメモの情報はメモリ上に保存され、終了時にデータベースに書き込まれるため、終了出来なかった場合の情報は破棄されることとなります。次のセッションで演算を続きから再開することもできません。

付属の SampleRobot をカブロボサーバー環境において、300銘柄で1年間実行した時間を末尾に添付しております。ご参考下さい。データベース等システムが異なるため、カブロボサーバー上ではより高速に動作します。

4. 成績表

AbstractRobot の terminate メソッドをオーバーライドすることで、セルフテスト時の最終日後に成績表を表示させることができます。勝率や利益率などから自分なりの評価を出す場合に利用できます。

5. ログ出力

RobotLogManager の log メソッドを利用することで、ログをファイルに書き出すことやデータベースにいれることができます。資産評価額やポートフォリオなどの表示ログはコンフィグファイルによって出力可否が制御できます。標準出力(System.out.println)はログに出力されることはありません。また、出力されたログにつきましてはコンテスト時にメールの本文に載せてお知らせする予定です。

6. Javaでプログラミングしたカブロボの提出方法

カブロボサイト(kaburobo.jp)に直接Web経由でアップロードして提出します。その際に、カブロボの設定情報を記述した robot-profile.xml を同時にアップロードします。これにはクラス名・エンコード・銘柄組入比率上限値・空売りの有無などがあり、サーバー上でのカブロボの挙動に関わるので記述をお願いいたします。

7. プログラミング規約

オブジェクトメモの容量は 5MB までとします。正確には、5242880 バイトまでです。オブジェクトのサイズは MemoManager から計算することができます。パッケージの使用は許可します。複数ファイルでも問題ありません。

コードの容量は、展開した状態(コンパイルせず、zip にも固めてない状態)で 10MB までとします。正確には、10485760 バイトまでです。

- ・ 以下に羅列したクラス及びパッケージ、カブロボ SDK 内で Deprecated になっているメソッドは使用不可
- ・ java.lang.Object の finalize メソッドのオーバーライドは不可、また finalize と同名のメソッドも不可
- ・ System の static import は不可
- ・ 下記のクラス及びパッケージと同一名のクラスやパッケージの使用不可

```
java.applet
java.awt
java.beans
java.beancontext
java.io(
    Serializable 以外
)
java.lang(
    ClassLoader
    Compiler
    InheritableThreadLocal
    Process
    Runtime
    RuntimePermission
    SecurityManager
    System[
        exit()
        getenv()
```

```
    getProperties()
    getProperty()
    getSecurityManager()
    load()
    loadLibrary()
    mapLibraryName()
    setProperties()
    setProperty()
    setSecurityManager()
]
Thread
ThreadGroup
ThreadLocal
)
java.net
java.nio.channels
java.nio.channels.spi
java.rmi
java.security
java.sql
java.util(
    PropertyResourceBundle
    ResourceBundle
    Timer
    TimerTask
)
java.util.jar
java.util.logging
java.util.prefs
java.util.zip
```

■ 変更履歴

2008/07/30 version 2.1.1

【追加】プログラミング規約に、finalize メソッドについての制限を追加

2008/04/14 version 2.1.0

【変更】自動損切りの基準値を設定可能にしたため、関連する記述を修正

2007/05/22 version 2.0.0

【変更】説明文を一部修正

2007/04/03 version 2.0.0 rc1

【変更】KB(Kaburobo Builder)との共通ルール部分については、別途「カブロボ作成ルール」ファイルに移行

【追加】条件付特別注文を追加

2007/03/13 version 2.0.0 beta2

【追加】上場直後の銘柄に関する約定ルールを追加

【追加】カブロボ自主規制による注文制限のまとめを追加

2007/03/01 version 2.0.0 beta1

【追加】提出方法についての項目を追加

【追加】新規注文時の銘柄組入比率上限値を設定する機能を追加

【追加】各注文に対して注文理由を付与する機能を追加

【変更】逆指値注文の必要取引余力を指値注文と同等の計算方法に変更

【変更】銘柄毎の注文量の上限を新規売買注文は過去5日間の平均出来高の2パーセント、反対売買注文は3パーセントに設定

【変更】カブロボサーバーにおける実行時間の値を更新

【変更】最終成績表についての記載を追加

【変更】約定ルールを追加