

Руководство пользователя

Acceleration

LUA

Создание стратегий

СОДЕРЖАНИЕ

1	Введение	4
2	Структура стратегии	5
2.1	Таблицы и функции стратегии	5
2.2	Минимальный код стратегии	7
2.3	Функции обратного вызова	7
3	Функции программы	9
3.1	Изменение свойств визуальных компонент VCL	9
3.2	Получение параметров инструмента	9
3.3	Функции сообщений/вызова всплывающих окон	10
3.4	Функции работы со временем	11
3.5	Функции работы со строками	12
3.6	Функции работы с таблицами	14
3.7	Математические функции	14
3.8	Функции кодировки	16
3.9	Функции работы с файлами	17
3.10	Функции получения торговых параметров	18
3.11	Функции получения данных графиков	21
3.12	Функции получения данных стакана котировок	21
3.13	Получение параметров стоп-заявок	23
3.14	Функции для работы с флагами заявок/стоп-заявок	24
3.15	Функции получения цены позиции	25
3.16	Функции получения финансового результата	26
3.17	Функции работы со стратегиями	27
3.18	Транзакции торговых операций	30
3.19	Алгоритмические заявки	34
4	Параметры инструмента	36
4.1	Настройки торговли	36
4.2	Торговые параметры	41

ПЕРЕЧЕНЬ ТАБЛИЦ

Табл. 2.1 – Таблицы и функции стратегии	5
Табл. 2.2 – Функции обратного вызова	7
Табл. 3.1 – Изменение свойств визуальных компонент VCL	9
Табл. 3.2 – Получение параметров инструмента	9
Табл. 3.3 – Функции сообщений/вызова всплывающих окон	10
Табл. 3.4 – Функции работы со временем	11
Табл. 3.5 – Функции работы со строками	12
Табл. 3.6 – Функции работы с таблицами	14
Табл. 3.7 – Математические функции	14
Табл. 3.8 – Функции кодировки	16
Табл. 3.9 – Функции работы с файлами	17
Табл. 3.10 – Функции получения торговых параметров	18
Табл. 3.11 – Функции получения данных графиков	21
Табл. 3.12 – Функции получения данных стакана котировок	21
Табл. 3.13 – Получение параметров стоп-заявок	23
Табл. 3.14 – Функции для работы с флагами заявок/стоп-заявок	24
Табл. 3.15 – Функции получения цены позиции	25
Табл. 3.16 – Функции получения финансового результата	26
Табл. 3.17 – Функции работы со стратегиями	27
Табл. 3.18 – Транзакции торговых операций	30
Табл. 3.18 – Алгоритмические заявки	34
Табл. 4.1 – Таблица настроек торговли инструмента	36
Табл. 4.2 – Таблица торговых параметров инструмента	41

1 ВВЕДЕНИЕ

Стратегии программы представляют собой файлы ***.lua** или ***.luac**, размещенные в папке **strategies**. Стратегия должна содержать описание настраиваемых полей через интерфейс программы и возвращать сигнал создания заявок. Заявки создаются по сигналам стратегии в модуле стратегий, либо напрямую из стратегии.

Загрузка стратегий в программу осуществляется в последнюю очередь после загрузки всех модулей.

С целью уменьшения повторяющегося кода можно использовать свои модули, размещенные в папке **custom**. Загрузка модулей из **custom** осуществляется перед загрузкой стратегий после загрузки всех модулей.

Визуальная часть программы построена с использованием библиотеки **VCLua** версии 0.6.3. Для создания настраиваемых полей стратегии также используется данная библиотека.

Другие доступные модули:

- **cjson**
- **lfs**
- **luacom**
- **socket**
- **w32**

2 СТРУКТУРА СТРАТЕГИИ

2.1 Таблицы и функции стратегии

Для стратегий в программе выделена отдельная таблица **iS.mts**

Стратегия должна представлять собой таблицу, включающую в себя набор обязательных таблиц и функций, описанных в Табл. 2.1.

Табл. 2.1 – Таблицы и функции стратегии

№	Параметр	Тип	Обяз.	Описание																																				
1.	info	table		Содержит перечисление полей, описывающих свойства стратегии: <table><tr><th>Поле</th><th>Тип</th><th>Обяз.</th><th>Описание</th></tr><tr><td>version</td><td>string</td><td></td><td>версия, выводится в окне Настроек торговли</td></tr><tr><td>title</td><td>string</td><td></td><td>заголовок, выводится в таблице состояний</td></tr><tr><td>description</td><td>string</td><td></td><td>описание</td></tr><tr><td>author</td><td>string</td><td></td><td>автор</td></tr><tr><td>copyright</td><td>string</td><td></td><td>авторские права</td></tr><tr><td>datebuild</td><td>string</td><td></td><td>дата последнего изменения</td></tr><tr><td>isEditor</td><td>boolean</td><td></td><td>Признак стратегии, созданной в Редакторе стратегий</td></tr><tr><td>isRemoved</td><td>boolean</td><td></td><td>Признак удаленной стратегии</td></tr></table>	Поле	Тип	Обяз.	Описание	version	string		версия, выводится в окне Настроек торговли	title	string		заголовок, выводится в таблице состояний	description	string		описание	author	string		автор	copyright	string		авторские права	datebuild	string		дата последнего изменения	isEditor	boolean		Признак стратегии, созданной в Редакторе стратегий	isRemoved	boolean		Признак удаленной стратегии
Поле	Тип	Обяз.	Описание																																					
version	string		версия, выводится в окне Настроек торговли																																					
title	string		заголовок, выводится в таблице состояний																																					
description	string		описание																																					
author	string		автор																																					
copyright	string		авторские права																																					
datebuild	string		дата последнего изменения																																					
isEditor	boolean		Признак стратегии, созданной в Редакторе стратегий																																					
isRemoved	boolean		Признак удаленной стратегии																																					
2.	ctrls	table		Содержит список графических элементов, отображаемых в окне Настройки торговли на закладке МТС.																																				
3.	conditions	table		Содержит перечисление условий для Редактора стратегий: <table><tr><th>Поле</th><th>Тип</th><th>Обяз.</th><th>Описание</th></tr><tr><td>openLong</td><td>string</td><td></td><td>условия открытия Лонга</td></tr><tr><td>closeLong</td><td>string</td><td></td><td>условия закрытия Лонга</td></tr><tr><td>openShort</td><td>string</td><td></td><td>условия открытия Шорта</td></tr><tr><td>closeShort</td><td>string</td><td></td><td>условия закрытия Шорта</td></tr></table>	Поле	Тип	Обяз.	Описание	openLong	string		условия открытия Лонга	closeLong	string		условия закрытия Лонга	openShort	string		условия открытия Шорта	closeShort	string		условия закрытия Шорта																
Поле	Тип	Обяз.	Описание																																					
openLong	string		условия открытия Лонга																																					
closeLong	string		условия закрытия Лонга																																					
openShort	string		условия открытия Шорта																																					
closeShort	string		условия закрытия Шорта																																					
4.	indicators	table		Содержит список используемых стратегий индикаторов. Каждый индикатор представляет собой таблицу с полями: <table><tr><th>Поле</th><th>Тип</th><th>Обяз.</th><th>Описание</th></tr><tr><td>identifier</td><td>string</td><td>Да</td><td>Идентификатор графика индикатора</td></tr><tr><td>candle_count</td><td>number</td><td>Да</td><td>Число получаемых свечей индикатора</td></tr><tr><td>is_price_chart</td><td>boolean</td><td></td><td>Признак графика цены, используется</td></tr></table>	Поле	Тип	Обяз.	Описание	identifier	string	Да	Идентификатор графика индикатора	candle_count	number	Да	Число получаемых свечей индикатора	is_price_chart	boolean		Признак графика цены, используется																				
Поле	Тип	Обяз.	Описание																																					
identifier	string	Да	Идентификатор графика индикатора																																					
candle_count	number	Да	Число получаемых свечей индикатора																																					
is_price_chart	boolean		Признак графика цены, используется																																					

№	Параметр	Тип	Обяз.	Описание				
				<table><tr><td></td><td></td><td></td><td>для определения временного интервала</td></tr></table>				для определения временного интервала
			для определения временного интервала					
5.	createCtrls	function		createCtrls (VCL group) group – группа Параметры Функция создания графических элементов в группе Параметры окна Настройки торговли. function iS.mts.My_Strategy.createCtrls(group) iS.mts.My_Strategy.ctrls.lbl = VCL.Label(group, {Top=5, Left=5, Caption="Тест"}) end				
6.	init	function		init (number index) index – индекс инструмента Функция, запускаемая при первом старте стратегии				
7.	stop	function		stop (number index) index – индекс инструмента Функция, запускаемая после остановки стратегии				
8.	main	function	Да	main (number index) index – индекс инструмента Функция, возвращающая сигнал стратегии: • number open_signal – 0: Нет сигнала, 1: Лонг, 2: Шорт; • boolean close_long – true: Закрыть Лонг; • boolean close_short – true: Закрыть Шорт. function iS.mts.My_Strategy.main(index) return 0, false, false end				

2.2 Минимальный код стратегии

Минимально возможный код стратегии приведен ниже.

```
-- Атрибуты стратегии
iS.mts.minimal = {}
-- Стратегия
function iS.mts.minimal.main(index)
    -- open_signal = 0: Нет сигнала, 1: Лонг, 2: Шорт
    -- close_long = true/false: Заккрытие Лонга
    -- close_short = true/false: Заккрытие Шорта
    return 0, false, false
end
```

2.3 Функции обратного вызова

Для использования функций обратного вызова QUIK необходимо определить их в таблице стратегии. Функции стратегии содержат те же входные параметры, что и функции QUIK, с добавлением в начале индекса инструмента.

```
-- Функция вызывается терминалом QUIK при изменении текущих параметров
function iS.mts.minimal.OnParam(index, class_code, sec_code)
    local last_price = tonumber(getParamEx(class_code, sec_code, "last").param_value)
    message("last price " .. sec_code .. " = " .. tostring(last_price), 1)
end
```

Доступны функции обратного вызова, перечисленные в Табл. 2.2.

Табл. 2.2 – Функции обратного вызова

№	Функция	Описание
1.	OnConnected(index, flag)	Функция вызывается терминалом QUIK при установлении связи с сервером QUIK и получении терминалом описания хотя бы одного класса
2.	OnDisconnected(index)	Функция вызывается терминалом QUIK при отключении от сервера QUIK
3.	OnAccountBalance(index, tab)	Функция вызывается терминалом QUIK при получении изменений текущей позиции по счету
4.	OnAccountPosition(index, tab)	Функция вызывается терминалом QUIK при изменении денежной позиции по счету
5.	OnAllTrade(index, tab)	Функция вызывается терминалом QUIK при получении обезличенной сделки
6.	OnCleanUp(index)	Функция вызывается терминалом QUIK при смене сессии
7.	OnClose(index)	Функция вызывается перед закрытием терминала QUIK и при выгрузке файла qlua.dll
8.	OnDepoLimit(index, tab)	Функция вызывается терминалом QUIK при получении изменений лимита по бумагам
9.	OnDepoLimitDelete(index, tab)	Функция вызывается терминалом QUIK при удалении клиентского лимита по бумагам

№	Функция	Описание
10.	OnFirm(index, tab)	Функция вызывается терминалом QUIK при получении описания новой фирмы от сервера
11.	OnFuturesClientHolding(index, tab)	Функция вызывается терминалом QUIK при изменении позиции по срочному рынку
12.	OnFuturesLimitChange(index, tab)	Функция вызывается терминалом QUIK при получении изменений ограничений по срочному рынку
13.	OnFuturesLimitDelete(index, tab)	Функция вызывается терминалом QUIK при удалении лимита по срочному рынку
14.	OnMoneyLimit(index, tab)	Функция вызывается терминалом QUIK при получении изменений по денежному лимиту клиента
15.	OnMoneyLimitDelete(index, tab)	Функция вызывается терминалом QUIK при удалении денежного лимита
16.	OnNegDeal(index, tab)	Функция вызывается терминалом QUIK при получении внебиржевой заявки
17.	OnNegTrade(index, tab)	Функция вызывается терминалом QUIK при получении сделки для исполнения
18.	OnOrder(index, tab)	Функция вызывается терминалом QUIK при получении новой заявки или при изменении параметров существующей заявки
19.	OnParam(index, sec_code, class_code)	Функция вызывается терминалом QUIK при изменении текущих параметров
20.	OnQuote(index, sec_code, class_code)	Функция вызывается терминалом QUIK при получении изменения стакана котировок
21.	OnStop(index, s)	Функция вызывается терминалом QUIK при остановке скрипта из диалога управления и при закрытии терминала QUIK
22.	OnStopOrder(index, tab)	Функция вызывается терминалом QUIK при получении новой стоп-заявки или при изменении параметров существующей стоп-заявки
23.	OnTrade(index, tab)	Функция вызывается терминалом QUIK при получении сделки
24.	OnTransReply(index, tab)	Функция вызывается терминалом QUIK при получении ответа на транзакцию пользователя

3 ФУНКЦИИ ПРОГРАММЫ

Указатели на общедоступные функции программы перечислены в таблице iS.f

```
-- Стратегия
function iS.mts.minimal.main(index)
-- Получение числа инструментов в настройках
local sec_count = iS.f.getSecCount()

return 0, false, false
end
```

3.1 Изменение свойств визуальных компонент VCL

Табл. 3.1 – Изменение свойств визуальных компонент VCL

№	Функции
1.	-- Заполнение списка компоненты VCL SetListItems (sender, tab) -- АРГУМЕНТЫ: -- (VCL) sender - компонента VCL (CheckGroup, CheckListBox, ComboBox, ListBox, RadioGroup) -- (table) tab - список строк для выбора в компоненте VCL -- ВОЗВРАЩАЕТ:

3.2 Получение параметров инструмента

Табл. 3.2 – Получение параметров инструмента

№	Функции
2.	-- Получение свойств скрипта getProperties () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблица (указатель) свойств скрипта: -- (string) version - Версия скрипта -- (string) name - Название скрипта -- (string) copyright - Авторские права -- (string) quik_required - Требуемая версия QUIK
3.	-- Получение таблицы атрибутов инструментов из настроек getSecAttrib () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблица атрибутов инструментов из настроек: -- (string) client_code - Код клиента -- (string) class_code - Код класса -- (string) sec_code - Код инструмента -- (string) account - Торговый счет -- (boolean) is_run - Признак включенного инструмента (true) -- (boolean) is_emulator - Признак торговли в режиме имитации (true)
4.	-- Получение числа инструментов в настройках getSecCount () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (number) число инструментов в настройках

№	Функции
5.	-- Получение таблицы настроек и таблицы торговых параметров инструмента getSecParams(index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (table) таблица (указатель) настроек инструмента -- (table) таблица (указатель) торговых параметров инструмента
6.	-- Получение таблицы всех пользовательских заявок getOrders() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблица (указатель) всех пользовательских заявок
7.	-- Получение таблицы всех пользовательских стоп-заявок getStoporders() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблица (указатель) всех пользовательских стоп-заявок
8.	-- Получение таблицы всех пользовательских сделок getTrades() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблица (указатель) всех пользовательских сделок

3.3 Функции сообщений/вызова всплывающих окон

Табл. 3.3 – Функции сообщений/вызова всплывающих окон

№	Функции
1.	-- Вызов окна выбора продолжения queryPopup(diag) -- АРГУМЕНТЫ: -- (string) diag - текстовое сообщение -- ВОЗВРАЩАЕТ: -- (boolean) true, если "Да", иначе false
2.	-- Вызов окна с сообщением и одной кнопкой "ОК" sPopup(diag) -- АРГУМЕНТЫ: -- (string) diag - текстовое сообщение -- ВОЗВРАЩАЕТ:
3.	-- Запись строки в файл с переводом на новую строку, формата "<SERVERTIME> <LOCALTIME> <TITLE> <MODE> <KEY> <DATA>" write2log(data, title, key, mod_type) -- АРГУМЕНТЫ: -- (string) data - строка для записи -- (string) title - титул инструмента -- (string) key - ключ пояснения (0:"", 1:Script, 2:Quik, 3:TransReplay, 4:TradesList, 5:Position, 6:SO, 7:Mts) -- (number) mod_type - режим торговли (0:"", 1:Real, 2:Emulator) -- ВОЗВРАЩАЕТ: -- (boolean) true: строка была записана, иначе false
4.	-- Вывод сообщения в окно сообщений в QUIK, формата "<TITLE> <DATA>" writeMessage(data, title) -- АРГУМЕНТЫ:

№	Функции
	-- (string) data - строка сообщения -- (string) title - титул инструмента -- ВОЗВРАЩАЕТ:
5.	-- Вывод сообщения в окно сообщений в QUIK с приставкой "ВНИМАНИЕ!", формата "ВНИМАНИЕ! <TITLE> <DATA>" writeNoteMessage (data, title, key, mod_type) -- АРГУМЕНТЫ: -- (string) data - строка сообщения -- (string) title - титул инструмента -- (string) key - ключ пояснения (0:"", 1:Script, 2:Quik, 3:TransReplay, 4:TradesList, 5:Position, 6:SO, 7:Mts) -- (number) mod_type - режим торговли (0:"", 1:Real, 2:Emulator) -- ВОЗВРАЩАЕТ:
6.	-- Вывод сообщения в лог с приставкой "quik", формата "<TITLE> Quik <DATA>" writeSystemMessage (data, title) -- АРГУМЕНТЫ: -- (string) data - строка сообщения -- (string) title - титул инструмента -- ВОЗВРАЩАЕТ:
7.	-- Вывод сообщения в окно сообщений в QUIK и в лог, формата "<TITLE> <DATA>" write2messages (data, title, key, mod_type) -- АРГУМЕНТЫ: -- (string) data - строка сообщения -- (string) title - титул инструмента -- (string) key - ключ пояснения (0:"", 1:Script, 2:Quik, 3:TransReplay, 4:TradesList, 5:Position, 6:SO, 7:Mts) -- (number) mod_type - режим торговли (0:"", 1:Real, 2:Emulator) -- ВОЗВРАЩАЕТ:
8.	-- Получение режима торговли для записи в лог файл getLogMode (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (number) режим торговли (0:"", 1:Real, 2:Emulator)

3.4 Функции работы со временем

Табл. 3.4 – Функции работы со временем

№	Функции
1.	-- Получение таблицы (datetime) времени сервера getServerDatetime () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблицу с year (4 знака), month (1-12), day (1-31), hour (0-23), min (0-59), sec (0-61) без ведущих нулей
2.	-- Получение часового пояса getTimeZone () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (string) часовой пояс в формате ISO 8601:2000 (+hhmm or -hhmm) +0400
3.	-- Строка времени с миллисекундами getMsTime () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (string) время (HH:MM:SS,mmm))

№	Функции
4.	-- Смещение часового пояса в секундах offsetSeconds(s) -- АРГУМЕНТЫ: -- (number) s - время в секундах -- ВОЗВРАЩАЕТ: -- (number) смещение часового пояса в секундах
5.	-- Парсинг строки в таблицу (разделители строки знаки пунктуации) splitTime(s) -- АРГУМЕНТЫ: -- (string) s - строка -- ВОЗВРАЩАЕТ: -- (table) таблица подстрок
6.	-- Перевод таблицы датавремени в строку, затем в число datetime2number(dt) -- АРГУМЕНТЫ: -- (table) dt - датавремя -- ВОЗВРАЩАЕТ: -- (number) датавремя формата YYYYMMDDHHMMSS, иначе
7.	-- Преобразование локального времени в number localtime2number() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (number) время 105959 (HHMMSS)
8.	-- Преобразование датывремени QUIK в строку Дата date2string(dt) -- АРГУМЕНТЫ: -- (table) dt - датавремя из таблиц QUIK -- ВОЗВРАЩАЕТ: -- (string) дата формата "year-mm-dd"
9.	-- Преобразование датывремени QUIK в строку datetime2string(dt) -- АРГУМЕНТЫ: -- (table) dt - датавремя из таблиц QUIK -- ВОЗВРАЩАЕТ: -- (string) датувремени формата "year-mm-dd hour:min:sec:ms"
10.	-- Преобразование датывремени QUIK в строку Время time2string(dt) -- АРГУМЕНТЫ: -- (table) dt - датавремя из таблиц QUIK -- ВОЗВРАЩАЕТ: -- (string) время формата "hour:min:sec:ms"

3.5 Функции работы со строками

Табл. 3.5 – Функции работы со строками

№	Функции
1.	-- Добавление ведущего нуля к переменной insertLeadingZero(v) -- АРГУМЕНТЫ: -- (string) v - переменная -- ВОЗВРАЩАЕТ:

№	Функции
	-- (string) переменную с ведущим нулем, если её длина равна 1, формат 7 --> 07
2.	-- Удаление ведущего нуля переменной deleteLeadingZero(v) -- АРГУМЕНТЫ: -- (string) v - переменная -- ВОЗВРАЩАЕТ: -- (string) переменную без ведущего нуля, если он есть, формат 07 --> 7
3.	-- Обрезание пробелов строки слева и справа trim(s) -- АРГУМЕНТЫ: -- (string) s - строка -- ВОЗВРАЩАЕТ: -- (string) строка без пробелов по краям, иначе nil
4.	-- Обрезание строки слева и справа trimPattern(s, r) -- АРГУМЕНТЫ: -- (string) s - строка -- (string) r - паттерн -- ВОЗВРАЩАЕТ: -- (string) строка
5.	-- Обрезание строки слева ltrim(s, r) -- АРГУМЕНТЫ: -- (string) s - строка -- (string) r - паттерн -- ВОЗВРАЩАЕТ: -- (string) строка
6.	-- Обрезание строки справа rtrim(s, r) -- АРГУМЕНТЫ: -- (string) s - строка -- (string) r - паттерн -- ВОЗВРАЩАЕТ: -- (string) строка
7.	-- Парсинг строки в таблицу (разделители строки знаки пунктуации) split(s) -- АРГУМЕНТЫ: -- (string) s - строка -- ВОЗВРАЩАЕТ: -- (table) таблица подстрок
8.	-- Парсинг строки в таблицу с заданным паттерном str2tab(s, p) -- АРГУМЕНТЫ: -- (string) s - строка -- (string) p - паттерн (разделитель) -- ВОЗВРАЩАЕТ: -- (table) таблицу подстрок с индексами по количеству подстрок (1, 2 и т.д.)
9.	-- Замена точки на запятую или наоборот replaceSeparators(s, sep_old, sep_new) -- АРГУМЕНТЫ: -- (string) s - строка -- (string) sep_old - существующий разделитель

№	Функции
	-- (string) sep_new - новый разделитель -- ВОЗВРАЩАЕТ: -- (string) строка с заменой разделителей
10.	-- Удаление всех пробелов в строке http://stackoverflow.com/questions/10460126/how-to-remove-spaces-from-a-string-in-lua removeSpaces(s) -- АРГУМЕНТЫ: -- (string) s - строка -- ВОЗВРАЩАЕТ: -- (string) строку без пробелов

3.6 Функции работы с таблицами

Табл. 3.6 – Функции работы с таблицами

№	Функции
1.	-- Преобразование таблицы в строку tab2string(t) -- АРГУМЕНТЫ: -- (table) t - таблица -- ВОЗВРАЩАЕТ: -- (string) строка
2.	-- Добавления полей одной таблицы к другой. Добавляются только поля, которых в целевой таблице нет. tabAdd(c, t) -- АРГУМЕНТЫ: -- (table) c - таблица, в которую добавляем -- (table) t - таблица, из которой добавляем -- ВОЗВРАЩАЕТ: -- (table) таблица c
3.	-- Объединение нескольких таблиц в одну. Большой приоритет для задания полей у таблиц, идущих в списке первыми. tabUnion(p) -- АРГУМЕНТЫ: -- (table) p - таблицы, которые нужно объединить (result = helper.tab_union{table1, table2, table3}) -- ВОЗВРАЩАЕТ: -- (table) таблица p

3.7 Математические функции

Табл. 3.7 – Математические функции

№	Функции
1.	-- Получение знака параметра sign(v) -- АРГУМЕНТЫ: -- (number) v - параметр -- ВОЗВРАЩАЕТ: -- (number) -1 или 0 или 1
2.	-- Добавление нулей после запятой с заданной точностью setPrecision(v, p) -- АРГУМЕНТЫ: -- (number) v - число, которому надо добавить нули после запятой

№	Функции
	-- (number) p - количество символов после запятой -- ВОЗВРАЩАЕТ: -- (string) число с недостающими нулями после запятой
3.	-- Округление до целого вниз с точностью шага цены floorStepprice (class_code, sec_code, v) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (number) v - округляемое значение -- ВОЗВРАЩАЕТ: -- (number) округленное значение, иначе не округленное
4.	-- Округление до целого вверх с точностью шага цены roundStepprice (class_code, sec_code, v) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (number) v - округляемое значение -- ВОЗВРАЩАЕТ: -- (number) округленное значение, иначе не округленное
5.	-- Округление до целого вниз с заданной точностью floorPrecision (v, p) -- АРГУМЕНТЫ: -- (number) v - округляемое значение -- (number) p - точность -- ВОЗВРАЩАЕТ: -- (number) округленное значение
6.	-- Округление до целого вверх с заданной точностью roundPrecision (v, p) -- АРГУМЕНТЫ: -- (number) v - округляемое значение -- (number) p - точность -- ВОЗВРАЩАЕТ: -- (number) округленное значение
7.	-- Сложение value и shift valuePlusShift (v, shift, is_percent, step) -- АРГУМЕНТЫ: -- (number) v - значение -- (number) shift - добавляемое значение -- (number) is_percent - 0: процент, 1: число -- (number) step - шаг цены -- ВОЗВРАЩАЕТ: -- (number) округленное значение
8.	-- Вычитание shift из value valueMinusShift (v, shift, is_percent, step) -- АРГУМЕНТЫ: -- (number) v - значение -- (number) shift - добавляемое значение -- (number) is_percent - 0: процент, 1: число -- (number) step - шаг цены -- ВОЗВРАЩАЕТ: -- (number) округленное значение
9.	-- Округление с заданной точностью round (v, d)

№	Функции
	-- АРГУМЕНТЫ: -- (number) v - число которое надо округлить -- (number) d - количество символов после запятой -- ВОЗВРАЩАЕТ: -- (number) округленное число
10.	-- Округление с заданной точностью http://lua-users.org/wiki/FormattingNumbers roundEx(v, d) -- АРГУМЕНТЫ: -- (number) v - число которое надо округлить -- (number) d - количество символов после запятой -- ВОЗВРАЩАЕТ: -- (number) округленное число
11.	-- Формат числа http://lua-users.org/wiki/FormattingNumbers formatNum(v, d) -- АРГУМЕНТЫ: -- (number) v - число -- (number) d - количество символов после запятой, по умолчанию 2 -- ВОЗВРАЩАЕТ: -- (string) строку с разделением числа на разряды, 10 078 890.78
12.	-- Формат числа с заданной точностью http://lua-users.org/wiki/FormattingNumbers formatNumEx(v, d) -- АРГУМЕНТЫ: -- (number) v - число -- (number) d - количество символов после запятой -- ВОЗВРАЩАЕТ: -- (string) строку с разделением числа на разряды, 10 078 890.78 -- ПРИМЕРЫ: -- math.format_num(1333444.1,2) 1,333,444.10 -- math.format_num(1333444.1,-2) 1,333,400 -- math.format_num(-22333444.5634,3) -22,333,444.563
13.	-- Преобразование числа 1e-001 в 0.01 formatNumber(v) -- АРГУМЕНТЫ: -- (number) v - цена -- ВОЗВРАЩАЕТ: -- (string) преобразованное число 1e-001 в 0.01
14.	-- Преобразование number в integer toInt(v) -- АРГУМЕНТЫ: -- (number) v - число -- ВОЗВРАЩАЕТ: -- (integer) целая часть числа. Пример, -109.0934 --> -109

3.8 Функции кодировки

Табл. 3.8 – Функции кодировки

№	Функции
1.	-- Преобразование строки из кодировки utf8 в win-1251 utf2win(s) -- АРГУМЕНТЫ: -- (string) s - строка

№	Функции
	-- ВОЗВРАЩАЕТ: -- (string) строка в кодировке win-1251
2.	-- Преобразование строки из кодировки win-1251 в utf8 win2utf(s) -- АРГУМЕНТЫ: -- (string) s - строка -- ВОЗВРАЩАЕТ: -- (string) строка в кодировке utf8

3.9 Функции работы с файлами

Табл. 3.9 – Функции работы с файлами

№	Функции
1.	-- Проверка наличия файла isFileExists(p) -- АРГУМЕНТЫ: -- (string) p - полное имя файла -- ВОЗВРАЩАЕТ: -- (boolean) true: файл существует, иначе false
2.	-- Проверка наличия папки isDirExists(p) -- АРГУМЕНТЫ: -- (string) p - полное путь к папке -- ВОЗВРАЩАЕТ: -- (boolean) true: папка существует, иначе false
3.	-- Получение списка файлов getDirFiles(p) -- АРГУМЕНТЫ: -- (string) p - полный путь папки -- ВОЗВРАЩАЕТ: -- (table) таблица перечисления имен файфлов
4.	-- Получение количества символов в файле getFileSymbolsCount(p) -- АРГУМЕНТЫ: -- (string) p - полный путь с именем файла -- ВОЗВРАЩАЕТ: -- (number) количество символов в файле
5.	-- Получение количества строк в файле getLineCount(p) -- АРГУМЕНТЫ: -- (string) p - полный путь с именем файла -- ВОЗВРАЩАЕТ: -- (number) количество строк в файле
6.	-- Поиск файлов в папке scanDir(p) -- АРГУМЕНТЫ: -- (string) p - полный путь папки ("C:\Acceleration\settings\ -- ВОЗВРАЩАЕТ: -- (table) таблицу с названиями файлов
7.	-- Воспроизведение звукового файла

№	Функции
	playSoundFile (file, alias) ----- АРГУМЕНТЫ ----- -- (string) file - полное имя файла (*.wav, *.mp3, *.midi, *.avi) -- (string) alias - дополнительный идентификатор файла ----- ВОЗВРАЩАЕТ -----
8.	-- Остановка воспроизведения звукового файла stopSoundFile (file, alias) ----- АРГУМЕНТЫ ----- -- (string) file - полное имя файла (*.wav, *.mp3, *.midi, *.avi) -- (string) alias - дополнительный идентификатор файла ----- ВОЗВРАЩАЕТ -----

3.10 Функции получения торговых параметров

Табл. 3.10 – Функции получения торговых параметров

№	Функции
1.	-- Получение таблицы доступных кодов класса getClassesTab () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (table) таблица кодов классов [0]class1, [1]class2 и т.д.
2.	-- Получение типа рынка getExchange (class_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- ВОЗВРАЩАЕТ: -- (number) -1: Неизвестный код класса, 0: Валютный рынок, 1: ММББ, 2: ФОРТС -- (nil) если входные параметры не определены
3.	-- Получение допустимого количества лотов для заявки, скорректированное на лимит по счетам getCorrectCalcBuySell (client_code, class_code, sec_code, account, qty, operation, is_market) -- АРГУМЕНТЫ: -- (string) client_code - код клиента -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (number) qty - количество лотов позиции -- (string) operation - операция ("B", "S") -- (number) is_market - признак рыночной заявки (true - рыночная заявка, false - лимитированная). Необязательный параметр, значение по умолчанию: false -- ВОЗВРАЩАЕТ: -- (number) допустимое количество лотов для заявки
4.	-- Получение ID Фирмы из таблицы "Торговые счета" getFirmId (class_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- ВОЗВРАЩАЕТ: -- (string) ID фирмы или nil
5.	-- Получение вида лимита getLimitKind (sec_code, client_code) -- АРГУМЕНТЫ: -- (string) sec_code - код инструмента -- (string) client_code - код клиента

№	Функции
	-- ВОЗВРАЩАЕТ: -- (number) вид лимита (0: T0, 1: T1, 2: T2), иначе nil
6.	-- Получение размера лота getLotSize (class_code, sec_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ: -- (number) размер лота инструмента, при неудаче 1
7.	-- Расчет цены закрытия позиции getPosClosePrice (class_code, sec_code, qty, spread, is_percents, is_market, order_base_price) -- АРГУМЕНТЫ: -- (string) class_code - код класса инструмента -- (string) sec_code - код инструмента -- (number) qty - размер позиции в лотах -- (number) spread - защитный спрэд цены закрытия -- (boolean) is_percents - true: защитный спрэд в процентах, иначе false -- (boolean) is_market - true: цена закрытия рыночная, иначе false -- (number) order_base_price - тип базовой цены (0:Рыночная, 1:Бид/Оффер, 2:Оффер/Бид) -- ВОЗВРАЩАЕТ: -- (number) price, (string) operation
8.	-- Получение размера открытой позиции в лотах getQty (client_code, class_code, sec_code, account) -- АРГУМЕНТЫ: -- (string) client_code - код клиента -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) account - торговый счет -- ВОЗВРАЩАЕТ: -- (number) размер позиции в лотах
9.	-- Получение количества заявки как доля, % от суммы getQtyRate (class_code, sec_code, operation, rate, sum) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) operation - операция ("B", "S") -- (number) rate - доля, % -- (number) sum - сумма, руб. -- ВОЗВРАЩАЕТ: -- (number) количество лотов на заданный объем
10.	-- Получение количества заявки как объем, руб. getQtyValue (class_code, sec_code, operation, value) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) operation - операция ("B", "S") -- (number) value - объем, руб. -- ВОЗВРАЩАЕТ: -- (number) количество лотов на заданный объем
11.	-- Получение допустимого количества лотов для заявки getSecLimit (index, operation, qty) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента

№	Функции
	-- (string) operation - операция ("B", "S") -- (number) qty - количество позиции -- ВОЗВРАЩАЕТ: -- (number) допустимое количество лотов для заявки
12.	-- Статус торговли бумагой getSecTradeStatus (class_code, sec_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ: -- (boolean) true: торгуется, иначе false -- (number) состояние торговли инструментом, -1 -- (string) наименование состояния торговли инструментом, иначе ""
13.	-- Сравнение версий, в частности проверка версии терминала QUIK compareVersion (req, cur) -- АРГУМЕНТЫ: -- (string) req - требуемая версия -- (string) cur - текущая версия, может быть не обязательной. Если nil, то req сравнивается с версией QUIK -- ВОЗВРАЩАЕТ: -- (boolean) true: версия старше или совпадает, false: версия младше
14.	-- Сравнение пользовательских кода класса и кода бумаги с доступными в QUIK compareClasses (class_code, sec_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ: -- (boolean) true: class_code и sec_code существуют, иначе false
15.	-- Конвертация числа шагов цены в пункты convertSteps2Pips (class_code, sec_code, steps) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (number) steps - шаги цены -- ВОЗВРАЩАЕТ: -- (number) пункты
16.	-- Получение типа счета - отдельный или объединенный isSeparateAccount (client_code, class_code, sec_code) -- АРГУМЕНТЫ: -- (string) client_code - код клиента -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ: -- (boolean) тип счета (true: отдельный, false: объединенный)
17.	-- Проверка на попадание цены в лимиты ФОРТС isFortsMinmaxPrice (class_code, sec_code, price) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (number) price - цена заявки -- ВОЗВРАЩАЕТ: -- (boolean) true: цена в пределах лимитов, иначе false
18.	-- Ограничение цены лимитными ценами ФОРТС setFortsMinmaxPrice (class_code, sec_code, price)

№	Функции
	-- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (number) price - цена заявки -- ВОЗВРАЩАЕТ: -- (number) цена ограниченная лимитными ценами, если выходит за их рамки

3.11 Функции получения данных графиков

Табл. 3.11 – Функции получения данных графиков

№	Функции
1.	-- Проверка существования графика isChartExist (identifier) -- АРГУМЕНТЫ: -- (string) identifier - идентификатор графика -- ВОЗВРАЩАЕТ: -- (boolean) true: график существует, иначе false
2.	-- Получение таблицы свечек getCandles (identifier, count, offset) -- АРГУМЕНТЫ: -- (string) identifier - идентификатор графика -- (string) count - количество запрашиваемых свечек с конца (справа налево) -- (number) offset - смещение первой свечки на offset влево (не обязательный, по умолчанию 0) -- ВОЗВРАЩАЕТ: -- (table) table[line][candle], где line от 1 до числа линий, candle от 0 до числа заказанных свечек слева направо -- table[1][0] = doesExist, open, high, low, close, volume, datetime = {year, month, day, week_day, hour, min, sec, ms} -- (number) число полученных линий
3.	-- Получение временного интервала графика getTimeframe (identifier) -- АРГУМЕНТЫ: -- (string) identifier - идентификатор графика -- ВОЗВРАЩАЕТ: -- (number) временной интервал графика в секундах, иначе nil
4.	-- Конвертирование таймфрейма из секунд в минуты convertTimeframe (timeframe) -- АРГУМЕНТЫ: -- (number) timeframe - временной интервал графика в секундах -- ВОЗВРАЩАЕТ: -- (number) временной интервал графика в минутах или -1: день, -2: неделя, -3: месяц, иначе nil

3.12 Функции получения данных стакана котировок

Табл. 3.12 – Функции получения данных стакана котировок

№	Функции
1.	-- Подписка на получение котировок без открытия стакана subscribeQuotes (class_code, sec_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ:

№	Функции
2.	-- Отписка от получения котировок стакана unsubscribeQuotes (class_code, sec_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ:
3.	-- Получение цены/объема из стакана котировок getQuote (class_code, sec_code, quote_type, quote_value, row) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) quote_type - тип заявки ("BID", "OFFER") -- (string) quote_value - тип значения ("PRICE", "QUANTITY") -- (number) row - номер строки в стакане (лучшая 1) -- ВОЗВРАЩАЕТ: -- (number) цену/объем заявки (считая от лучшей цены), иначе 0
4.	-- Получение цены/объема из стакана котировок getBestQuote (class_code, sec_code, quote_type, quote_value) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) quote_type - тип заявки ("BID", "OFFER") -- (string) quote_value - тип значения ("PRICE", "QUANTITY") -- ВОЗВРАЩАЕТ: -- (number) цену/объем заявки (считая от лучшей цены), иначе 0
5.	-- Получение лучшей цены/объема из стакана котировок по заданному объему getBestValueQuote (class_code, sec_code, quote_type, quote_value, value) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- (string) quote_type - тип заявки ("BID", "OFFER") -- (string) quote_value - тип значения ("PRICE", "QUANTITY") -- (number) value - объем заявок -- ВОЗВРАЩАЕТ: -- (number) цену/объем заявки (считая от лучшей цены), иначе 0
6.	-- Получение цены/объема из стакана котировок по таблице котировок getQuoteFromTab (quotes, quote_type, quote_value, row) -- АРГУМЕНТЫ: -- (table) quotes - таблице котировок -- (string) quote_type - тип заявки ("BID", "OFFER") -- (string) quote_value - тип значения ("PRICE", "QUANTITY") -- (number) row - номер строки в стакане -- ВОЗВРАЩАЕТ: -- (number) цену/объем заявки (считая от лучшей цены), иначе 0
7.	-- Получение наименьшего числа строк в стакане котировок из BID и OFFER minQuoteRows (class_code, sec_code) -- АРГУМЕНТЫ: -- (string) class_code - код класса -- (string) sec_code - код инструмента -- ВОЗВРАЩАЕТ: -- (number) наименьшее число строк в стакане котировок из BID и OFFER, иначе 0

3.13 Получение параметров стоп-заявок

Табл. 3.13 – Получение параметров стоп-заявок

№	Функции
1.	-- Конвертация валюты в проценты convertPrice (index, value, units) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) value - отступ/смещение -- (number) units - тип ед. измерения (0: проценты, 1: шаги цены) -- ВОЗВРАЩАЕТ: -- (number) цену в процентах
2.	-- Получение значения стоп(тейк)-индикатора getSoIndicator (index, operation, so_type) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция В или S -- (string) so_type - тип стоп-заявки (S: стоп-лосс, T: тейк-профит) -- ВОЗВРАЩАЕТ: -- (number) стоп-цену
3.	-- Получение цены getSoPrice (index, stopprice, operation) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) stopprice - стоп-цена -- (string) operation - операция В или S -- ВОЗВРАЩАЕТ: -- (number) цену
4.	-- Получение цен стоп-заявки из стоп-заявки getSoPricesBySo (so) -- АРГУМЕНТЫ: -- (table) so - стоп-заявка -- ВОЗВРАЩАЕТ: -- (table) stopprice, price, takeprice, offset, offset_units, spread, spread_units
5.	-- Получение стоп-цены getSoStopprice (index, operation, trade_price) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция В или S -- (number) trade_price - цена позиции/сделки -- ВОЗВРАЩАЕТ: -- (number) стоп-цену
6.	-- Получение тейк-цены getSoTakeprice (index, operation, trade_price) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция В или S -- (number) trade_price - цена позиции/сделки -- ВОЗВРАЩАЕТ: -- (number) тейк-цену
7.	-- Получение типа стоп-заявки из стоп-заявки getSoTypeBySo (so) -- АРГУМЕНТЫ: -- (table) so - стоп-заявка

№	Функции
	-- ВОЗВРАЩАЕТ: -- (string) тип стоп-заявки
8.	-- Получение типа стоп-заявки по настройкам торговли getSoTypeBySettings(index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (string) тип стоп-заявки

3.14 Функции для работы с флагами заявок/стоп-заявок

Табл. 3.14 – Функции для работы с флагами заявок/стоп-заявок

№	Функции
1.	-- Получение статуса ордера по таблице битов getOrderAttrib(f) -- АРГУМЕНТЫ: -- (table) f - таблица битов -- ВОЗВРАЩАЕТ: -- (table) возвращает таблицу с параметрами -- (string) status - "active" / "cancelled" / "filled" / "unknown" (активна / снята / исполнена / неизвестный статус) -- (string) operation - ("B" - Покупка, "S" - Продажа, "" - неизвестная операция)
2.	-- Получение таблицы с описанием заявки по флагам f2tabOrder(f) -- АРГУМЕНТЫ: -- (table) f - таблица битов -- ВОЗВРАЩАЕТ: -- (table) возвращает таблицу с описанием заявки по флагам -- Атрибуты: -- (0x1 и 0x2) status: "active" / "cancelled" / "filled" / "unknown" (активна / снята / исполнена / неизвестный статус) -- (0x4) operation ("B": Buy, "S": Sell, "X": неизвестная операция) -- (0x8) is_limit (true: лимитная, false: рыночная) -- (0x10) mte (true: возможно исполнение заявки несколькими сделками) -- (0x20) fill_or_kill (true: исполнить заявку немедленно или снять) -- (0x40) mm_order (true: заявка маркет-мейкера. Для адресных заявок – заявка отправлена контрагенту) -- (0x80) received (true: для адресных заявок – заявка получена от контрагента) -- (0x100) cancell_rest (true: снять остаток) -- (0x200) is_iceberg (true: айсберг-заявка)
3.	-- Получение таблицы с описанием стоп-заявки по флагам f2tabStop(f) -- АРГУМЕНТЫ: -- (table) f - таблица битов -- ВОЗВРАЩАЕТ: -- (table) возвращает таблицу с описанием стоп-заявки по флагам -- Атрибуты: -- (0x1 и 0x2) status: "active" / "cancelled" / "filled" / "unknown" (активна / снята / исполнена / неизвестный статус) -- (0x4) operation ("B" - Buy, "S" - Sell, "" - неизвестная операция) -- (0x8) is_limit (true: лимитная, false: рыночная) -- (0x20) in_active (true: ожидает активации) -- (0x40) another_server (true: другой сервер)

№	Функции
	-- (0x100) take_by_order (true: в случае стоп-заявки типа тейк-профита по заявке) -- (0x200) user_activated (true: активирована вручную) -- (0x400) rejected_by_ts (true: сработала, но была отвергнута торговой системой) -- (0x800) not_pass_limits (true: сработала, но не прошла контроль лимитов) -- (0x1000) linked_killed (true: снята, так как снята связанная заявка) -- (0x2000) linked_filled (true: снята, так как связанная заявка исполнена) -- (0x8000) minmax_calc (true: идет расчет минимума-максимума)
4.	-- Получение таблицы с описанием стоп-заявки по флагам дополнительных параметров f2tabStopEx(f) -- АРГУМЕНТЫ: -- (table) f - таблица битов -- ВОЗВРАЩАЕТ: -- (table) возвращает таблицу с описанием стоп-заявки по флагам -- Атрибуты: -- (0x1) use_balance (true: Использовать остаток основной заявки) -- (0x2) kill_partially (true: При частичном исполнении заявки снять стоп-заявку) -- (0x4) fill_by_linked (true: Активировать стоп-заявку при частичном исполнении связанной заявки) -- (0x8) offset_per (true: Отступ задан в процентах, иначе – в пунктах цены) -- (0x10) spread_per (true: Защитный спред задан в процентах, иначе – в пунктах цены) -- (0x20) is_today (true: Срок действия стоп-заявки ограничен сегодняшним днем) -- (0x40) is_expiry (true: Установлен интервал времени действия стоп-заявки) -- (0x80) market_take (true: Выполнение тейк-профита по рыночной цене) -- (0x100) market_stop (true: Выполнение стоп-заявки по рыночной цене)
5.	-- Получение таблицы с описанием сделки по флагам f2tabTrade(f) -- АРГУМЕНТЫ: -- (table) f - таблица битов -- ВОЗВРАЩАЕТ: -- (table) возвращает таблицу с описанием сделки по флагам -- Атрибуты: operation ("B" - Buy, "S" - Sell, "" - неизвестная операция)

3.15 Функции получения цены позиции

Табл. 3.15 – Функции получения цены позиции

№	Функции
1.	-- Получение средневзвешенной цены открытой позиции getAWP(index, qty, price, trades) -- АРГУМЕНТЫ: -- (number) index - порядковый номер инструмента в таблице S -- (number) qty - лоты позиции -- (number) price - цена перенесенной позиции, по которой нет сделок -- (number) trades - таблица всех сделок -- ВОЗВРАЩАЕТ: -- (number) цену позиции с учетом комиссии, иначе 0
2.	-- Получение эффективной цены позиции по методике биржи getEffectivePrice(index, qty, price, trades) -- АРГУМЕНТЫ: -- (number) index - порядковый номер инструмента в таблице S -- (number) quantity - лоты позиции -- (number) price - цена перенесенной позиции, по которой нет сделок -- (number) trades - таблица всех сделок -- ВОЗВРАЩАЕТ:

№	Функции
	-- (number) цену позиции, иначе 0
3.	-- Получение эффективной цены позиции на ФОРТС getFortsAWP (sec_code, account) -- АРГУМЕНТЫ: -- (string) sec_code - код инструмента -- (string) account - торговый счет -- ВОЗВРАЩАЕТ: -- (number) эффективную цену позиции, иначе 0 -- (number) размер позиции, иначе 0
4.	-- Получение цены приобретения на ММББ getDepoAWP (sec_code, client_code, account, limit_kind) -- АРГУМЕНТЫ: -- (string) sec_code - код инструмента -- (string) client_code - код клиента -- (string) account - торговый счет -- (number) limit_kind - срок расчета T+ (необязателен) -- ВОЗВРАЩАЕТ: -- (number) цену приобретения позиции, иначе 0 -- (number) размер позиции, иначе 0
5.	-- Получение цены позиции на валютном рынке getCurrencyAWP (sec_code, class_code, client_code, limit_kind) ----- АРГУМЕНТЫ ----- -- (string) sec_code - код инструмента -- (string) class_code - код класса -- (string) client_code - код клиента -- (number) limit_kind - срок расчета T+ (необязателен) ----- ВОЗВРАЩАЕТ ----- -- (number) цену позиции, иначе 0 -- (number) размер позиции, иначе 0
6.	-- Получение таблицы сделок открытой позиции getTradesList (index, qty, price, trades) -- АРГУМЕНТЫ: -- (number) index - порядковый номер инструмента в таблице S -- (number) qty - лоты позиции -- (number) price - цена перенесенной позиции, по которой нет сделок -- (number) trades - таблица всех сделок -- ВОЗВРАЩАЕТ: -- (table) таблицу сделок ([1], [2], и т.д.) -- number -- номер -- operation -- операция "B" или "S" -- qty -- количество в лотах -- price -- цена -- value -- объем -- comission -- комиссия

3.16 Функции получения финансового результата

Табл. 3.16 – Функции получения финансового результата

№	Функции
1.	-- Получение текущей стоимости портфеля (контракты + деньги) по всем счетам ФОРТС из таблицы "Ограничения по клиентским счетам" getAllFortsMoney (profit)

№	Функции
	-- АРГУМЕНТЫ: -- (number) profit - прибыль/убыток по всем счетам ФОРТС с учетом комиссии -- ВОЗВРАЩАЕТ: -- (number) текущая стоимость портфеля по всем счетам ФОРТС с учетом комиссии
2.	-- Получение финансового результата по всем счетам ФОРТС из таблицы "Ограничения по клиентским счетам" getAllFortsProfit() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (number) прибыль/убыток по всем счетам ФОРТС с учетом комиссии
3.	-- Получение текущей стоимости портфеля по всем счетам ММББ из таблицы "Клиентский портфель" getAllMicexMoney() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (number) текущая стоимость портфеля по всем счетам ММББ с учетом комиссии
4.	-- Получение финансового результата по всем счетам ММББ из таблицы "Клиентский портфель" getAllMicexProfit() -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (number) прибыль/убыток по всем счетам ММББ с учетом комиссии
5.	-- Получение общего объема открытых позиций в руб. getAllPosValue(is_forts) -- АРГУМЕНТЫ: -- (boolean) is_forts - true: ФОРТС, иначе ММББ -- ВОЗВРАЩАЕТ: -- (number) общий объем открытых позиций в руб.
6.	-- Расчет результата открытой позиции в пунктах getResPips(qty, trade_price, last_price, precision) -- АРГУМЕНТЫ: -- (number) qty - лоты позиции -- (number) trade_price - цена позиции -- (number) last_price - рыночная цена -- (number) precision - точность округления -- ВОЗВРАЩАЕТ: -- (number) результат открытой позиции в пунктах, иначе 0
7.	-- Получение размера позиции в лотах в начале торгов getStartQty(index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (number) Размер позиции в лотах в начале торгов
8.	-- Получение количества полных трейдов getFullTradesCount (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (number) Количество полных трейдов за текущую торговую дату

3.17 Функции работы со стратегиями

Табл. 3.17 – Функции работы со стратегиями

№	Функции
1.	-- Принудительный запуск стратегии runStrategy (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента
2.	-- Расчет количества лотов в заявке открытия позиции calculateOpenQty (index, operation) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция (B: Покупка, S: Продажа) -- ВОЗВРАЩАЕТ: -- (number) количество лотов
3.	-- Расчет цены заявки calculatePrice (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) signal - тип сигнала (0: Без сигнала, 1: Лонг, 2: Шорт, 3: Закрывать лонг, 4: Закрывать шорт) -- (boolean) is_risk - признак расчета цены закрытия по финансовому результату (модуль "Риск") -- ВОЗВРАЩАЕТ: -- (number) цену заявки -- (boolean) признак рыночной заявки (true), в этом случае цена = 0
4.	-- Расчет количества лотов в заявке при реверсе позиции calculateReverseQty (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) signal - тип сигнала (0: Без сигнала, 1: Лонг, 2: Шорт, 3: Закрывать лонг, 4: Закрывать шорт) -- ВОЗВРАЩАЕТ: -- (number) количество лотов
5.	-- Расчет количества лотов согласно риску с ограничением позиции по размеру стоп-лосс calculateRiskQty (index, qty, operation) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) qty - плановый размер открываемой позиции -- (string) operation - операция (B: Покупка, S: Продажа) -- ВОЗВРАЩАЕТ: -- (number) количество лотов, скорректированных на риск -- (boolean) true, стоп-лосс попадает в диапазон риска, иначе false
6.	-- Инверсия сигналов стратегии inverseSignal (signal) -- АРГУМЕНТЫ: -- (table) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ: -- (table) инвертированный сигнал стратегии
7.	-- Проверка на условие сделок на закрытии бара isCloseBarTime (index, sec_to_bar_end) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) sec_to_bar_end - секунды до закрытия бара -- ВОЗВРАЩАЕТ: -- (boolean) true: до закрытия бара осталось sec_to_bar_end заданных секунд, иначе false
8.	-- Сделки на закрытии бара isCloseOfBar (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента

№	Функции
	-- (number) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ: -- (boolean) true: период закрытия бара, иначе false
9.	-- Признак пропуска первого сигнала стратегии после запуска isFirstSignal (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ: -- (boolean) true: первый сигнал, иначе false
10.	-- Проверка на изменение цены бумаги на -3% к цене закрытия предыдущей сессии isMicexLastchangeLimit (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (boolean) достигнут лимит изменения цены бумаги, запрет Шорта
11.	-- Один трейд за бар isOneTradeBar (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (boolean) true: за текущий бар была сделка, иначе false
12.	-- Закрытие по сигналам открытия open2closeSignal (signal) -- АРГУМЕНТЫ: -- (table) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ: -- (table) сигнал стратегии
13.	-- Снятие блокировки заявок открытия позиции unlockSignal (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (table) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ: -- (boolean) true: можно снимать блокировку, иначе false
14.	-- Открытие Лонга openLong (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ:
15.	-- Открытие Шорта openShort (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ:
16.	-- Закрытие Лонга closeLong (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (table) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ:
17.	-- Закрытие Шорта

№	Функции
	closeShort (index, signal) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (table) signal - таблица сигнала стратегии -- ВОЗВРАЩАЕТ:

3.18 Транзакции торговых операций

Табл. 3.18 – Транзакции торговых операций

№	Функции
1.	-- Получение идентификатора транзакции getTransId () -- АРГУМЕНТЫ: -- ВОЗВРАЩАЕТ: -- (number) trans_id
2.	-- Конвертация цены для lua5.3 formatPrice (price, sec_scale) -- АРГУМЕНТЫ: -- (number) price - цена -- (number) sec_scale - точность цены -- ВОЗВРАЩАЕТ: -- (string) price с точностью шага цены, иначе price (пример, 1.0 --> 1 при точности == 0)
3.	-- Создание заявки по инструменту sendOrder (index, operation, qty, price, is_market, trans_id, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция "B" или "S" -- (number) qty - количество лотов -- (number) price - цена заявки -- (boolean) is_market - рыночная заявка (true: да, false: нет) -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
4.	-- Создание заявки по инструменту sendOrderEx (index, client_code, class_code, sec_code, account, operation, qty, price, is_market, trans_id, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента (торговый счет для ФОРТС) -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (string) operation - операция "B" или "S" -- (number) qty - количество лотов -- (number) price - цена заявки -- (boolean) is_market - рыночная заявка (true: да, false: нет) -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
5.	-- Создание заявки на внебиржевую сделку sendNegOrder (index, operation, qty, price, partner, trans_id, reason)

№	Функции
	-- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция "B" или "S" -- (number) qty - количество лотов -- (number) price - цена заявки -- (string) partner - код организации – партнера по внебиржевой сделке -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
6.	-- Создание заявки на внебиржевую сделку sendNegOrderEx (index, client_code, class_code, sec_code, account, operation, qty, price, partner, trans_id, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента (торговый счет для ФОРТС) -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (string) operation - операция "B" или "S" -- (number) qty - количество лотов -- (number) price - цена заявки -- (string) partner - код организации – партнера по внебиржевой сделке -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
7.	-- Создание стоп-заявки по инструменту sendStoporder (index, so_type, operation, qty, stopprice, price, takeprice, trans_id, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) so_type - "S", "T", "TS" или "L" - вид стоп-заявки: стоп-лимит, тейк-профит, теки-профит и стоп-лимит, стоп со связ. заявкой -- (string) operation - операция "B" или "S" -- (number) qty - количество лотов -- (number) stopprice - стоп-цена -- (number) price - цена заявки после исполнения стоп-цены -- (number) takeprice - цена тейк-профит -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания стоп-заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
8.	-- Создание стоп-заявки по инструменту sendStoporderEx (index, client_code, class_code, sec_code, account, so_type, expiry, operation, qty, stopprice, price, takeprice, takeoffset, takeoffset_units, takespread, takespread_units, trans_id, reason, stoplimit_units, expiry_days) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента (торговый счет для ФОРТС) -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (string) so_type - "S", "T", "TS" или "L" - вид стоп-заявки: стоп-лимит, тейк-профит, теки-профит и стоп-лимит, стоп со связ. заявкой -- (number) expiry - срок действия (0: до отмены, 1: сегодня) -- (string) operation - операция "B" или "S" -- (number) qty - количество лотов -- (number) stopprice - стоп-цена

№	Функции
	-- (number) price - цена заявки после исполнения стоп-цены -- (number) takeprice - цена тейк-профит -- (number) takeoffset - отступ от мин (макс) тейк-профит -- (number) takeoffset_units - единицы измерения отступа от мин (макс) тейк-профит (0: проценты, 1: валюта) -- (number) takespread - защитный спрэд (проскальзывание) тейк-профит -- (number) takespread_units - единицы измерения защитного спрэда тейк-профит (0: проценты, 1: валюта) -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания стоп-заявки, может быть nil -- (number) stoplimit_units - единицы измерения защитного спрэда стоп-лимит (0: проценты, 1: валюта, 2: рыночная) -- (number) expiry_days - число дней действия стоп-заявки -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
9.	-- Создание реверсивной заявки по инструменту sendReverse (index, qty, new_qty, trans_id, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) qty - количество лотов позиции со знаком -- (number) new_qty - количество лотов новой (противоположной) позиции -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
10.	-- Создание реверсивной заявки по инструменту sendReverseEx (index, client_code, class_code, sec_code, account, qty, new_qty, trans_id, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента (торговый счет для ФОРТС) -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (number) qty - количество лотов позиции со знаком -- (number) new_qty - количество лотов новой (противоположной) позиции -- (number) trans_id - идентификатор транзакции -- (string) reason - причина создания заявки, может быть nil -- ВОЗВРАЩАЕТ: -- (boolean) true: транзакция создана, иначе false
11.	-- Снятие активной заявки или стоп-заявки killOrder (index, order, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (table) order - таблица параметров заявки/стоп-заявки -- (string) reason - причина снятия, может быть nil -- ВОЗВРАЩАЕТ: -- (string) trans_id - идентификатор транзакции, иначе -1
12.	-- Снятие активной заявки или стоп-заявки killOrderEx (index, class_code, sec_code, ord_type, order_num, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) ord_type - тип ордера (ord: заявка, so: стоп-заявка) -- (number) order_num - номер ордера -- (string) reason - причина снятия, может быть nil -- ВОЗВРАЩАЕТ:

№	Функции
	-- (string) trans_id - идентификатор транзакции, иначе -1
13.	-- Снятие всех активных ордеров killAllOrders (index, ord_type, operation, reason, is_all) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) ord_type - тип ордера ("ord": заявка, "so": стоп-заявка) -- (string) operation - операция ("S": только продажа, "B": только покупка, "": все), может быть nil == все -- (string) reason - причина снятия, может быть nil -- (boolean) is_all - тип снятия (true: по всему счету, false: только по инструменту), может быть nil == только по инструменту -- ВОЗВРАЩАЕТ: -- (table) таблицу с перечислением созданных trans_id
14.	-- Снятие всех активных ордеров killAllOrdersEx (index, client_code, class_code, sec_code, account, ord_type, operation, reason, is_all) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (string) ord_type - тип ордера ("ord": заявка, "so": стоп-заявка) -- (string) operation - операция ("S": только продажа, "B": только покупка, "": все), может быть nil == все -- (string) reason - причина снятия, может быть nil -- (boolean) is_all - тип снятия (true: по всему счету, false: только по инструменту), может быть nil == только по инструменту -- ВОЗВРАЩАЕТ: -- (table) таблицу с перечислением созданных trans_id
15.	-- Снятие всех активных заявок и стоп-заявок killAll (index, operation, reason, is_all) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) operation - операция (S: только продажа, B: только покупка, "": все), может быть nil == все -- (string) reason - причина снятия, может быть nil -- (boolean) is_all - тип снятия (true: по всему счету, false: только по инструменту), может быть nil == только по инструменту -- ВОЗВРАЩАЕТ: -- (table) таблицу с перечислением созданных trans_id
16.	-- Снятие всех активных заявок и стоп-заявок killAllEx (index, client_code, class_code, sec_code, account, operation, reason, is_all) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (string) operation - операция (S: только продажа, B: только покупка, "": все), может быть nil == все -- (string) reason - причина снятия, может быть nil -- (boolean) is_all - тип снятия (true: по всему счету, false: только по инструменту), может быть nil == только по инструменту -- ВОЗВРАЩАЕТ: -- (table) таблицу с перечислением созданных trans_id
17.	-- Снятие всех активных заявок и стоп-заявок по всем счетам killAlls (index, ord_type, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента

№	Функции
	-- (string) ord_type - тип ордера (ord: заявка, so: стоп-заявка) -- (string) reason - причина снятия, может быть nil -- ВОЗВРАЩАЕТ: -- (table) таблицу с перечислением созданных trans_id
18.	-- Закрытие всех позиций closeAll (index, reason, is_all) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) reason - причина снятия, может быть nil -- (boolean) is_all - тип снятия (true: по всему счету, false: только по инструменту), может быть nil == только по инструменту -- ВОЗВРАЩАЕТ: -- (string) trans_id - идентификатор транзакции, иначе -1
19.	-- Закрытие всех позиций closeAllEx (index, client_code, class_code, sec_code, account, reason, is_all) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (string) client_code - код клиента -- (string) class_code - код класса бумаги -- (string) sec_code - код инструмента -- (string) account - торговый счет -- (string) reason - причина снятия, может быть nil -- (boolean) is_all - тип снятия (true: по всему счету, false: только по инструменту), может быть nil == только по инструменту -- ВОЗВРАЩАЕТ: -- (string) trans_id - идентификатор транзакции, иначе -1

3.19 Алгоритмические заявки

Табл. 3.19 – Алгоритмические заявки

№	Функции
1.	-- Получение параметров активных лестниц заявок getLadders (index) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- ВОЗВРАЩАЕТ: -- (table) аблицу параметров лестниц заявок инструмента, иначе nil
2.	-- Создание лестницы заявок sendLadder (index, direction, operation, startQty, stratPrice, ordersNumber, koef, step, stepUnit, offset, offsetUnit, reason) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента в таблице S -- (number) direction - направление лестницы (0: за ценой, 1: против цены) -- (string) operation - операция лестницы ("B", "S") -- (number) startQty - количество лотов первой заявки -- (number) stratPrice - ступень первой заявки -- (number) ordersNumber - количество заявок -- (number) koef - коэффициент расчета лотов лестницы -- (number) step - шаг лестницы -- (number) stepUnit - единицы шага лестницы (0: проценты, 1: шаги цены) -- (number) offset - отступ заявки от ступени лестницы -- (number) offsetUnit - единицы отступа заявки (0: проценты, 1: шаги цены) -- (string) reason - причина создания, может быть nil -- ВОЗВРАЩАЕТ:

№	Функции
	-- (number) идентификатор лестницы заявок, иначе nil
3.	-- Отмена лестницы заявок killLadder (index, ladder_id) -- АРГУМЕНТЫ: -- (number) index - индекс инструмента -- (number) ladder_id - идентификатор лестницы заявок, если nil, то все лестницы инструмента -- ВОЗВРАЩАЕТ: -- (boolean) true при успешной отмене, иначе false -- (string) причина отказа при неуспешной отмене лестницы заявок
4.	-- Создание айсберг заявки sendIceberg (index, operation, visqty, qty, price, lot_spread, price_spread, pause) ----- АРГУМЕНТЫ ----- -- (number) id - индекс инструмента в таблице S -- (string) operation - операция айсберг заявки ("B", "S") -- (number) visqty - видимое количество лотов -- (number) qty - общее количество лотов -- (number) price - цена заявок -- (number) lot_spread - разброс лотов (%), иначе 0 -- (number) price_spread - разброс цен (%), иначе 0 -- (number) pause - пауза между заявками (сек), иначе 0 ----- ВОЗВРАЩАЕТ ----- -- (number) идентификатор айсберг заявки, иначе nil
5.	-- Отмена айсберг заявок killIceberg (index, is_all) ----- АРГУМЕНТЫ ----- -- (number) id - индекс инструмента в таблице S -- (boolean) is_all - снимать все айсберги (true:да по всем инструментам, false:нет, только по заданному инструменту)

4 ПАРАМЕТРЫ ИНСТРУМЕНТА

Параметры инструмента разделены на две таблицы:

- Таблица настроек торговли;
- Таблица текущих торговых параметров.

Доступ к таблицам осуществляется через функцию **getSecParams**(number index)

```
-- Атрибуты стратегии
iS.mts.minimal = {}
-- Стратегия
function iS.mts.minimal.main(index)
    -- Параметры инструмента (настройки торговли, торговые параметры)
    local s, t = iS.f.getSecParams(index)
    message(s.Security.sec_code.." last price = "..tostring(t.Security.last_price), 1)

    return 0, false, false
end
```

4.1 Настройки торговли

Таблица настроек инструмента имеет структуру, описанную в Табл. 4.1.

Табл. 4.1 – Таблица настроек торговли инструмента

№	Параметры инструмента
1.	Инструмент ["Security"] = { ["is_run"] = false, -- Включено (true/false) ["title"] = "NEW #1", -- Титул инструмента (sec_code..#) ["sec_code"] = "NEW", -- Код инструмента ["class_code"] = "NEW_CLASS", -- Код класса ["account"] = "L01+00000F00", -- Торговый счет ["client_code"] = "12345", -- Код клиента ["ba_sec_code"] = "", -- Код бумаги базового актива ["ba_class_code"] = "" -- Код класса базового актива }
2.	Закрытие позиций по времени ["Time"] = { ["one_enabled"] = false, -- Включено время 1 (true/false) ["one_seconds"] = 5, -- Секунды до закрытия 1 ["one_time"] = "140000", -- Время закрытия 1 ["two_enabled"] = false, -- Включено время 2 (true/false) ["two_seconds"] = 5, -- Секунды до закрытия 2 ["two_time"] = "184500", -- Время закрытия 2 ["three_enabled"] = false, -- Включено время 3 (true/false) ["three_seconds"] = 5, -- Секунды до закрытия 3 ["three_time"] = "235000", -- вВремя закрытия 3 ["position_enabled"] = false, -- Закрытие позиции через N секунд (true/false) ["position"] = 0, -- Время открытой позиции в секундах ["posplus_enabled"] = false, -- Закрытие позиции через N секунд только в "плюсе" (true/false) ["posminus_enabled"] = false -- Закрытие позиции через N секунд если позиция в "минусе" (true/false) }
3.	Дополнительные опции ["More"] = { ["quik_tradeprice"] = 0, -- Цена позиции (0: Из таблиц QUIK, 1: Средневзвешенная по результату, 2: Эффективная по методике биржи)

№	Параметры инструмента
	<pre> ["quik_posqty"] = 0, -- Расчет размера позиции (0: По таблицам QUIK, 1: По сделкам) ["quik_avlots"] = 0 -- Расчет доступных лотов ФОРТС (0: По методике QUIK, 1: По плановым позициям) } </pre>
4.	Режим торговли <pre> ["Mode"] = { ["is_emulator"] = false -- Режим имитации (true/false) } </pre>
5.	Метки <pre> ["Labels"] = { ["enabled"] = false, -- Отображать метки сделок в режиме Имитации (true/false) ["identifier"] = "" -- Идентификатор графика цены для меток } </pre>
6.	Стакан котировок <pre> ["DOM"] = { ["is_dom_iceberg"] = false, -- Признак создания айсберг заявок из стакана котировок ["is_dom_risk"] = false, -- Признак риска позиции в стакане котировок ["is_dom_maxqty"] = false, -- Признак максимума лотов в стакане котировок ["maxqty"] = 99999 -- Максимум лотов в стакане котировок }, </pre>
7.	Заккрытие позиций по финансовому результату / Комиссия брокера <pre> ["Risk"] = { ["day_profit"] = true, -- Расчет прибыли за текущий день (true/false) ["forts_profit"] = false, -- Расчет прибыли ФОРТС по таблице Позиции по клиентским счетам (true/false) ["dis_profit"] = false, -- Дискретный рост лимита прибыли (true/false) ["value"] = 0, -- Размер комиссии брокера ["units"] = 0, -- Единицы комиссии брокера (0: Процент от транзакции, 1: Валюта за лот) ["enabled"] = false, -- Включено (true/false) ["offset_always"] = false, -- Всегда показывать Откат от прибыли (true/false) ["profit"] = 0, -- Максимальная прибыль ["loss"] = 0, -- Максимальный убыток ["offset"] = 0, -- Откат прибыли ["offset_units"] = 0, -- Ед. отката прибыли (0: проценты, 1: валюта) ["drop_connection"] = false, -- Разрывать соединение с информационным сервером (true/false) ["params_enabled"] = false, -- По параметру при открытии позиции (true/false) ["param"] = 0, -- Сигнальный параметр закрытия позиции (0:Open, 1:Close, 2:High/Low, 3:Макс./мин. из Open, 4:Макс./мин. из Close, 5:Макс./мин. из High/Low, 6:Макс./мин. дня) ["param_bar_number"] = 2, -- Номер свечи ["param_chart_id"] = "NEW_PRICE", -- Идентификатор графика цены ["param_condition"] = 0, -- Условие (0:Close, 1:High/Low, 2:Last) ["param_spread"] = 0, -- Спрэд ["param_spread_units"] = 0, -- Ед. спреда (0:проценты, 1:шаги цены) ["order_type"] = 0, -- Заявка (0: Рыночная, 1: Лимитная) ["order_time"] = 60, -- Срок жизни заявки в секундах ["order_base_price"] = 0, -- Базовая цена заявки закрытия позиции (0:Рыночная, 1:Бид/Оффер, 2:Оффер/Бид, 3:Open, 4:High/Low, 5:Close, 6:Median, 7:Typical, 8:Фикс.) ["order_spread"] = 0, -- Защитный спрэд ["spread_units"] = 0, -- Ед. спреда (0: проценты, 1: шаги цены) ["bar_number"] = 2, -- Номер свечи ["chart_id"] = "NEW_PRICE", -- Идентификатор графика цены ["fix_price_long"] = 0, -- Фиксированная цена заявки закрытия Лонга ["fix_price_short"] = 0, -- Фиксированная цена заявки закрытия Шорта } </pre>

№	Параметры инструмента
	<pre> ["risk_enabled"] = false, -- Ограничение позиции по размеру стоп-лосс (включено) (true/false) ["base_equity"] = 100000, -- Базовый капитал (руб.) ["pos_limit"] = 80, -- Лимит позиции (%) ["trade_risk"] = 1.2, -- Риск трейда (%) ["risk_indicator"] = 0, -- Тип стоп-индикатора (0: ATR, 1: Parabolic SAR, 2: Moving Average, 3: Tenkan Ichimoku, 4: Kijun Ichimoku, 5: Price Channel, 6: Envelopes, 7: Bollinger Bands) ["risk_atr"] = 0, -- Коэффициент ATR, число свечей ["risk_offset"] = 0, -- Отступ от индикатора ["risk_offset_units"] = 1, -- Ед. отступа (0: проценты, 1: шаги цены) ["risk_identifier"] = "NEW_STOP", -- Идентификатор стоп-лосс ["range_enabled"] = false, -- Диапазон риска стоп-лосс (включено) (true/false) ["range_units"] = 1, -- Ед. диапазона (0: проценты, 1: шаги цены) ["range_min"] = 10, -- Минимум (нижняя граница диапазона) ["range_max"] = 25 -- Максимум (верхняя граница диапазона) } </pre>
8.	Заявки <pre> ["Orders"] = { ["quantity"] = 1, -- Количество (лоты) в окне Заявки ["steps"] = 1, -- Шаги в окне Заявки ["value"] = 100, -- Объем в окне Заявки ["lad_num"] = 1, -- Заявки (лестница) в окне Заявки ["lad_dir"] = 1, -- Направление (0:За ценой, 1:Против цены) ["lad_cp_orders"] = false, -- Сразу создавать заявки против цены (true/false) ["lad_first_qty"] = 1, -- Лоты первой заявки ["lad_koef"] = 1, -- Коэффициент лотов ["lad_first_price"] = 0, -- Цена первой ступени ["lad_step"] = 1, -- Шаг ступени ["lad_unit"] = 0, -- Ед. шага ступени ["lad_offset"] = 0, -- Отступ цены заявки ["lad_offset_unit"] = 0, -- Ед. отступа цены заявки ["lad_kill_stops"] = false, -- Снимать стоп-заявки с новым шагом лестницы (true/false) ["lad_time_kill"] = false, -- Снимать лестницы при снятии заявки после создания (true/false) ["check_lots"] = true, -- Проверять доступные лоты (true/false) ["wait_open_otime"] = 3, -- Пауза после создания заявки для открытия позиции ["wait_close_otime"] = 3, -- Пауза после создания заявки для закрытия позиции ["kill_enabled"] = false, -- Снимать заявки через kill_after секунд (true/false) ["kill_after"] = 30, -- Секунды жизни заявки ["partially_filled"] = true, -- Снимать частично исполненные (true/false) ["iceberg_qty"] = 1, -- Лоты ВСЕГО айсберг заявки ["iceberg_visqty"] = 1 -- ВИДИМЫЕ лоты айсберг заявки } </pre>
9.	Стоп-заявки <pre> ["Stoporders"] = { ["auto_stop"] = false, -- Автоматическое управление стоп-заявками (true/false) ["breakeven"] = false, -- Стоп-лосс в безубыток (true/false) ["kill_stop"] = true, -- Снимать стоп-заявки, если нет открытой позиции (true/false) ["dir_control"] = true, -- Снимать стоп-заявки с операцией в направлении позиции (true/false) ["lots_control"] = true, -- Выравнивать лоты стоп-заявок с лотами позиции (true/false) ["stop_indicator"] = false, -- Стоп-лосс по индикатору (true/false) ["take_indicator"] = false, -- Тейк-профит по индикатору (true/false) ["so_count"] = 0, -- Число стоп-заявок (0: Одна стоп-заявка, 1: Пул стоп-заявок, 2: По сделкам) } </pre>

№	Параметры инструмента
	<pre> ["so_type"] = 0, -- Тип стоп-заявки (0: Stop-limit, 1: Trailing stop-limit, 2: Take-profit, 3: Take & stop, 4: Trailing stop & take, 5: Stop with linked, 6: Trailing stop & linked) ["expiry"] = 0, -- Срок действия (0: до отмены, 1: сегодня, 2: дни) ["expiry_days"] = 30, -- Число дней действия стоп-заявки ["stoploss"] = 0, -- Размер стоп-лосс для расчета стоп-цены ["so_units"] = 0, -- Ед. измерения (0: проценты, 1: валюта(шаги)) ["stopspread"] = 0, -- Защитный спрэд для расчета цены ["slspread_units"] = 2, -- Ед. измерения (0: проценты, 1: валюта(шаги), 2: рыночная заявка) ["threshold"] = 0, -- Порог для скользящего стоп-лосс ["deviation"] = 0, -- Шаг рыночной цены для скользящего стоп-лосс ["shift"] = 0, -- Шаг стоп-цены для скользящего стоп-лосс ["take"] = 0, -- Размер тейк-профит для расчета тейк-цены ["offset"] = 0, -- Отступ тейк-профит ["offset_units"] = 0, -- Ед. измерения (0: проценты, 1: валюта(шаги)) ["spread"] = 0, -- Защитный спрэд тейк-профит ["spread_units"] = 0, -- Ед. измерения (0: проценты, 1: валюта(шаги), 2: рыночная заявка) ["br_type"] = 0, -- Тип стопа в безубыток (0: порог, 1: время, 2: порог и время, 3: порог или время) ["br_threshold"] = 0, -- Порог рыночной цены относительно цены позиции для стоп-лосс в безубыток ["br_shift"] = 0, -- Смещение стоп-цены относительно цены позиции для стоп-лосс в безубыток ["br_time"] = 0, -- Время в секундах после открытия позиции для смещения стоп-цены ["so_indicator"] = 0, -- Тип стоп-индикатора (0: ATR, 1: Parabolic SAR, 2: Moving Average, 3: Tenkan Ichimoku, 4: Kijun Ichimoku, 5: Price Channel, 6: Envelopes, 7: Bollinger Bands) ["so_atr"] = 0, -- Коэффициент ATR для стоп-лосс ["so_offset"] = 0, -- Отступ от стоп-индикатора ["so_identifier"] = "NEW_STOP", -- Идентификатор стоп-лосс ["so_ind_always"] = false, -- Всегда по индикатору (true/false) ["so_ind_step"] = 5, -- Всегда с шагом ["tp_indicator"] = 0, -- Тип тейк-индикатора (0: ATR, 1: Parabolic SAR, 2: Moving Average, 3: Tenkan Ichimoku, 4: Kijun Ichimoku, 5: Price Channel, 6: Envelopes, 7: Bollinger Bands) ["tp_atr"] = 0, -- Коэффициент ATR для тейк-профит ["tp_offset"] = 0, -- Отступ от тейк-индикатора ["tp_identifier"] = "NEW_TAKE", -- Идентификатор тейк-профит ["tp_ind_always"] = false, -- Всегда по индикатору (true/false) ["tp_ind_step"] = 5 -- Всегда с шагом } </pre>
10.	Стратегии <pre> ["Strategy"] = { ["enabled"] = false, -- Торговля по стратегии (включено) (true/false) ["name"] = "", -- Имя стратегии ["version"] = "", -- Версия стратегии ["o_begin"] = "000000", -- Время начала условия (00:00:00) ["o_end"] = "235959", -- Время окончания условия (23:59:59) ["c_begin"] = "000000", -- Время начала условия (00:00:00) ["c_end"] = "235959", -- Время окончания условия (23:59:59) ["posopen_type"] = 0, -- Открытие позиций (0: Выключено, 1: Лонг, 2: Шорт, 3: Лонг/Шорт) ["posclose_type"] = 0, -- Закрытие позиций (0: Выключено, 1: Лонг, 2: Шорт, 3: Лонг/Шорт) ["long_size"] = 1, -- Размер Лонга ["short_size"] = 1, -- Размер Шорта ["lsize_type"] = 1, -- Тип размера позиции Лонга (0: Объем, 1: Лоты) ["ssize_type"] = 1, -- Тип размера позиции Шорта (0: Объем, 1: Лоты) ["skip_1_signal"] = true, -- Пропуск сигнала стратегии при старте (true/false) ["pos_reverse"] = false, -- Реверс позиции (true/false) ["inverse_signal"] = false, -- Инверсия сигналов стратегии (true/false) ["otc_signal"] = false, -- Закрытие по сигналам открытия (true/false) ["unlock_signal"] = false, -- Снятие блокировки заявок открытия позиции (true/false) ["one_trade_bar"] = false, -- Один трейд за бар (true/false) ["close_on_bar_end"] = false, -- Сделки на закрытии бара (true/false) } </pre>

№	Параметры инструмента
	<pre> ["otype_bar_end"] = 2, -- Тип заявок на закрытии бара (0: Открытие, 1: Закрытие, 2: Открытие/Закрытие) ["sec_to_bar_end"] = 5, -- Секунды до закрытия бара ["trade_on_lastbar"] = false, -- Торговля по данным на незакрытом баре (true/false) ["is_fulltrades_limit"] = false, -- Лимит трейдов за текущую торговую дату (true/false) ["fulltrades_limit"] = 1000, -- Лимит трейдов за текущую торговую дату (number) ["is_opentime_limit"] = false, -- Время действия сигнала открытия позиции (true/false) ["opentime_limit"] = 3600, -- Время действия сигнала открытия позиции (секунды) (number) ["is_opensignal_limit"] = false, -- Лимит сигналов открытия за текущую торговую дату (true/false) ["opensignal_limit"] = 5000, -- Лимит сигналов открытия за текущую торговую дату (number) ["poslimit_enabled"] = false, -- Лимит увеличения позиции по новым сигналам (true/false) ["poslimit_type"] = 0, -- Тип увеличения позиции (0: Увеличение, 1: Уменьшение) ["new_pos_limit"] = 0, -- Лимит увеличения позиции по новым сигналам (число увеличений) ["new_pos_size"] = 0, -- Размер увеличения позиции ["new_size_type"] = 1, -- Тип размера увеличения позиции (0: Объем, 1: Лоты, 2: Коэффициент) ["run_on_timer"] = true, -- Запуск стратегии по таймеру (true/false) ["o_order_type"] = 0, -- Заявка открытия позиции (0: Рыночная, 1: Лимитная) ["o_order_time"] = 60, -- Срок жизни заявки в секундах ["o_base_price"] = 0, -- Базовая цена заявки открытия позиции (0:Рыночная, 1:Бид/Оффер, 2:Оффер/Бид, 3:Open, 4:High/Low, 5:Close, 6:Median, 7:Typical, 8:Фикс.) ["o_order_spread"] = 0, -- Защитный спрэд ["o_spread_units"] = 0, -- Ед. спрэда (0: проценты, 1: шаги цены) ["o_bar_number"] = 2, -- Номер свечи ["o_chart_id"] = "NEW_PRICE", -- Идентификатор графика цены ["o_fix_price_long"] = 0, -- Фиксированная цена заявки открытия Лонга ["o_fix_price_short"] = 0, -- Фиксированная цена заявки открытия Шорта ["o_resend"] = false, -- Перевыставлять заявки открытия (true/false) ["o_retimes"] = 5, -- Перевыставлять заявки открытия число раз ["o_relast"] = false, -- По цене предыдущей заявки открытия (true/false) ["o_algo_orders_enabled"] = false, -- Создать алгоритмическую заявку (true/false) ["o_algo_order"] = 0, -- Тип алгоритмической заявки (0:Айсберг заявка, 1:Лестница заявок) ["o_correct_to_max"] = false, -- Корректировать лоты заявки на доступные лоты (true/false) ["c_order_type"] = 0, -- Заявка закрытия позиции (0: Рыночная, 1: Лимитная) ["c_order_time"] = 60, -- Срок жизни заявки в секундах ["c_base_price"] = 0, -- Базовая цена заявки закрытия позиции (0:Рыночная, 1:Бид/Оффер, 2:Оффер/Бид, 3:Open, 4:High/Low, 5:Close, 6:Median, 7:Typical, 8:Фикс.) ["c_order_spread"] = 0, -- Защитный спрэд ["c_spread_units"] = 0, -- Ед. спрэда (0: проценты, 1: шаги цены) ["c_bar_number"] = 2, -- Номер свечи ["c_chart_id"] = "NEW_PRICE", -- Идентификатор графика цены ["c_fix_price_long"] = 0, -- Фиксированная цена заявки закрытия Лонга ["c_fix_price_short"] = 0, -- Фиксированная цена заявки закрытия Шорта ["c_resend"] = true, -- Перевыставлять заявки закрытия (true/false) ["c_retimes"] = 99, -- Перевыставлять заявки закрытия число раз ["c_relast"] = false, -- По цене предыдущей заявки закрытия (true/false) ["c_algo_orders_enabled"] = false, -- Создать алгоритмическую заявку (true/false) ["c_algo_order"] = 0 -- Тип алгоритмической заявки (0:Айсберг заявка, 1:Лестница заявок) ["is_sndLong"] = false, -- Признак воспроизведения звуковых файлов при сигнале Лонг ["snd_longFile"] = "C:\\Windows\\Media\\Alarm07.wav", -- Полный путь к файлу сигнала Лонг ["snd_longInitialDir"] = "C:\\Windows\\Media", -- Папка файла сигнала Лонг ["is_sndShort"] = false, -- Признак воспроизведения звуковых файлов при сигнале Шорт ["snd_shortFile"] = "C:\\Windows\\Media\\Alarm07.wav", -- Полный путь к файлу сигнала Шорт ["snd_shortInitialDir"] = "C:\\Windows\\Media", -- Папка файла сигнала Шорт ["is_sndCloseLong"] = false, -- Признак воспроизведения звуковых файлов при сигнале Закрыть Лонг ["snd_closeLongFile"] = "C:\\Windows\\Media\\Alarm04.wav", -- Полный путь к файлу сигнала Закрыть Лонг ["snd_closeLongInitialDir"] = "C:\\Windows\\Media", -- Папка файла сигнала Закрыть Лонг </pre>

№	Параметры инструмента
	<pre> ["is_sndCloseShort"] = false, -- Признак воспроизведения звуковых файлов при сигнале Закрыть Шорт ["snd_closeShortFile"] = "C:\\Windows\\Media\\Alarm04.wav", -- Полный путь к файлу сигнала Закрыть Шорт ["snd_closeShortInitialDir"] = "C:\\Windows\\Media", -- Папка файла сигнала Закрыть Шорт ["snd_playCount"] = 1, -- Число повторов звукового файла ["snd_playSeconds"] = 10 -- Пауза между стратами воспроизведения звуковых файлов в секундах } </pre>

4.2 Торговые параметры

Таблица торговых параметров инструмента имеет структуру, описанную в Табл. 4.2.

Табл. 4.2 – Таблица торговых параметров инструмента

№	Параметры инструмента
1.	Группа параметров инструмента <pre> ["Security"] = { ["incorrect_data"] = false, -- (boolean) Признак не корректных данных (true: данные некорректны, false: корректны) ["status"] = 0, -- (number) Статус (0: выключено, 1: торгуется, 2: по 1-му времени, 3: по 2-му времени, 4: по 3-му времени, 5: остановка на клиринг, 6: остановка по результату, 7: по времени в позиции) ["prev_status"] = -1, -- (number) Предыдущий статус инструмента ["session_is_open"] = false, -- (boolean) Сессия (true: открыта, false: закрыта) ["session_val"] = -1, -- (number) Сессия числом (-1, 1 и т.п.) ["session_img"] = "", -- (string) Сессия строкой (открыта, завершена и т.п.) ["exchange"] = -1, -- (number) Тип рынка (-1: неопределен, 0: Валютный, 1: ММВБ, 2: ФОРТС) ["qty"] = 0, -- (number) Текущий размер позиции в лотах ["start_qty"] = 0, -- (number) Размер позиции на начало торгов ["prev_qty"] = 0, -- (number) Предыдущий размер позиции в лотах (для лестницы) ["trade_price"] = 0, -- (number) Цена позиции ["prev_price"] = 0, -- (number) Предыдущая цена позиции ["last_price"] = 0, -- (number) Цена последней сделки ["step_price"] = -1, -- (number) Шаг цены ["sec_scale"] = -1, -- (number) Точность цены ["steppricet"] = -1, -- (number) Стоимость шага цены ["lot_size"] = 1, -- (number) Размер лота для ММВБ ["ex_koef"] = 1, -- (number) Биржевой коэффициент ["maxbuy"] = 0, -- (number) Лоты доступные к покупке ["maxsell"] = 0, -- (number) Лоты доступные к продаже ["pos_open_time"] = 0, -- (number) Время открытия позиции ["is_time_closed"] = -1, -- (number) Флаг закрытия позиции по времени или фин.результату (-1: нет, 2,3,4: по времени, 6: по фин.рез., 7: по времени в позиции) ["trades_count"] = 0 -- (number) Количество полных трейдов } </pre>
2.	Группа финансового результата <pre> ["Profit"] = { ["pips_profit"] = 0, -- (number) Результат открытой позиции в пунктах ["currency_profit"] = 0, -- (number) Результат торговли за текущий день в валюте ["max_profit"] = 0, -- (number) Максимальный полученный доход ["max_profit_gained"] = false, -- (boolean) Флаг достижения Max profit из настроек (false: нет, true: да) ["num_trades"] = 0, -- (number) Число сделок инструмента ["num_lots"] = 0, -- (number) Число лотов по сделкам инструмента ["prev_qty"] = 0, -- (number) Предыдущий размер позиции в лотах ["prev_profit"] = 0, -- (number) Предыдущий финансовый результата в рублях ["total_comm"] = 0, -- (number) Суммарная комиссия ["ex_koef"] = -1 -- (number) Средневзвешенный Биржевой коэффициент по сделкам } </pre>

№	Параметры инструмента
3.	["ACTIVE_ORDERS"] = {} -- (table) Таблица активных заявок по инструменту
4.	["ACTIVE_STOPORDERS"] = {} -- (table) Таблица активных стоп-заявок по инструменту
5.	["TRADES"] = {} -- (table) Таблица сделок по инструменту
6.	["ORDERS"] = {} -- (table) Таблица всех заявок по инструменту
7.	["STOPORDERS"] = {} -- (table) Таблица всех стоп-заявок по инструменту
8.	["SOpool"] = {} -- (table) Таблица пулов стоп-заявок