

# Раздел 8. Алгоритмический язык QPILE

8.1 Общие положения .....	2
8.2 Работа с таблицами QPILE .....	4
8.3 Структура файла программы .....	10
8.4 Конструкции языка QPILE.....	14
8.5 Общие функции.....	24
8.6 Математические функции .....	26
8.7 Функции для работы с коллекциями объектов (COLLECTION).....	28
8.8 Функции для работы с ассоциативными массивами (MAP) .....	30
8.9 Функции для обращения к строкам произвольных таблиц QUIK.....	32
8.10 Функции для обращения к спискам доступных параметров.....	58
8.11 Функции для работы с программируемой таблицей.....	59
8.12 Функции для получения значений Таблицы текущих торгов .....	63
8.13 Функции для получения значений Окна котировок.....	71
8.14 Функции для получения значений таблицы «Позиции по инструментам» .....	71
8.15 Функции для получения значений таблицы «Позиции по деньгам» .....	74
8.16 Функции расчета маржинальных позиций .....	75
8.17 Функции получения значений таблиц «Клиентский портфель» и «Купить/Продать» .....	77
8.18 Функции для работы с файлами .....	85
8.19 Функции для работы со строками.....	87
8.20 Функции для работы с графиками .....	88
8.21 Функции для работы с заявками .....	92
8.22 Функции для работы с метками .....	93
8.23 Сервисные функции.....	96
8.24 Отладка программ QPILE .....	100
ПРИЛОЖЕНИЕ 1. Синтаксис команд языка QPILE .....	102
ПРИЛОЖЕНИЕ 2. Рекомендации к составлению программ на QPILE.....	104

Раздел описывает применение алгоритмического языка QPILE, встроенного в Рабочее место системы QUIK. В начале раздела рассматривается базовый функционал клиентского места, предназначенный для работы с программируемыми таблицами. Для пользователей, заинтересованных в разработке собственных программ в документации приводится описание конструкций языка QPILE и встроенных функций языка, а также пример реализации – расчет показателей маржинального кредитования по методике ФСФР.

## 8.1 Общие положения

### 8.1.1 Назначение

**Язык QPILE** (QUIK Programmable Interface and Logic Environment) представляет собой набор команд, интерпретируемых рабочим местом QUIK.

Областью применения QPILE является создание новых таблиц для расчета в режиме реального времени собственных показателей на основе информации из других таблиц.

Данный функционал удобен, прежде всего, для брокера, так как у каждого брокера существует своя собственная стратегия расчета показателей клиентской позиции. Применение встроенного языка позволяет реализовать практически любой алгоритм.

Примеры использования QPILE:

1. Динамический пересчет стоимости активов клиентов на рабочем месте брокера.
2. Динамический пересчет стоимости активов в портфеле клиента и его суммарной стоимости.
3. Вычисление показателей, отсутствующих в системе QUIK либо торговой системе биржи, на основе собственных алгоритмов.
4. Расчет параметров маржинального кредитования в соответствии с принятыми нормативами.
5. Программирование торговой стратегии, генерирующей сигналы на изменение позиций по инструментам.

**Основной областью применения языка QPILE является расчет стоимости портфелей ценных бумаг, поэтому в терминологии описания типов таблиц часто используется определение «Портфели».**

### 8.1.2 Как работает QPILE

1. Структура таблицы (назначение столбцов, строк, и формулы для вычисления параметров) описывается в виде программы на языке QPILE. Для вычисления в формулах можно использовать стандартные математические и логические операции, использовать переменные и массивы данных, запрашивать информацию из других таблиц системы QUIK.

2. Код программы может быть получен с сервера QUIK (**серверный**) или загружен с диска на компьютере пользователя (**локальный**). Полученный код обрабатывается интерпретатором языка QPILE на рабочем месте QUIK, который производит пересчет значений в формулах с заданной периодичностью. Таким образом, формируется внутренний источник данных для отображения полученных значений в таблицах. Несколько таблиц на основе одной и той же программы используют общий источник данных, при этом вычисления не будут дублироваться и снижать производительность системы.
3. На основе программы можно создавать таблицы, обладающие стандартными функциями при работе с таблицами в системе QUIK.
4. Рабочее место QUIK имеет встроенный отладчик кода программ на QPILE, позволяющий вести пошаговую отладку и контролировать текущие значения переменных. Подробнее см. п. [8.24](#).

**Нумерация строк в таблицах QPILE начинается с единицы, а нумерация символов в строках языка QPILE начинается с нуля.**

### 8.1.3 Основные возможности

Основные функции языка QPILE:

1. Описание новых таблиц произвольной структуры,
2. Вычисление полей таблиц на основе математических формул и логических выражений,
3. Выделение ячеек таблицы цветом в зависимости от значений какого-либо показателя,
4. Уведомление пользователя звуковым сигналом и текстовыми сообщениями.

Таблица на языке QPILE поддерживает основные действия с таблицами в системе QUIK:

- Редактирование, в том числе выбор параметров для отображения и настройку их очередности,
- Управление «горячими клавишами»,
- Размещение на «экранных вкладках»,
- Поиск значения в ячейках таблицы,
- Печать таблицы с предварительным просмотром,
- Копирование данных в Буфер Обмена Windows (Clipboard),
- Экспорт данных в Excel,
- Экспорт данных по ODBC.

В качестве **исходных данных** для расчетов параметров таблиц может использоваться информация из следующих типов таблиц QUIK, а также из графиков:

№ Таблица	№ Таблица
1 Таблица текущих торгов	10 Таблица заявок на внебиржевые сделки
2 Таблица обезличенных сделок	11 Таблица сделок для исполнения
3 Таблица заявок	12 Таблица заявок-отчетов на сделки РПС
4 Таблица стоп-заявок	13 Таблица «Клиентский портфель»
5 Таблица сделок	14 Таблица «Купить/Продать»
6 Позиции по инструментам	15 Позиции участника по деньгам
7 Позиции по деньгам	16 Позиции участника по инструментам
8 Позиции по клиентским счетам	17 Позиции участника по торговым счетам
9 Ограничения по клиентским счетам	18 Таблица, создаваемая при расчете программы

#### 8.1.4 Функциональные ограничения

Текущая версия языка QPILE и созданные на его основе таблицы не поддерживают следующих операций:

- Сортировку в таблицах;
- Применение фильтров;
- Использование таблицы в качестве источника данных для построения графиков;
- Сохранение данных из таблицы в текстовый файл через контекстное меню;
- Экспорт данных в системы технического анализа.

## 8.2 Работа с таблицами QPILE

### 8.2.1 Загрузка программы

Этот этап добавляет описания собственных таблиц в список доступных типов. Если предполагается работа с описаниями таблиц на сервере, его можно пропустить.

Загрузка кода программы выполняется выбором пункта меню программы **Сервисы/ QPILE скрипты...**, или нажатие клавиш «Ctrl»+«F11».

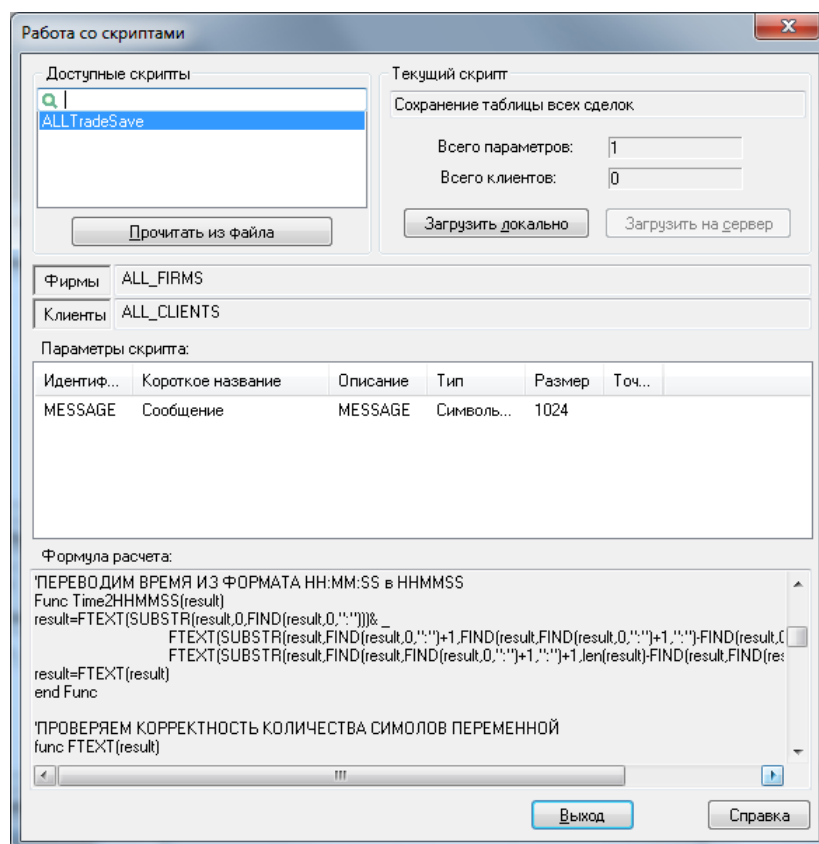
1. Нажмите кнопку «Добавить» и выберите файл, загружаемый с диска. Файлы имеют стандартные расширения «.QPL». Название таблицы появится в списке «Доступные скрипты».
2. При чтении файла осуществляется проверка правильности программы. Если программа содержит ошибки, система уведомит об этом в «Окне сообщений». В сообщении указывается имя файла и номер строки, в которой обнаружена ошибка.

3. Если файл прочитан успешно, то в полях отобразятся соответствующие параметры:

Поле	Назначение
Текущий скрипт	Название таблицы
Всего параметров	Число параметров (столбцов), описанных в структуре таблицы
Всего клиентов	Количество доступных кодов клиентов, из числа перечисленных в структуре таблицы. В новой версии языка параметр не актуален
Фирмы	Перечень идентификаторов фирмы в торговой системе (значения, соответствующие полю «Дилер» в Таблице заявок или Таблице сделок)
Клиенты	Перечень идентификаторов клиентов для отображения в таблице
Параметры скрипта	Перечень описанных параметров и расширенные сведения по ним
Формула расчета	Исходный код программы на языке QPILE

4. Для загрузки прочитанной из файла таблицы нажмите кнопку «Загрузить локально».

5. Кнопка «Загрузить на сервер» позволяет загрузить программу на сервер, которая станет доступна всем его пользователям. Право загрузки программ на сервер назначается администратором системы QUIK.



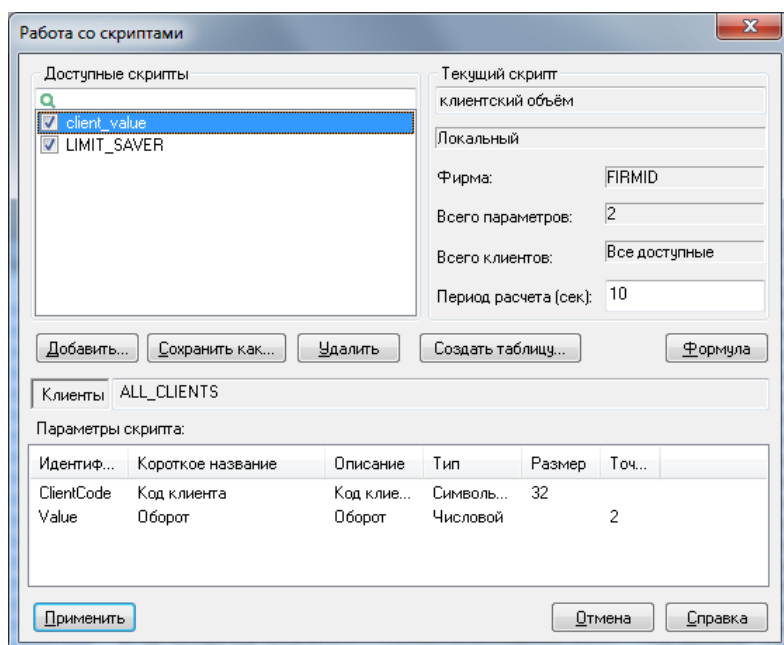
При локальной загрузке код выполняется на текущем рабочем месте и просмотр таблицы на его основе доступен тоже только на этом месте.

**Первую загрузку каждого нового описания настоятельно рекомендуется производить локально, а уже после просмотра и проверки корректности его работы и расчета параметров – загружать на сервер.**

### 8.2.2 Настройка параметров программы

Этот этап определяет список обрабатываемых программ и определяет периодичность вычислений. Если программы были загружены локально, то они автоматически помечаются выбранными.

Выбор доступных программ с описаниями типов таблиц выполняется выбором пункта меню программы **Сервисы/QPILE скрипты...**, или нажатием клавиш «Ctrl»+«F11».



1. В списке «Доступные скрипты» приведены все программы, доступные пользователю для создания таблиц, как загруженные локально, так и серверные. Выберите программу из числа доступных, установив на ней флажок. В полях диалога отобразятся параметры, относящиеся к структуре программируемой таблицы.
2. Установите период расчета значений таблицы. Для исполнения программ с большим количеством вычислений период расчета значений рекомендуется устанавливать не менее 5-10 секунд.
3. Нажмите кнопку «Сохранить как...» для сохранения настроек.
4. Нажатие кнопки «Удалить» приводит к удалению из списка выбранной строки. Серверные типы таблиц удалить нельзя.
5. Нажатием кнопки «Формула» открывается окно с исходным кодом программы и описанием таблицы (справочно).

6. Нажатие кнопки «Создать таблицу» открывается окно «Создать таблицу для просмотра параметров скрипта» (см. п. [8.2.3](#)).
7. Чтобы закрыть окно с сохранением настроек, нажмите кнопку «Применить». Нажатие кнопки «Отмена» закрывает окно без сохранения изменений.

**Настройки данного этапа распространяются и на ранее созданные программируемые таблицы. Этот этап можно использовать для включения/отключения вычислений в таблицах, а также для регулирования частоты обновления данных в них.**

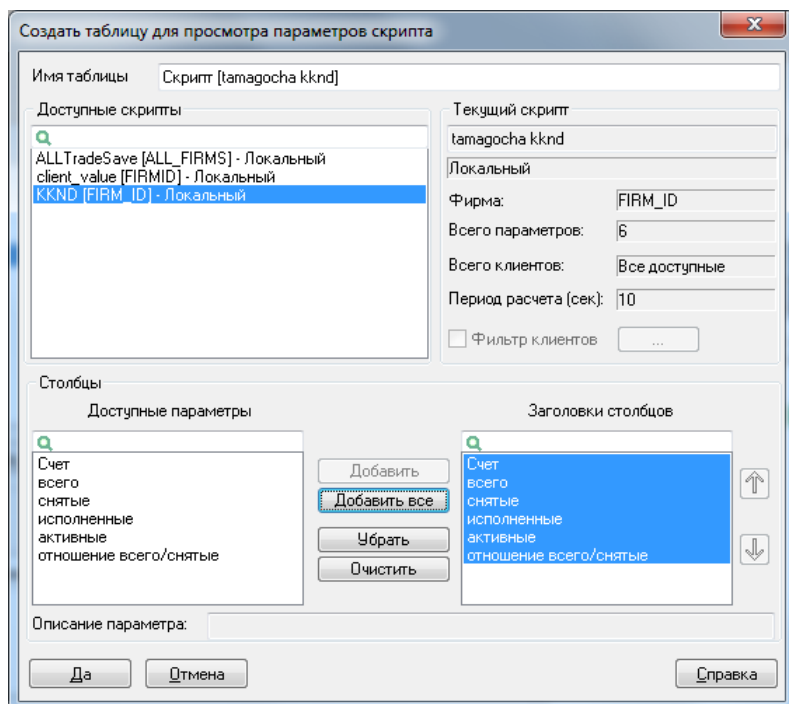
Применение настроек к существующим таблицам:

- Как отключить пересчет значений в таблице?
  - Выберите пункт контекстного меню «Приостановить расчет» либо нажмите «Ctrl»+«F11» и снимите флажок с требуемого типа таблицы. При этом в таблице будут отображены последние вычисленные значения.
  - Если по одной программе было сформировано несколько таблиц, то расчет приостановится во всех этих таблицах.
  - Для отключения пересчета во всех таблицах нужно нажать «Ctrl»+«F11» и снять все флажки.
- Как изменить период расчета в существующей таблице?
  - Выберите пункт контекстного меню «Параметры скрипта» или нажмите «Ctrl»+«F11», затем измените значение параметра «Период расчета (сек)» и нажмите кнопку «Применить».


### 8.2.3 Создание таблицы

Этот этап формирует таблицу на основании программы. На базе одной программы можно создать несколько таблиц для удобства их расположения на экране.

Создание программируемой таблицы выполняется нажатием кнопки «Создать таблицу» в окне «Работа со скриптами» (см. п. [8.2.2](#)).



1. Выберите программу в списке «Доступные скрипты». В полях раздела «Текущий скрипт» отобразятся данные по структуре таблицы.
2. Если необходимо, ограничьте количество отображаемых строк в таблице с помощью фильтра «Фильтр клиентов».
3. Сформируйте список столбцов таблицы из числа доступных параметров и последовательность их отображения в таблице. В поле «Описание параметра» справочно приводится расширенное описание выделенного параметра.
4. Нажатием кнопки «Да» создается таблица.

**Полученная таблица имеет такие же функции управления, как и другие таблицы системы QUIK. Например, для редактирования таблицы можно нажать кнопку  на панели инструментов, или клавиши «Ctrl» + «E». Расчет параметров таблицы ведется, в том числе на основе Таблицы текущих торгов. Убедитесь, что необходимые для расчета параметров данные принимаются с сервера (не отключены фильтром в списке принимаемых параметров и инструментов).**

#### 8.2.4 Функции, доступные в таблице

Данные таблицы доступны для копирования, экспорта в Excel, экспорта через ODBC.

Функции, доступные из контекстного меню таблицы:

- «Приостановить расчет» – остановить вычисление параметров таблицы (при этом в таблице отображаются последние вычисленные данные);



- «Возобновить расчет» – возобновляет вычисление параметров таблицы, если он был остановлен;
- «Начать расчет заново» – сбросить все значения и начать расчет заново;
- «Начать расчет в режиме отладки» – сбрасывает все значения, запускает окно отладки и начинает расчет заново;
- «Сохранить описание в файл» – сохранить в текстовый файл программу, описывающую данную таблицу;
- «Посмотреть формулу» – показать в окне код программы данной таблицы;
- «Параметры скрипта» – открыть окно с параметрами программы, в том числе период расчета в секундах.

Описание стандартных функций контекстного меню для таблиц см. п. 2.8.4 Раздела 2 «Основные принципы работы с программой».

Описание применения пользовательских фильтров и условного форматирования таблиц приведено соответственно в пп. 2.8.8 и 2.8.9 Раздела 2 «Основные принципы работы с программой».

Полный список управляющих клавиш для всех типов таблиц приведен в Приложении к Разделу 2.

**ПРИМЕР:** Локальная загрузка файла программы с расчетом показателей маржинального кредитования по методике ФСФР.

Пример находится в одном архиве с этим Руководством пользователя, в папке «qpile».

**Перед загрузкой файла, приведенного в примере, измените код участника торгов в строках 4 и 60 на код фирмы Вашего брокера.**

1. Нажатием клавиш «Ctrl»+«F11» открывается окно, в котором выбираем требуемый файл на диске (в примере это fksb.qpl) и нажимаем кнопку «Открыть».
2. Нажимаем кнопку «Загрузить локально». В окне сообщений появляется уведомление о прочтении файла. Нажимаем кнопку «Выход».
3. Следующий этап пропускаем, потому что загруженные локально описания таблиц автоматически отмечаются доступными для создания таблиц.
4. Нажатием кнопки «Создать таблицу» открывается окно создания таблицы. Формируем список «Заголовки столбцов» выбором параметров из числа доступных.
5. Нажатием кнопки «Да» создается таблица, в строках которой перечислены доступные счета клиентов, со следующими параметрами в столбцах:

Название поля	Значение
Задолженность брокеру	Сумма задолженности клиента брокеру по деньгам и инструментам, в денежном выражении
Только деньги клиента	Сумма всех текущих остатков клиента по деньгам
Оценка	Сумма всех текущих положительных остатков клиента по инструментам, в денежном выражении
Средства в рублях	Доступные собственные средства клиента
Все средства	Текущая стоимость средств клиента, сумма параметров «Оценка» и «Средства в рублях»
Уровень маржи	Текущий уровень маржи, рассчитанный по методике ФСФР
Состояние	Текстовый комментарий, поясняющий значение уровня маржи

В ячейках таблицы появятся данные, пересчитываемые с заданной регулярностью.

Если данные в таблице отсутствуют, то, скорее всего, в коде программы указаны недоступные идентификатор фирмы, коды инструментов или коды клиентов.

При необходимости можно изменить периодичность расчета или сократить список кодов клиентов с помощью фильтра, выбрав пункт контекстного меню «Параметры скрипта» или нажав «Ctrl»+«F11».

## 8.3 Структура файла программы

Файл с кодом программы представляет собой текстовый файл в кодировке CP-1251 (Windows). Один файл служит для описания одного типа таблицы. Используются стандартные расширения имени файлов «.QPL».

Описание состоит из трех частей:

1. «Заголовок» – наименование таблицы и определение основных параметров,
2. «Тело программы» – собственно код программы,
3. «Описание столбцов таблицы» – определение названий столбцов и формата соответствующих им ячеек таблицы,

Все разделы являются обязательными и должны следовать строго друг за другом. Файл всегда начинается с заголовка и заканчивается строкой «END\_PORTFOLIO» (в новой версии языка должно применяться «END\_PORTFOLIO\_EX»).

Примеры файлов см. В папке QPILE, которая поставляется вместе с Руководством пользователя.

### 8.3.1 Заголовок файла

Заголовок файла содержит основные параметры таблицы:

Параметр	Назначение
PORTFOLIO	Название таблицы – символьный идентификатор латинскими буквами без пробелов.
PORTFOLIO_EX	«PORTFOLIO» – параметр, применявшийся в первой версии языка, «PORTFOLIO_EX» – обозначение программы на расширенной версии языка (начиная с версии QUIK 4.09)
DESCRIPTION	Текстовое описание таблицы
* CLIENTS_LIST	Список кодов клиентов через запятую, для которых вычисляются значения в таблице. Значение «ALL_CLIENTS» соответствует выбору всех кодов клиентов. Значения по каждому счету клиента формируют новую строку таблицы
FIRMS_LIST	Список идентификаторов фирм (участников торгов) через запятую, клиентам которых будет доступна данная таблица

\* – параметр предыдущей версии языка, в текущей версии является необязательным.

Каждый параметр должен быть описан на отдельной строке и заканчиваться символом «;».

Пример заголовка:

```
PORTFOLIO AVAILABLE_MONEY;  
DESCRIPTION Доступные деньги клиента;  
CLIENTS_LIST ALL_CLIENTS;  
FIRMS_LIST MC0012300000;
```

Данная таблица будет отображаться в списке доступных с именем «AVAILABLE\_MONEY». При выборе доступных описаний таблиц в поле «Текущий скрипт» отобразится «Доступные деньги клиента». Это же наименование будет являться заголовком таблицы, принимаемым по умолчанию. Расчет параметров осуществляется по всем счетам клиентов, доступных пользователю в таблицах позиций и имеющих код фирмы «MC0012300000».

### 8.3.2 Тело программы

Раздел содержит код программы для расчета значений в ячейках таблицы. Начало раздела обозначается строкой «PROGRAM», окончание – строкой «END\_PROGRAM». Для выхода из тела программы используется оператор «RETURN».

- 1. Регистр символов (верхний или нижний) в командах по умолчанию игнорируется интерпретатором. Все строковые константы переводятся**

**автоматически в верхний регистр. Но в случае необходимости можно отменить автоматический перевод символов в верхний регистр. Для этого необходимо между заголовком и телом программы добавить следующую строку:**

```
USE_CASE_SENSITIVE_CONSTANTS
```

При добавлении указанного ключа значения строковых переменных не будут переводиться в верхний регистр, а будут оставаться такими, какие они есть.

- 2. Множественные пробелы игнорируются интерпретатором, за исключением пробелов внутри строковых переменных.**
- 3. Для переноса длинных строк используется комбинация «<пробел>\_» в конце строки. Например:**

```
CLIENTS_LIST 0001, 0002, 0003, 0004, 0005, 0006, 0007, 0008, 0009, 0010, _  
0011, 0012, 0013, 0014, 0015;
```

- 4. Символ «'» является комментарием. Комментарий распространяется до конца строки.**
- 5. Разделителем операторов служит перевод строки, т.е. каждый оператор описывается отдельной строкой.**
- 6. Нумерация символов в строке начинается с нуля.**

Пример программы:

```
PROGRAM  
FirmCode = "MC0012300000"  
CurrentBalance = MONEY_CURRENT_BALANCE(ROWNAME, FirmCode, "EQT", "SUR")  
CurrentLimit = MONEY_CURRENT_LIMIT(ROWNAME, FirmCode, "EQT", "SUR")  
Locked = MONEY_LIMIT_LOCKED(ROWNAME, FirmCode, "EQT", "SUR")  
AvailableMoney = MoneyCurrentBalance + MoneyCurrentLimit - MoneyLocked  
If AvailableMoney > 0  
Status = "Заявки доступны"  
Else  
Status = "Заявки недоступны"  
SET_ROW_COLOR(ROWNAME, "RGB(255,138,138)", "DEFAULT_COLOR")  
End If  
END_PROGRAM
```

Пример относится к формированию таблицы «Доступные деньги клиента». Создаваемая таблица будет частично дублировать поля таблицы «Позиции по деньгам», а также содержать два вычисляемых поля.

### 8.3.3 Определение столбцов таблицы

Для отображения в таблице вычисленных значений необходимо описать ее столбцы и формат соответствующих им ячеек таблицы. Для этого предназначен данный раздел файла.

Каждый столбец таблицы описывается следующим набором параметров:

Параметр	Назначение
PARAMETER	Название переменной в программе (максимальная длина 31 символ), значение которой будет отображаться в данном столбце
PARAMETER_TITLE	Название столбца (максимальная длина 31 символ), отображаемое в таблице
PARAMETER_DESCRIPTION	Расширенное описание параметра (максимальная длина 127 символов)
PARAMETER_TYPE	Формат данных в ячейках таблицы, относящихся к столбцу. Возможны два типа данных: <ul style="list-style-type: none"><li>_ NUMERIC(&lt;размер_числа_в_символах&gt;, &lt;кол_во_знаков_после_запятой&gt;) – двойные с плавающей точкой (double),</li><li>_ STRING(&lt;длина_строки&gt;) – строковые (string)</li></ul>

Описание каждого параметра завершается строкой «END».

После описания параметров в конце файла обязательно должна быть строка «END\_PORTFOLIO» для первой версии языка, либо «END\_PORTFOLIO\_EX» для его новой версии.

Пример описания:

```
PARAMETER AvailableMoney;  
PARAMETER_TITLE Доступно;  
PARAMETER_DESCRIPTION Доступные деньги клиента;  
PARAMETER_TYPE NUMERIC(10,2);  
END
```

Значения переменной «AvailableMoney» будут отображаться в столбце таблицы «Доступно». При настройке таблицы в поле «Описание параметра» отображается текст «Доступные деньги клиента». Данные в столбце будут отображаться в виде числа с двумя знаками после запятой. Этот же формат данных рекомендуется указывать при настройке экспорта из таблицы по ODBC.

### 8.3.4 Подключение дополнительных файлов

Для подключения дополнительных файлов с функциями к программе используется раздел INCLUDE. Данный раздел размещается между заголовком и телом программы и описывается следующим образом:

```
INCLUDE file1, file2, ..., fileN;
```

где file1, file2, ..., fileN – относительные или полные пути файлов с описанием функций (прописываются через запятую).

Пример описания:

```
INCLUDE C:/Program Files/ADDITION/object_1, C:/Program Files/ADDITION/object_2;
```

## 8.4 Конструкции языка QPILE

### 8.4.1 Типы данных

1. Различаются следующие типы данных:

– «STRING» – строковый,

«Суммарный спрос»

– «DOUBLE» – вещественный, или число с плавающей точкой. Точность – 15 десятичных разрядов, но не более 8 знаков после запятой.

1234567.89

– «COLLECTION» – коллекция,

– **Коллекция** является списком объектов, с индексацией по целочисленному ключу (начиная с 0).

0	1	2	3
«HYDR»	12	7.890	«BUY»

– «MAP» – ассоциативный массив,

- **Ассоциативный массив (MAP)** является последовательностью пар (ключ, значение), которая обеспечивает получение значения по ключу. Каждому ключу соответствует максимум одно значение, иными словами, каждый ключ в ассоциативном массиве уникален. В качестве ключа всегда выступает строковое значение.

«NUMBER»	«TIME»	«OPERATION»
67890	«12:34:56»	«BUY»

**2.** Типы данных «COLLECTION» и «MAP» являются структурными. Они могут:

- Быть неоднородными, т.е. содержать значения разных типов,
- Содержать в качестве значений элементы любых типов, в том числе и переменные типов «COLLECTION» и «MAP».

**3.** Область видимости всех переменных, кроме формальных параметров, является глобальной и определяется контекстом исполнения. Если в процессе исполнения переменная была присвоена, то она считается определенной до конца исполнения программы.

Переменные, сохраняющие свои значения между периодами расчета программы, именуются **глобальными**. Глобальные переменные описываются специальной функцией «NEW\_GLOBAL ()» (см. п. [8.5.1](#)).

**4.** Количество используемых в программе переменных ограничено 1000.

#### 8.4.2 Приведение типов

1. Переменные языка не типизированы и могут изменять тип данных в процессе исполнения программы.
2. Применение операций «+», «-», «\*», «/» к строковым переменным приводит их к вещественным значениям. Если такое приведение не возможно, то вещественное значение принимается равным «0.0». Для структурных переменных результат этих операций не определен.
3. Операция сравнения вещественных и строковых переменных допустима только для переменных со значениями одного типа. Для переменных структурных типов «COLLECTION» и «MAP» результат сравнения не определен.
4. Применение операции конкатенации строк «&» к вещественным переменным приводит их значения к строковым.
5. При вызове внешних функций всегда осуществляется приведение типов аргументов.
6. Изменение значений переменных структурных типов и обращение к их полям выполняется специальными функциями, описанными в п. [8.7](#)–[8.8](#).

### 8.4.3 Выражения

1. Приоритет выполнения математических операций («+», «-», «\*», «/», унарный «-») является стандартным.
2. Логические операции имеют одинаковый приоритет (за исключением «AND» и «OR») и исполняются слева направо. «AND» и «OR» исполняются в последнюю очередь, слева направо. Например:

**A < B or A = C равносильно (A < B) or (A = C)**

3. В выражениях и сравнениях возможно использование скобок.  
Допустимые выражения:

Операция	Значение
<b>Математические</b>	
+	Сложение
-	Вычитание
*	Умножение
/	Деление
Унарный «-»	Смена знака в значении на противоположный
D или E	Экспоненциальная форма записи числа, «3D2» равносильно $3 \cdot 10^2$
<b>Логические</b>	
==	Равенство
=	Равенство, аналогично предыдущему
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
!=	Не равно
<>	Не равно, аналогично предыдущему
AND	Логическое «И»
OR	Логическое «ИЛИ»



Операция	Значение
----------	----------

Текстовые	
-----------	--

&	Склейка (конкатенация) строк
---	------------------------------

**При присвоении значения переменной необходимо использовать приведение типов данных. Например, выражение  $A = \text{«3E2»} + 0$  даст результат 300, в отличие от  $A = \text{«5E3»}$ , которым переменной будет присвоено строковое выражение «5E3».**

#### 8.4.4 Условия

Синтаксис оператора условия:

```
IF условие
последовательность инструкций
ELSE
последовательность инструкций
END IF
```

«Условием» является логическое выражение. Вложенность операторов условия не ограничена. В операторе «END IF» допускается только один пробел.

Пример:

```
IF DealerMoney = 0
Margin = 100
ELSE
Margin=0
END IF
```

#### 8.4.5 Циклы

Синтаксис оператора цикла:

1. выполняет «последовательность инструкций» для каждого значения «переменной», входящего в «список значений». «Список значений» определяется переменной типа «STRING» со значениями, перечисленными через запятую.

```
FOR переменная IN список значений
последовательность инструкций
END FOR
```

- 2.** выполняет последовательность инструкций для каждого значения переменной в диапазоне от «значение1» до «значение2» с шагом 1, в качестве которых могут выступать математические выражения. Если «значение2» < «значение1», цикл не обрабатывается.

```
FOR переменная FROM значение1 TO значение2
последовательность инструкций
END FOR
```

Вложенность операторов цикла не ограничена. В операторе «END FOR» допускается только один пробел.

Пример:

```
FkcbSecsList = "HYDR,SBER,MSNG,LKOH,YUKO,RTKM"

FOR Sec IN FkcbSecsList
DCPos = DEPO_CURRENT_BALANCE(ROWNAME, FirmList, Sec, DefDepoAcc)
DCLim = DEPO_CURRENT_LIMIT(ROWNAME, FirmList, Sec, DefDepoAcc)
DOLim = DEPO_OPEN_LIMIT(ROWNAME, FirmList, Sec, DefDepoAcc)
SecPos = DCPos + DCLim - DOLim
SecPos = SecPos * GET_PARAM(ClassCode, Sec, "LAST") * GET_PARAM(ClassCode, Sec,
"LOTSIZE")
DepoPos = DepoPos + ignore_negative(SecPos)
DealerMoney = DealerMoney + dealer(SecPos)
END FOR
```

Данный цикл перебирает все инструменты из списка инструментов «FkcbSecsList» и по каждому инструменту обрабатывает позицию по этому инструменту для текущего рассчитываемого клиента.

Для прерывания цикла до его окончания служит оператор «BREAK». Он прерывает выполнение оператора «FOR» и передает управление следующему за ним оператору.

Для передачи управления на следующую итерацию оператора «FOR» служит оператор «CONTINUE». В операторе «FOR» следующая итерация начинается вычислением выражения цикла оператора «FOR». После вычисления условного выражения в зависимости от его результатов происходит либо прекращение выполнения оператора либо выполнение его тела.

### 8.4.6 Функции

Общие положения:

1. Описания функций могут находиться в любом месте программы.
2. Все функции должны иметь различные имена – перегрузка по типу параметров и их количеству не осуществляется.
3. Функции могут возвращать значение с помощью переменной с именем «RESULT».
4. Все параметры передаются в функции по значению.
5. Функция может являться процедурой. В этом случае в месте вызова такой функции не используется возвращаемое значение, и присваивание переменной с именем «RESULT» внутри такой функции не обязательно.
6. Для выхода из тела функции может использоваться оператор «RETURN».

1. В описании синтаксиса функций, возвращающих или принимающих переменную любого типа, для обозначения применяется тип «ANY».
2. Таблица, создаваемая при расчете программы, обозначается «OWN».

Функции, определенные пользователем, имеют следующий синтаксис:

```
FUNC функция (список аргументов)
последовательность инструкций
END FUNC
```

В операторе «END FUNC» допускается только один пробел.

Пример:

```
FUNC ignore_negative(x)
If x > 0
RESULT = x
Else
RESULT = 0
End If
END FUNC
```

### 8.4.7 Функции языка QPILE

Стандартные функции языка QPILE предназначены для работы со структурными типами переменных и получения значений из таблиц QUIK.

Функция	Назначение
NEW_GLOBAL	Инициализация глобальной переменной
MESSAGE	Вывод текста в Окне сообщений

### Математические функции

ABS	Модуль
ACOS	Арккосинус
ASIN	Арсинус
ATAN	Арктангенс
CEIL	Округление вверх
COS	Косинус
EXP	Экспонент
FLOOR	Округление вниз
LOG	Логарифм
POW	Возведение в степень
RAND	Случайное число
RANDOMIZE	Генерация случайных чисел
SIN	Синус
SQRT	Квадратный корень числа
TAN	Тангенс

### Функции для работы с коллекциями объектов

CREATE_COLLECTION	Определение коллекции
GET_COLLECTION_COUNT	Количество элементов в коллекции
REMOVE_COLLECTION_ITEM	Удаление элемента коллекции
INSERT_COLLECTION_ITEM	Вставка элемента коллекции
SET_COLLECTION_ITEM	Замена значения элемента коллекции
GET_COLLECTION_ITEM	Чтение значения элемента коллекции

Функция	Назначение
<b>Функции для работы с ассоциативными массивами</b>	
CREATE_MAP	Определение массива
SET_VALUE	Добавление нового элемента в массив
GET_VALUE	Чтение значения элемента из массива
<b>Функции для обращения к строкам произвольных таблиц QUIK</b>	
GET_ITEM	Строка с указанным номером из таблицы QUIK
GET_NUMBER_OF	Количество записей в указанной таблице QUIK
<b>Функции для обращения к спискам доступных параметров</b>	
GET_CLASSES_LIST	Список кодов классов, доступных в текущем сеансе связи
GET_CLASS_SECURITIES	Список кодов инструментов для заданного списка классов
GET_SECURITY_INFO	Информация по инструменту с определенным кодом из заданного класса
<b>Функции для работы с программируемой таблицей</b>	
ADD_ITEM	Добавление новой строки в таблицу
MODIFY_ITEM	Изменение указанной строки таблицы
DELETE_ITEM	Удаление указанной строки таблицы
DELETE_ALL_ITEMS	Очистка всех значений таблицы
SET_ROW_COLOR	Указание цвета подсветки для строки таблицы
SET_ROW_COLOR_EX	Указание цвета фона и шрифта для строки таблицы
<b>Функции для получения значений Таблицы текущих торгов</b>	
GET_PARAM*	Получение значений из Таблицы текущих торгов
GET_PARAM_EX	Получение всех значений из Таблицы текущих торгов
<b>Функции для получения значений Окна котировок</b>	
GET_QUOTES_II_LEVEL_DATA	Получение значений из Таблицы котировок

Функция	Назначение
<b>Функции для получения значений таблицы «Позиции по инструментам»*</b>	
DEPO_OPEN_BALANCE	Входящий остаток по инструментам
DEPO_OPEN_LIMIT	Входящий лимит по инструментам
DEPO_CURRENT_BALANCE	Текущий остаток по инструментам
DEPO_CURRENT_LIMIT	Текущий лимит по инструментам
DEPO_LIMIT_AVAILABLE	Доступное количество инструментов
DEPO_LIMIT_LOCKED	Заблокированное Количество лотов инструментов
DEPO_LIMIT_LOCKED_BUY	Количество лотов инструментов, заблокированных на покупку
DEPO_LIMIT_LOCKED_BUY_VALUE	Стоимость инструментов, заблокированных на покупку
<b>Функции для получения значений таблицы «Позиции по деньгам»*</b>	
MONEY_OPEN_BALANCE	Входящий остаток по денежным средствам
MONEY_OPEN_LIMIT	Входящий лимит по денежным средствам
MONEY_CURRENT_BALANCE	Текущий остаток по денежным средствам
MONEY_CURRENT_LIMIT	Текущий лимит по денежным средствам
MONEY_LIMIT_AVAILABLE	Доступное количество денежных средств
MONEY_LIMIT_LOCKED	Заблокированное в заявках на покупку количество денежных средств
<b>Функции расчета маржинальных позиций</b>	
SHORT_VALUE	Стоимость всех коротких значений
LONG_VALUE	Стоимость всех длинных позиций
<b>Функции получения значений таблиц «Клиентский портфель» и «Купить/Продать»</b>	
GET_CLIENT_MARGINAL_PORTFOLIO_INFO	Значения параметров таблицы «Клиентский портфель»
GET_CLIENT_MARGINAL_BUY_SELL_INFO	Значение параметров таблицы «Купить/Продать»
<b>Функции для работы с файлами</b>	
CLEAR_FILE	Очистить файл
WRITE	Записать строку в конец файла

Функция	Назначение
WRITELN	Записать строку с переводом строки в конец файла
GET_FILE_LEN	Количество строк в файле
READ_LINE	Прочитать из файла строку с заданным номером
<a href="#">Функции для работы со строками</a>	
LEN	Количество символов в строке
TRIM	Отсечение пробелов в конце строки
SUBSTR	Подстрока
FIND	Поиск в строке позиции подстроки
<a href="#">Функции для работы с графиками</a>	
GET_CANDLE	Значения цен и объемов (OHLCV) «свечи» на графике и ее индикаторов
GET_CANDLE_EX	Значения цен и объемов (OHLCV) «свечи» на графике и ее индикаторов, состоящих из нескольких линий
<a href="#">Функции для работы с заявками</a>	
SEND_TRANSACTION	Ввод новой заявки
<a href="#">Функции для работы с метками</a>	
ADD_LABEL	Добавить метку
DELETE_LABEL	Удалить метку
DELETE_ALL_LABELS	Удалить все метки
GET_LABEL_PARAMS	Получить параметры метки
SET_LABEL_PARAMS	Задать параметры метки
<a href="#">Сервисные функции</a>	
GET_TRADE_DATE	Дата текущей торговой сессии
GET_DATETIME	Текущие дата и время
APPLY_SCALE	Округление с заданной точностью
IS_CONNECTED	Определяет состояние подключения клиентского места к

Функция	Назначение
	серверу
GET_INFO_PARAM	Позволяет получить параметры для информационного окна (Связь/Информационное окно)
BREAKPOINT	Позволяет прервать выполнение программы и вызвать окно «Отладка»

\* – функции предыдущей версии языка QPILE, сохраненные для совместимости. Значения, возвращаемые этими функциями, также могут быть получены чтением строки таблицы функцией «GET\_ITEM» и выборкой из нее значения функцией «GET\_VALUE»

Функции предыдущих версий, поддержка которых прекращена:

- MONEY\_LIMIT\_LOCKED\_NONMARGINAL\_VALUE

## 8.5 Общие функции

### 8.5.1 NEW\_GLOBAL

Функция предназначена для инициализации глобальной переменной. Глобальная переменная сохраняет свое значение между итерациями вычислений значений таблицы. Инициализация может быть выполнена переменной любого типа, при этом создается глобальная переменная соответствующего типа.

NEW\_GLOBAL (STRING Name, ANY InitValue)

Параметры:

№	Параметр	Тип	Описание
1	NAME	STRING	Строковое имя создаваемой переменной
2	INITVALUE	ANY	Значение для инициализации глобальной переменной

В качестве первого параметра может фигурировать строковая константа, либо же переменная, имеющая строковое значение. В последнем случае создается глобальная переменная с тем же именем, что и значение этой строковой переменной.



Пример:

```
\n\nNEW_GLOBAL("GLOBAL","MyFirstGlobal")\nNEW_GLOBAL(Global,1)\n\n
```

После выполнения этих двух строк будут созданы две глобальные переменные. Строковая переменная с именем «GLOBAL» и значением «MyFirstGlobal», а также вещественная переменная с именем «MyFirstGlobal» и значением 1.

### 8.5.2 MESSAGE

Открывает Окно сообщений с заданным текстом.

MESSAGE (STRING Text, DOUBLE Msg\_type)

Параметры:

№	Параметр	Тип	Описание
1	TEXT	STRING	Текст сообщения
2	MSG_TYPE	DOUBLE	Тип сообщения, определяющий вид иконки в Окне и звуковой сигнал. Возможные значения: <ul style="list-style-type: none"><li>– 1 – «Информация»;</li><li>– 2 – «Внимание»;</li><li>– 3 – «Ошибка»</li></ul>

Пример:

```
\n\nMESSAGE ("Hello!",1)\n\n
```

## 8.6 Математические функции

### 8.6.1 ABS

Возвращает модуль числа.

DOUBLE      ABS (DOUBLE Value)

### 8.6.2 ACOS

Возвращает значение арккосинуса аргумента.

DOUBLE      ACOS (DOUBLE Value)

### 8.6.3 ASIN

Возвращает значение арксинуса аргумента.

DOUBLE      ASIN (DOUBLE Value)

### 8.6.4 ATAN

Возвращает значение арктангенса аргумента.

DOUBLE      ATAN (DOUBLE Value)

### 8.6.5 CEIL

Возвращает ближайшее целое число, большее аргумента.

DOUBLE      CEIL (DOUBLE Value)

### 8.6.6 COS

Возвращает значение косинуса аргумента.

DOUBLE      COS (DOUBLE Value)

### 8.6.7 EXP

Вычисление экспонента от аргумента.

DOUBLE      EXP (DOUBLE Value)

### 8.6.8 FLOOR

Возвращает ближайшее целое число, меньшее аргумента.

DOUBLE FLOOR (DOUBLE Value)

### 8.6.9 LOG

Вычисление натурального логарифма от аргумента.

DOUBLE LOG (DOUBLE Value)

### 8.6.10 POW

Возведение аргумента в степень.

DOUBLE POW (DOUBLE Value, DOUBLE Power)

### 8.6.11 RAND

Возвращает случайное целочисленное значение в диапазоне от 0 до 32767.

DOUBLE RAND ()

### 8.6.12 RANDOMIZE

Генерация случайных чисел.

DOUBLE RANDOMIZE ()

Функция инициализирует генератор случайных чисел для определения случайной последовательности генерируемых чисел. Функцию RANDOMIZE () необходимо вызвать до того, как будет использоваться функция RAND().

### 8.6.13 SIN

Возвращает значение синуса аргумента.

DOUBLE SIN (DOUBLE Value)

### 8.6.14 SQRT

Возвращает значение квадратного корня аргумента.

DOUBLE SQRT (DOUBLE Value)

### 8.6.15 TAN

Возвращает значение тангенса аргумента.

DOUBLE      TAN (DOUBLE Value)

Пример:

```
'  
MESSAGE ("ACOS 0.5 - " & acos(0.5),1)  
MESSAGE ("ASIN 0.5 - " & asin(0.5),1)  
MESSAGE ("ATAN 2 - " & atan(2),1)  
MESSAGE ("CEIL 2.2 - " & ceil(2.2),1)  
MESSAGE ("COS 0.5 - " & cos(0.5),1)  
MESSAGE ("EXP 2 - " & exp(2),1)  
MESSAGE ("FLOOR 4.5 - " & floor(4.5),1)  
MESSAGE ("LOG 0.5 - " & log(0.5),1)  
MESSAGE ("POW 2,3 - " & pow(2,3),1)  
MESSAGE ("RAND - " & rand(),1)  
MESSAGE ("SIN 0.5 - " & sin(0.5),1)  
MESSAGE ("SQRT 2 - " & sqrt(2),1)  
MESSAGE ("TAN 0.5 - " & tan(0.5),1)  
'
```

При выполнении примера на экране будет появляться Окно сообщений с отображением функции и результата вычисления ее значения.

## 8.7 Функции для работы с коллекциями объектов (COLLECTION)

Функции из этой группы предназначены для оперирования переменными типа COLLECTION.

**Коллекция** является списком объектов, с индексацией по целочисленному ключу (начиная с 0). Коллекция может содержать элементы любых типов, в том числе и переменные типа COLLECTION. Коллекция может быть неоднородной, т.е. содержать объекты разных типов. В текущей реализации доступ к элементу коллекции по индексу осуществляется за линейное время.

### 8.7.1 CREATE\_COLLECTION

Первая функция создает пустую коллекцию, вторая является конструктором копирования.

COLLECTION    CREATE\_COLLECTION ()

COLLECTION CREATE\_COLLECTION (COLLECTION IntValue)

### 8.7.2 GET\_COLLECTION\_COUNT

Функция возвращает количество элементов в коллекции.

DOUBLE GET\_COLLECTION\_COUNT (COLLECTION Name)

### 8.7.3 REMOVE\_COLLECTION\_ITEM

Функция удаляет элемент с индексом «index» из коллекции «Name».

COLLECTION REMOVE\_COLLECTION\_ITEM (COLLECTION Name, DOUBLE index)

### 8.7.4 INSERT\_COLLECTION\_ITEM

Функция вставляет значение «value» в качестве элемента с индексом «index».

COLLECTION INSERT\_COLLECTION\_ITEM (COLLECTION Name, DOUBLE index, ANY value)

Выполняется вставка в коллекцию элемента с индексом «index», существующий элемент коллекции с индексом «index» сдвигается на единицу в элемент с индексом «index+1», и т.д. для всех существующих элементов, индекс которых равен или больше значения индекса вставки. См. пример к п. [8.7.6](#).

### 8.7.5 SET\_COLLECTION\_ITEM

Функция заменяет значение элемента с индексом «index» на значение «value».

COLLECTION SET\_COLLECTION\_ITEM (COLLECTION Name, DOUBLE index, ANY value)

### 8.7.6 GET\_COLLECTION\_ITEM

Функция возвращает значение элемента с индексом «index».

ANY GET\_COLLECTION\_ITEM (COLLECTION Name, DOUBLE index)

Параметры:

№	Параметр	Тип	Описание
1	INTVALUE	COLLECTION	Коллекция для инициализации вновь создаваемого объекта
2	NAME	STRING	Имя коллекции
3	INDEX	DOUBLE	Индекс для обращения к коллекции. Должен содержать целочисленное

№	Параметр	Тип	Описание
			значение
4	VALUE	ANY	Значение произвольного типа для включения в коллекцию

Пример:

```
\
col=CREATE_COLLECTION()
FOR i FROM 0 TO 9
col=INSERT_COLLECTION_ITEM(col,0,0)
col=SET_COLLECTION_ITEM(col,0,i)
END FOR
s = GET_COLLECTION_ITEM(col,5)
len = GET_COLLECTION_COUNT(col)
\
```

Создается пустая коллекция. Затем 10 раз в коллекцию вставляется нулевой элемент со значением 0 и тут же его значение меняется на значение переменной цикла. В результате получается коллекция из 10-ти элементов. После выполнения программы «s»=4, «len»=10, а «col» имеет следующий вид:

0	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	0

## 8.8 Функции для работы с ассоциативными массивами (MAP)

Функции из этой группы предназначены для оперирования переменными типа MAP.

**Ассоциативный массив (MAP)** является последовательностью пар (ключ, значение), которая обеспечивает получение значения по ключу. Каждому ключу соответствует максимум одно значение. Ассоциативный массив может содержать элементы любых типов, в том числе и переменные структурных типов, разные элементы такого массива могут содержать объекты разных типов в качестве значений. В качестве ключа всегда выступает строковое значение. В текущей реализации доступ к элементу коллекции по ключу осуществляется за линейное время.

### 8.8.1 CREATE\_MAP

Первая функция создает пустой массив, вторая является конструктором копирования.

```
MAP CREATE_MAP ()
```

```
MAP CREATE_MAP (MAP IntValue)
```

### 8.8.2 SET\_VALUE

Функция добавляет в массив «Name» элемент со значением «Value» и ключом «Key».

Если массив уже содержал элемент с ключом «Key», то значение этого элемента заменяется на «Value». Функция возвращает модифицированный массив.

```
MAP SET_VALUE(MAP Name, STRING Key, ANY Value)
```

### 8.8.3 GET\_VALUE

Функция возвращает значение элемента с ключом «Key» из массива «Name». Если элемент не найден, то возвращается пустая строка.

```
STRING GET_VALUE(MAP Name, STRING Key)
```

Параметры:

№	Параметр	Тип	Описание
1	INTVALUE	MAP	Массив для инициализации вновь создаваемого объекта
2	NAME	STRING	Имя массива
3	KEY	STRING	Значение ключа для вставки или доступа к элементу массива
4	VALUE	ANY	Значение произвольного типа для добавления в массив

Пример:

```
\nmap=CREATE_MAP ()\nFOR i FROM 0 TO 9\n  map=SET_VALUE(map, "key" & i, i)\nEND FOR\ns = GET_VALUE(map, "key5")\n\
```

Создается пустой массив. Затем, в цикле, в массив вставляются элементы со значением равным переменной цикла и ключом вида «"key" & i». После выполнения программы «s»=5.

**Функция возвращает значение типа `STRING`, если оно не будет явно преобразовано к типу `DOUBLE`.**

Например:

```
RES=0+GET_VALUE
```

## 8.9 Функции для обращения к строкам произвольных таблиц QUIK

Функции из этой группы предназначены для доступа к данным, содержащимся в таблицах Рабочего места QUIK.

### 8.9.1 GET\_ITEM

Функция возвращает ассоциативный массив (MAP), содержащий информацию о данных из строки с номером «Index» из таблицы с именем «TableName».

```
MAP GET_ITEM(STRING TableName,DOUBLE Index)
```

Возвращаемый массив, в качестве своих элементов, содержит значения ячеек таблицы клиентского места, ключами элементов являются названия колонок. Возможные значения поля «TableName» и ключей приведены ниже в таблицах. Для программируемой таблицы «OWN» значения ключей соответствуют названиям колонок, указанным в секции описания параметров.

### 8.9.2 GET\_NUMBER\_OF

Функция возвращает количество записей в таблице TableName.

```
DOUBLE GET_NUMBER_OF(STRING TableName)
```

Параметры:

№	Параметр	Тип	Описание
1	TABLERNAME	STRING	Имя таблицы QUIK



№	Параметр	Тип	Описание
2	INDEX	DOUBLE	Индекс для обращения к коллекции. Должен содержать целочисленное значение.

Пример:

```
\
n=GET_NUMBER_OF("ALL_TRADES")
value=0
FOR i FROM 1 to n
trade = GET_ITEM ("ALL_TRADES", i)
value = value + GET_VALUE (trade, "VALUE")
END FOR
\
```

В данном примере сначала запрашивается количество записей в Таблице обезличенных сделок, затем в цикле для каждой записи получается соответствующий MAP из которого достаётся значение по ключу «VALUE». В итоге, переменная «value» содержит суммарный объем обезличенных сделок на данный момент.

### 8.9.3 Описания таблиц и параметров

1. Таблицы, используемые в функциях «GET\_NUMBER\_OF» и «GET\_ITEM»:

TableName	Таблица
ORDERS	Заявки
STOP_ORDERS	Стоп заявки
TRADES	Сделки
ALL_TRADES	Обезличенные сделки
MONEY_LIMITS	Позиции по денежным средствам
DEPO_LIMITS	Позиции по инструментам
FUTURES_CLIENT_HOLDINGS	Позиции по клиентским счетам (фьючерсы)
FUTURES_CLIENT_LIMITS	Ограничения по клиентским счетам (фьючерсы)
NEG_DEALS	Таблица заявок на внебиржевые сделки
NEGOTIATION_TRADES	Таблица сделок для исполнения

TableName	Таблица
NEG_DEAL_REPORTS	Таблица заявок-отчетов на сделки РПС
POSITIONS	Позиции участника по деньгам
FIRM_HOLDING	Позиции участника по инструментам
ACCOUNT_BALANCE	Позиции участника по торговым счетам
OWN	Таблица, создаваемая при расчете программы

## 2. Описание параметров **Таблицы заявок**, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Номер заявки в торговой системе	
2	EXCHANGE_CODE	STRING	Код биржи в торговой системе	
3	DATE	DOUBLE	Дата выставления	
4	TIME	DOUBLE	Время выставления	
5	ACTIVATION_TIME	DOUBLE	Время активации	
6	WITHDRAW_TIME	DOUBLE	Время снятия заявки	
7	SECURITY	STRING	Краткое наименование инструмента	
8	SECCODE	STRING	Код инструмента заявки	
9	CLASS	STRING	Краткое название класса заявки	
10	CLASSCODE	STRING	Код класса заявки	
11	OPERATION	STRING	Операция	«SELL» или «BUY»
12	ACCOUNT	STRING	Торговый счет	
13	PRICE	DOUBLE	Цена	
14	QUANTITY	DOUBLE	Количество в лотах	
15	BALANCE	DOUBLE	Остаток	
16	VALUE	DOUBLE	Объем в денежных средствах	

№	Параметр	Тип	Описание	Возможные значения
17	TRADE_CURRENCY	STRING	Валюта заявки	
18	YIELD	DOUBLE	Доходность	
19	ACCRUEDINT	DOUBLE	Накопленный купонный доход	
20	USERID	STRING	Идентификатор трейдера	
21	FIRMID	STRING	Идентификатор фирмы	
22	CLIENTCODE	STRING	Код клиента	
23	COMMENT	STRING	Комментарий	
24	STATUS	STRING	Статус заявки	«ACTIVE» или «KILLED» или «FILLED»
25	TYPE	STRING	Тип заявки	Последовательность трех символов: _ 1-й: «L» – лимитированная, «M» – рыночная; _ 2-й: «S» – разрешить исполнение по разным ценам, «O» – по одной цене; _ 3-й: «N» – немедленно или отклонить, «W» – снять остаток, « » – нет условия
26	TRANS_ID	DOUBLE	Идентификатор транзакции	
27	SETTLECODE	STRING	Код расчетов	
28	PRICE2	DOUBLE	Цена выкупа	
29	IS_MARKET_MAKER_ORDER	STRING	Заявка Маркет-мейкера	«YES» или « »
30	SESSION_DATE	DOUBLE	Дата текущей торговой сессии	
31	TIME_MICROSEC	DOUBLE	Количество микросекунд во времени выставления	
32	WITHDRAW_DATE	DOUBLE	Дата снятия заявки	
33	WITHDRAW_TIME_MICROSEC	DOUBLE	Количество микросекунд во времени снятия	

№	Параметр	Тип	Описание	Возможные значения
34	PERIOD	DOUBLE	Период торговой сессии	_ «0» – открытие, _ «1» – нормальный, _ «2» – закрытие
35	VISIBLE_QUANTITY	DOUBLE	Видимое количество. Параметр заявок типа «Айсберг»	
36	LINKED_ORDER	DOUBLE	Номер заявки в торговой системе	
37	SEC_CURRENCY	STRING	Валюта расчетов	
38	EXPIRE_DATE	DOUBLE	Срок	
39	UID	DOUBLE	UID	

### 3. Описание параметров **Таблицы сделок**, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Номер сделки в торговой системе	
2	EXCHANGE_CODE	STRING	Код биржи в торговой системе	
3	DATE	DOUBLE	Дата совершения	
4	TIME	DOUBLE	Время совершения	
5	ORDER_NUMBER	DOUBLE	Номер заявки в торговой системе	
6	SECURITY	STRING	Краткое название инструмента	
7	SECCODE	STRING	Код инструмента заявки	
8	CLASS	STRING	Краткое название класса	
9	CLASSCODE	STRING	Код класса	
10	OPERATION	STRING	Операция	«SELL» или «BUY»
11	TYPE	STRING	Тип сделки	«MARGIN», если сделка маржинальная
12	ACCOUNT	STRING	Торговый счет	

№	Параметр	Тип	Описание	Возможные значения
13	PRICE	DOUBLE	Цена	
14	QUANTITY	DOUBLE	Количество в лотах	
15	VALUE	DOUBLE	Объем в денежных средствах	
16	TRADE_CURRENCY	STRING	Валюта	
17	SETTLE_CURRENCY	STRING	Валюта расчетов	
18	SETTLE_CODE	STRING	Код расчетов	
19	YIELD	DOUBLE	Доходность	
20	ACCRUEDINT	DOUBLE	Накопленный купонный доход	
21	USERID	STRING	Идентификатор трейдера	
22	STATION_ID	STRING	Идентификатор рабочей станции	
23	FIRMID	STRING	Идентификатор дилера	
24	FIRMNAME	STRING	Идентификатор организации трейдера	
25	CLIENTCODE	STRING	Код клиента	
26	COMMENT	STRING	Комментарий	
27	PARTNER_FIRMID	STRING	Идентификатор фирмы партнера	
28	PARTNER_FIRM_NAME	STRING	Название фирмы партнера	
29	PRICE2	DOUBLE	Цена выкупа	
30	REPORATE	DOUBLE	Ставка РЕПО(%)	
31	TS_COMISSION	DOUBLE	Комиссия торговой системы	
32	CLEARING_COMISSION	DOUBLE	Клиринговая комиссия (Московской Биржи)	
33	EXCHANGE_COMISSION	DOUBLE	Комиссия Фондовой биржи (Московской Биржи)	

№	Параметр	Тип	Описание	Возможные значения
34	TECH_CENTER_COMISSION	DOUBLE	Комиссия Технического центра (Московской Биржи)	
35	ACCRUED2	DOUBLE	Доход(%) на дату выкупа	
36	REPOVALUE	DOUBLE	Сумма РЕПО	
37	REPO2VALUE	DOUBLE	Объем выкупа РЕПО	
38	REPOTERM	DOUBLE	Срок РЕПО	
39	START_DISCOUNT	DOUBLE	Начальный дисконт(%)	
40	LOWER_DISCOUNT	DOUBLE	Нижний дисконт(%)	
41	UPPER_DISCOUNT	DOUBLE	Верхний дисконт(%)	
42	BLOCK_SECURITIES	STRING	Блокировка обеспечения	«Да» или «Нет»
43	SESSION_DATE	DOUBLE	Дата текущей торговой сессии	
44	TIME_MICROSEC	DOUBLE	Количество микросекунд во времени совершения	
45	PERIOD	DOUBLE	Период торговой сессии	_ «0» – открытие; _ «1» – нормальный; _ «2» – закрытие

№	Параметр	Тип	Описание	Возможные значения
46	KIND	DOUBLE	Вид сделки	<ul style="list-style-type: none"> <li>_ «1» – обычная;</li> <li>_ «2» – адресная;</li> <li>_ «3» – первичное размещение;</li> <li>_ «4» – перевод денег / инструментов;</li> <li>_ «5» – адресная сделка первой части РЕПО;</li> <li>_ «6» – расчетная по операции своп;</li> <li>_ «7» – расчетная по внебиржевой операции своп;</li> <li>_ «8» – расчетная сделка бивалютной корзины;</li> <li>_ «9» – расчетная внебиржевая сделка бивалютной корзины;</li> <li>_ «10» – сделка по операции РЕПО с ЦК;</li> <li>_ «11» – первая часть сделки по операции РЕПО с ЦК;</li> <li>_ «12» – вторая часть сделки по операции РЕПО с ЦК;</li> <li>_ «13» – адресная сделка по операции РЕПО с ЦК;</li> <li>_ «14» – первая часть адресной сделки по операции РЕПО с ЦК;</li> <li>_ «15» – вторая часть адресной сделки по операции РЕПО с ЦК;</li> <li>_ «16» – техническая сделка по возврату активов РЕПО с ЦК;</li> </ul>

№	Параметр	Тип	Описание	Возможные значения
				<ul style="list-style-type: none"> <li>– «17» – сделка по спреду между фьючерсами разных сроков на один актив;</li> <li>– «18» – техническая сделка первой части от спреда между фьючерсами;</li> <li>– «19» – техническая сделка второй части от спреда между фьючерсами;</li> <li>– «20» – адресная сделка первой части РЕПО с корзиной;</li> <li>– «21» – адресная сделка второй части РЕПО с корзиной;</li> <li>– «22» – перенос позиций срочного рынка»;</li> <li>– «23» – Late correction – XLON;</li> <li>– «24» – Not to mark – XLON;</li> <li>– «25» – Previous Day Contra;</li> <li>– «26» – Ordinary trade immediate publication – XLON;</li> <li>– «27» – Inter Fund Transfer delayed publication – XOFF;</li> <li>– «28» – Negotiated Trade delayed publication – XLON;</li> <li>– «29» – Negotiated Trade immediate publication – XLON;</li> <li>– «30» – OTC Late Correction – XOFF;</li> <li>– «31» – Ordinary Trade delayed publication – XLON;</li> <li>– «32» – Ordinary Trade Immediate publication – XOFF;</li> <li>– «33» – SI Late Correction;</li> <li>– «34» – SI Trade immediate publication;</li> <li>– «35» – SI Trade delayed publication;</li> <li>– «36» – OTC Trade delayed publication – XOFF;</li> <li>– «37» – OTC MTF TBA 1;</li> <li>– «38» – OTC trade – delayed publication MTF TBA 1;</li> <li>– «39» – Inter fund cross - delayed publication requested MTF TBA 1;</li> </ul>



№	Параметр	Тип	Описание	Возможные значения
				<ul style="list-style-type: none"> <li>– «40» – Cancellation of OTC trade after date of publication MTF TBA 1;</li> <li>– «41» – OTC MTF TBA 2;</li> <li>– «42» – OTC trade – delayed publication MTF TBA 2;</li> <li>– «43» – Inter fund cross - delayed publication requested MTF TBA 2;</li> <li>– «44» – Cancellation of OTC trade after date of publication MTF TBA 2;</li> <li>– «45» – OTC MTF TBA 3;</li> <li>– «46» – OTC trade – delayed publication MTF TBA 3;</li> <li>– «47» – Inter fund cross - delayed publication requested MTF TBA 3;</li> <li>– «48» – Cancellation of OTC trade after date of publication MTF TBA 3;</li> <li>– «49» – OTC MTF TBA 4;</li> <li>– «50» – OTC trade – delayed publication MTF TBA 4;</li> <li>– «51» – Inter fund cross - delayed publication requested MTF TBA 4;</li> <li>– «52» – Cancellation of OTC trade after date of publication MTF TBA 4;</li> <li>– «53» – Delayed Publication Late Correction XLON;</li> <li>– «54» – No to Mark Late Correction XLON;</li> <li>– «55» – Витринная сделка по операции своп;</li> <li>– «58» – Витринная сделка по адресной операции своп;</li> <li>– «59» – Витринная сделка по Беспоставочному свопу;</li> <li>– «60» – Сделка FX SPOT;</li> </ul>

№	Параметр	Тип	Описание	Возможные значения
				<ul style="list-style-type: none"> <li>«61» – Сделка по Беспоставочному валютному форвардному контракту;</li> <li>«62» – Сделка FX DEPOSIT;</li> <li>«63» – Сделка по Валютному расчетному форвардному контракту</li> </ul>

**4. Описание параметров Таблицы обезличенных сделок, возвращаемое функцией «GET\_ITEM»:**

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Номер сделки в торговой системе	
2	DATE	DOUBLE	Дата выставления	
3	TIME	DOUBLE	Время выставления	
4	SECURITY	STRING	Краткое наименование инструмента	
5	SECCODE	STRING	Код инструмента заявки	
6	CLASS	STRING	Краткое название класса	
7	CLASSCODE	STRING	Код класса	
8	PRICE	DOUBLE	Цена	
9	QUANTITY	DOUBLE	Количество в лотах	
10	VALUE	DOUBLE	Объем в денежных средствах	
11	OPERATION	STRING	Направление операции	«SELL» или «BUY»
12	ACCRUEDINT	DOUBLE	Накопленный купонный доход	
13	YIELD	DOUBLE	Доходность	
14	SETTLE_CODE	STRING	Код расчетов	
15	REPORATE	DOUBLE	Ставка РЕПО(%)	
16	REPOVALUE	DOUBLE	Сумма РЕПО	
17	REPO2VALUE	DOUBLE	Объем выкупа РЕПО	

№	Параметр	Тип	Описание	Возможные значения
18	REPOTERM	DOUBLE	Срок РЕПО	
19	SESSION_DATE	DOUBLE	Дата текущей торговой сессии	
20	TIME_MICROSEC	DOUBLE	Количество микросекунд во времени совершения	
21	PERIOD	DOUBLE	Период торговой сессии	_ «0» – открытие; _ «1» – нормальный; _ «2» – закрытие

#### 5. Описание параметров Таблицы стоп-заявок, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Регистрационный номер стоп-заявки на сервере QUIK	
2	DATE	DOUBLE	Дата выставления	
3	TIME	DOUBLE	Время выставления	
4	WITHDRAW_TIME	DOUBLE	Время снятия заявки	
5	STOP_ORDER_TYPE	DOUBLE	Вид стоп заявки	_ «1» – стоп-лимит; _ «2» – условие по другому инструменту; _ «3» – со связанной заявкой; _ «6» – тейк-профит; _ «7» – стоп-лимит по исполнению активной заявки; _ «8» – тейк-профит по исполнению активной заявки; _ «9» – тэйк-профит и стоп-лимит
6	TYPE	STRING	Тип заявки	Последовательность трех символов: _ 1-й: «L» – лимитированная, «M» – рыночная; _ 2-й: «S» – разрешить исполнение по разным ценам, «O» – по одной цене;

№	Параметр	Тип	Описание	Возможные значения
				3-й: «N» – немедленно или отклонить, «W» – снять остаток, « » – нет условия
7	SECURITY	STRING	Краткое название инструмента	
8	SECCODE	STRING	Код инструмента заявки	
9	CLASS	STRING	Краткое название класса заявки	
10	CLASSCODE	STRING	Код класса заявки	
11	OPERATION	STRING	Операция	«SELL» или «BUY»
12	ACCOUNT	STRING	Торговый счет	
13	CONDITION_SECURITY	STRING	Инструмент стоп-цены	
14	CONDITION_SECCODE	STRING	Код инструмента стоп-цены	
15	CONDITION_CLASS	STRING	Класс стоп-цены	
16	CONDITION_CLASSCODE	STRING	Код класса стоп-цены	
17	CONDITION	STRING	Направленность стоп-цены	«LESS_OR_EQUAL_VALUE» или «GREATER_OR_EQUAL_VALUE»
18	CONDITION_PRICE	DOUBLE	Стоп-цена	
19	CONDITION2	STRING	Направленность стоп-лимит цены (для заявок типа «Тэйк-профит и стоп-лимит»)	«LESS_OR_EQUAL_VALUE» или «GREATER_OR_EQUAL_VALUE»
20	CONDITION_PRICE2	DOUBLE	Стоп-лимит цена (для заявок типа «Тэйк-профит и стоп-лимит»)	
21	PRICE	DOUBLE	Цена	
22	MARKET_STOP_LIMIT	STRING	Признак исполнения заявки «стоп-лимит» по рыночной цене (для заявок типа «Тэйк-профит и стоп-лимит»)	«Да» либо пусто
23	QUANTITY	DOUBLE	Количество в лотах	

№	Параметр	Тип	Описание	Возможные значения
24	BALANCE	DOUBLE	Активное количество	
25	FILLED_VOLUME	DOUBLE	Исполненное количество	
26	FIRMID	STRING	Дилер	
27	UID	DOUBLE	UID	
28	CLIENTCODE	STRING	Код клиента	
29	COMMENT	STRING	Комментарий	
30	LINKED_ORDER	DOUBLE	Номер заявки в торговой системе, зарегистрированной по наступлению условия стоп-цены	
31	ALL_TRADE_NUMBER	DOUBLE	Сделка условия	
32	EXPIRE_DATE	DOUBLE	Срок	
33	EXPIRY_DATE_IS_TODAY	STRING	Срок действия заявки равен «Сегодня»	
34	ACTIVE_IN_TIME_INTERVAL	STRING	Признак действия заявки типа «Тэйк-профит и стоп-лимит» в течение периода времени	«Да» либо пусто
35	ACTIVE_FROM_TIME	DOUBLE	Время начала периода действия заявки типа «Тэйк-профит и стоп-лимит»	
36	ACTIVE_TO_TIME	DOUBLE	Время окончания периода действия заявки типа «Тэйк-профит и стоп-лимит»	
37	USE_BASE_ORDER_BALANCE	STRING	Брать исполненный объем в качестве количества выставленной стоп-заявки (для заявок «по исполнению»)	«Да» либо пусто
38	KILL_IF_LINKED_ORDER_PARTLY_FILLED	STRING	При частичном исполнении связанной	«Да» либо пусто

№	Параметр	Тип	Описание	Возможные значения
			заявки снять стоп-заявку (для заявок «по исполнению»)	
39	ACTIVATE_IF_BASE_ORDER_PARTLY_FILLED	STRING	Частичное исполнение учитывается (для заявок «по исполнению»)	«Да» либо пусто
40	TYPE_DESCRIPTION	STRING	Тип	
41	STATUS	STRING	Статус заявки	«ACTIVE», «KILLED» или «FILLED»
42	RESULT_DESCRIPTION	STRING	Результат	<ul style="list-style-type: none"> <li>_ «Отвергнута ТС»;</li> <li>_ «Не прошла контроль лимитов»;</li> <li>_ «Связ. заявка снята»;</li> <li>_ «Связ. заявка исполнена»;</li> <li>_ «Рассчитывается min\max»;</li> <li>_ «Ждет активации»;</li> <li>_ «Рассчитывается min\max и ждет активации»;</li> <li>_ «Снята»;</li> <li>_ «Выставлена заявка в ТС»</li> </ul>
43	CO_ORDER_NUMBER	DOUBLE	Связанная заявка	
44	CO_ORDER_PRICE	DOUBLE	Цена связанной заявки	
45	TRANS_ID	DOUBLE	Идентификатор транзакции	
46	OFFSET	DOUBLE	Отступ от min/max	
47	OFFSET_UNITS	STRING	Единицы отступа	«%» или «Д»
48	SPREAD	DOUBLE	Защитный спрэд	
49	SOE_USE_SPREAD_AS_PERCENTS	STRING	Значение «защитного спрэда» указано в процентах	«Да» либо пусто
50	MARKET_TAKE_PROFIT	STRING	Признак исполнения заявки «тэйк-профит» по рыночной цене (для заявок типа «Тэйк-профит и стоп-лимит»)	«Да» либо пусто
51	BASE_ORDER_NUM	DOUBLE	Заявка условия	
52	OWNER_SERVER	STRING	Сервер	«Другой» или «Текущий»

**6. Описание параметров таблицы «Позиции по деньгам», возвращаемое функцией «GET\_ITEM»:**

№	Параметр	Тип	Описание
1	FIRMID	STRING	Идентификатор фирмы
2	CURRCODE	STRING	Код валюты
3	TAG	STRING	Код позиции
4	CLIENT_CODE	STRING	Код клиента
5	OPEN_BALANCE	DOUBLE	Входящий остаток по деньгам
6	OPEN_LIMIT	DOUBLE	Входящий лимит по деньгам
7	CURRENT_BALANCE	DOUBLE	Текущий остаток по деньгам
8	CURRENT_LIMIT	DOUBLE	Текущий лимит по деньгам
9	LOCKED	DOUBLE	Заблокированное количество
10	AVAILABLE	DOUBLE	Доступное количество
11	LOCKED_VALUE	DOUBLE	Количество денежных средств, заблокированное на покупку немаржинальных инструментов
12	LIMIT_KIND	DOUBLE	Срок расчётов

**7. Описание параметров таблицы «Позиции по инструментам», возвращаемое функцией «GET\_ITEM»:**

№	Параметр	Тип	Описание
1	FIRMID	STRING	Идентификатор фирмы
2	SECCODE	STRING	Код инструмента
3	TRDACCID	STRING	Счет депо
4	CLIENT_CODE	STRING	Код клиента
5	OPEN_BALANCE	DOUBLE	Входящий остаток по инструментам
6	OPEN_LIMIT	DOUBLE	Входящий лимит по инструментам
7	CURRENT_BALANCE	DOUBLE	Текущий остаток по инструментам
8	CURRENT_LIMIT	DOUBLE	Текущий лимит по инструментам

№	Параметр	Тип	Описание
9	LOCKED_SELL	DOUBLE	Заблокировано
10	AVAILABLE	DOUBLE	Доступное количество
11	WA_POSITION_PRICE	DOUBLE	Цена приобретения
12	LIMIT_KIND	DOUBLE	Срок расчётов

**8. Описание параметров Таблицы ограничений по клиентским счетам, возвращаемое функцией «GET\_ITEM»:**

№	Параметр	Тип	Описание	Возможные значения
1	FIRMID	STRING	Идентификатор фирмы	
2	TRDACCID	STRING	Торговый счет	
3	TYPE	STRING	Тип лимита	«Ден.средства» или «Залоговые ден.средства» или «Всего»
4	LIQUIDITY_COEFF	DOUBLE	Коэффициент ликвидности	
5	CBP_PREV_LIMIT	DOUBLE	Предыдущий лимит открытых позиций	
6	CBPLIMIT	DOUBLE	Лимит открытых позиций	
7	CBPLUSED	DOUBLE	Текущие чистые позиции	
8	CBPLUSED_FOR_ORDERS	DOUBLE	Текущие чистые позиции (под заявки)	
9	CBPLUSED_FOR_POSITIONS	DOUBLE	Текущие чистые позиции (под открытые позиции)	
10	CBPLPLANNED	DOUBLE	Планируемые чистые позиции	
11	VARMARGIN	DOUBLE	Вариационная маржа	
12	ACCRUEDINT	DOUBLE	Накопленный доход	
13	OPTIONS_PREMIUM	DOUBLE	Премия по опционам	
14	TS_COMISSION	DOUBLE	Биржевые сборы	



№	Параметр	Тип	Описание	Возможные значения
15	KGO	DOUBLE	Коэффициент клиентского гарантийного обеспечения	

9. Описание параметров Таблицы позиций по клиентским счетам, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	FIRMID	STRING	Идентификатор фирмы	
2	TRDACCID	STRING	Торговый счет	
3	SECCODE	STRING	Код фьючерсного контракта	
4	SEC_SHORT_NAME	STRING	Краткое название контракта	
5	TYPE	STRING	Тип лимита	«Основной счет» или «Клиентские и дополнительные счета» или «Все счета торг. членов» или <пусто>
6	START_BUY	DOUBLE	Входящие длинные позиции	
7	START_SELL	DOUBLE	Входящие короткие позиции	
8	START_NET	DOUBLE	Входящие чистые позиции	
9	TODAY_BUY	DOUBLE	Текущие длинные позиции	
10	TODAY_SELL	DOUBLE	Текущие короткие позиции	
11	TOTAL_NET	DOUBLE	Текущие чистые позиции	
12	OPEN_BUY	DOUBLE	Активные на покупку	
13	OPEN_SELL	DOUBLE	Активные на продажу	
14	CBPLUSED	DOUBLE	Оценка текущих чистых позиций	

№	Параметр	Тип	Описание	Возможные значения
15	CBPLPLANNED	DOUBLE	Плановые чистые позиции	
16	VARMARGIN	DOUBLE	Вариационная маржа	
17	AVRPOSNPRICE	DOUBLE	Эффективная цена позиций	
18	POSITIONVALUE	DOUBLE	Стоимость позиций	

**10.** Описание параметров Таблицы заявок на внебиржевые сделки, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Номер	
2	QUOTENO	DOUBLE	Встречная безадресная заявка	
3	DATE	STRING	Дата выставления заявки	
4	TIME	STRING	Время выставления заявки	
5	ACTIVATION_DATE	DOUBLE	Дата активации заявки	
6	ACTIVATION_TIME	DOUBLE	Время активации заявки	
7	SECURITY	STRING	Краткое название инструмента	
8	SECCODE	STRING	Код инструмента	
9	CLASS	STRING	Класс	
10	CLASSCODE	STRING	Код класса	
11	OPERATION	STRING	Операция	«Купля» либо «Продажа»
12	ACCOUNT	STRING	Счет	
13	PRICE	DOUBLE	Цена	
14	QUANTITY	DOUBLE	Количество	
15	USERID	STRING	Треjder	
16	FIRMID	STRING	Идентификатор дилера	

№	Параметр	Тип	Описание	Возможные значения
17	FIRMNAME	STRING	Идентификатор организации трейдера	
18	UID	DOUBLE	UID	
19	CPUSERID	STRING	Трейдер партнера	
20	CPFIRMID	STRING	Идентификатор партнера	
21	CPFIRMNAME	STRING	Организация партнёр	
22	CLIENTCODE	STRING	Код клиента	
23	COMMENT	STRING	Комментарий	
24	MATCH_REFERENCE	STRING	Ссылка	
25	STATUS	STRING	Состояние	«Активна», «Исполнена» или «Снята»
26	SETTLE_CODE	STRING	Код расчетов	
27	DIRECTION	STRING	Направленность	«Отправлена», «Получена» или «Отправлена и получена»
28	YIELD	DOUBLE	Доходность	
29	VALUE	DOUBLE	Объем	
30	ACCRUEDINT	DOUBLE	Купонный процент	
31	PRICE2	DOUBLE	Цена выкупа	
32	REFUNDRATE	DOUBLE	Ставка возмещения (%)	
33	REPORATE	DOUBLE	Ставка РЕПО (%)	
34	TRANS_ID	DOUBLE	ID транзакции	
35	REPOVALUE	DOUBLE	Сумма РЕПО	
36	REPO2VALUE	DOUBLE	Объем выкупа РЕПО	
37	REPOENTRY	STRING	Тип ввода заявки РЕПО	«Цена1+Ставка», «Ставка+Цена2», «Цена1+Цена2», «Сумма РЕПО + Количество», «Сумма РЕПО + Дисконт», «Количество + Дисконт», «Сумма РЕПО», «Количество»
38	REPOTERM	DOUBLE	Срок РЕПО	
39	START_DISCOUNT	DOUBLE	Начальный дисконт(%)	

№	Параметр	Тип	Описание	Возможные значения
40	LOWER_DISCOUNT	DOUBLE	Нижний дисконт(%)	
41	UPPER_DISCOUNT	DOUBLE	Верхний дисконт(%)	
42	BLOCK_SECURITIES	STRING	Блокировка обеспечения	«Да» или «Нет»
43	ORIG_REPOVALUE	DOUBLE	Сумма РЕПО первоначальная	
44	ORIG_VOLUME	DOUBLE	Количество первоначальное	
45	ORIG_DISCOUNT	DOUBLE	Процент дисконта первоначальный	
46	WITHDRAW_TIME	DOUBLE	Время снятия заявки	
47	BALANCE	DOUBLE	Остаток	
48	SETTLE_CURRENCY	STRING	Валюта расчетов	

**11.** Описание параметров **Таблицы сделок для исполнения**, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Номер	
2	ORDER_NUMBER	DOUBLE	Номер заявки	
3	DATE	STRING	Дата торгов	
4	SETTLEDATE	STRING	Дата расчетов	
5	CLASS	STRING	Класс	
6	CLASSCODE	STRING	Код класса	
7	SECCODE	STRING	Код инструмента	
8	SECURITY	STRING	Краткое наименование инструмента	
9	OPERATION	STRING	Операция	«Купля» либо «Продажа»
10	CLIENTCODE	STRING	Код клиента	
11	COMMENT	STRING	Комментарий	
12	FIRMID	STRING	Идентификатор дилера	

№	Параметр	Тип	Описание	Возможные значения
13	FIRMNAME	STRING	Идентификатор организации трейдера	
14	ACCOUNT	STRING	Счет депо	
15	CPFIRMNAME	STRING	Партнер	
16	CPFIRMID	STRING	Идентификатор партнера	
17	CPACCOUNT	STRING	Счет депо партнера	
18	PRICE	DOUBLE	Цена	
19	QUANTITY	DOUBLE	Количество	
20	VALUE	DOUBLE	Объем	
21	STATUS	STRING	Состояние	<ul style="list-style-type: none"> <li>_ «FILLED» – исполнена;</li> <li>_ «NOT FILLED» – не исполнена;</li> <li>_ «INCLUDE IN REPORT» – включена в отчет</li> </ul>
22	ACCRUEDINT	DOUBLE	Купонный %	
23	PRICE1	DOUBLE	Цена 1ой части РЕПО	
24	PRICE2	DOUBLE	Цена выкупа	
25	REPORTTRADENO	DOUBLE	Номер сделки 1ой части РЕПО	
26	REPORATE	DOUBLE	Ставка РЕПО(%)	
27	SETTLE_CODE	STRING	Код расчетов	
28	REPORT_NUM	DOUBLE	Отчет	
29	CPREPORT_NUM	DOUBLE	Отчет партнера	
30	TS_COMISSION	DOUBLE	Комиссия торговой системы	
31	BALANCE	DOUBLE	Остаток	
32	SETTLETIME	STRING	Время исполнения	
33	AMOUNT	DOUBLE	Сумма обязательства	
34	REPOVALUE	DOUBLE	Сумма РЕПО	

№	Параметр	Тип	Описание	Возможные значения
35	REPOTERM	DOUBLE	Срок РЕПО	
36	REPO2VALUE	DOUBLE	Объем выкупа РЕПО	
37	RETURN_VALUE	DOUBLE	Сумма возврата РЕПО	
38	DISCOUNT	DOUBLE	Дисконт(%)	
39	LOWER_DISCOUNT	DOUBLE	Нижний дисконт(%)	
40	UPPER_DISCOUNT	DOUBLE	Верхний дисконт(%)	
41	BLOCK_SECURITIES	STRING	Блокировать обеспечение	«Да» либо «Нет»
42	URGENCY_FLAG	STRING	Исполнить	«Да» либо «Нет»
43	TRADE_TYPE	STRING	Тип	_ «Внесистемная сделка»; _ «Первая часть сделки РЕПО»; _ «Вторая часть сделки РЕПО»; _ «Компенсационный взнос»
44	TRADE_OPERATION_TYPE	STRING	Направленность	«Зачислить» либо «Списать»
45	EXPECTED_DISCOUNT	DOUBLE	Дисконт после взноса(%)	
46	EXPECTED_QUANTITY	DOUBLE	Количество после взноса	
47	EXPECTED_REPOVALUE	DOUBLE	Сумма РЕПО после взноса	
48	EXPECTED_REPO2VALUE	DOUBLE	Стоимость выкупа после взноса	
49	EXPECTED_RETURN_VALUE	DOUBLE	Сумма возврата после взноса	
50	REPORT_TRADE_DATE	DOUBLE	Дата заключения	
51	STATE_OF_CLEARING	STRING	Состояние расчетов	_ «Processed»; _ «Not processed»; _ «Is processing»
52	TYPE_OF_CLEARING	STRING	Тип клиринга	_ «Not set»; _ «Simple»; _ «Multilateral»
53	REPORT_COMISSION	DOUBLE	Комиссия за отчет	

№	Параметр	Тип	Описание	Возможные значения
54	COUPON_PAYMENT	DOUBLE	Купонная выплата	
55	COUPON_PAYMENT_DATE	DOUBLE	Дата купонной выплаты	
56	PRINCIPAL_PAYMENT	DOUBLE	Выплата по основному долгу	
57	PRINCIPAL_PAYMENT_DATE	DOUBLE	Дата выплаты по основному долгу	
58	SETTLE_CURRENCY	STRING	Валюта расчетов	

**12.** Описание параметров Таблицы заявок-отчетов на сделки РПС, возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	NUMBER	DOUBLE	Номер	
2	DATE	STRING	Дата	
3	TIME	STRING	Время выставления заявки	
4	CLASS	STRING	Класс	
5	SECCODE	STRING	Код инструмента	
6	SECURITY	STRING	Краткое наименование инструмента	
7	USERID	STRING	Идентификатор трейдера	
8	FIRMID	STRING	Идентификатор дилера	
9	FIRMNAME	STRING	Идентификатор организации трейдера	
10	ACCOUNT	STRING	Счет депо	
11	CPFIRMNAME	STRING	Партнер	
12	CPFIRMID	STRING	Идентификатор партнера	
13	CPACCOUNT	STRING	Счет депо партнера	
14	QUANTITY	DOUBLE	Количество	
15	VALUE	DOUBLE	Объем	

№	Параметр	Тип	Описание	Возможные значения
16	COMISSION	DOUBLE	Комиссия	
17	DIRECTION	STRING	Направленность	«Отправлен» или «Получен»
18	STATUS	STRING	Состояние	«Ждет исполнения», «Снят» либо «Исполнен»
19	REPORT_TYPE	STRING	Тип отчета	«EXECUTION» или «CANCEL EXECUTION»
20	REPORT_KIND	STRING	Вид отчета	

**13.** Описание параметров таблицы «Позиции участника по деньгам», возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	FIRMID	STRING	Фирма	
2	CURRCODE	STRING	Валюта	
3	TAG	STRING	Код позиции	
4	DESCRIPTION	STRING	Описание	
5	OPENBAL	DOUBLE	Входящий	
6	CURRENTPOS	DOUBLE	Текущий	
7	PLANNEDPOS	DOUBLE	Плановый	
8	LIMIT1	DOUBLE	Внешнее ограничение	
9	ORDERBUY	DOUBLE	На покупку (в заявках)	
10	ORDERSELL	DOUBLE	На продажу (в заявках)	
11	NETOBLIGATION	DOUBLE	Нетто	
12	PLANNEDBAL	DOUBLE	Контрольная	
13	BANK_ACC_ID	STRING	Идентификатор счета	

**14.** Описание параметров таблицы «Позиции участника по инструментам», возвращаемое функцией «GET\_ITEM»:



№	Параметр	Тип	Описание	Возможные значения
1	FIRMID	STRING	Фирма	
2	SEC_SHORT_NAME	STRING	Название инструмента	
3	SECCODE	STRING	Код инструмента	
4	OPENBAL	DOUBLE	Входящий	
5	CURRENTPOS	DOUBLE	Текущий	
6	PLANNEDPOSBUY	DOUBLE	Плановая покупка	
7	PLANNEDPOSSELL	DOUBLE	Плановая продажа	
8	USQTYB	DOUBLE	Куплено	
9	USQTYS	DOUBLE	Продано	

**15.** Описание параметров таблицы «Позиции участника по торговым счетам», возвращаемое функцией «GET\_ITEM»:

№	Параметр	Тип	Описание	Возможные значения
1	SECCODE	STRING	Код инструмента	
2	SEC_SHORT_NAME	STRING	Название инструмента	
3	FIRMID	STRING	Идентификатор фирмы	
4	TRDACCID	STRING	Торговый счет	
5	DEPACCID	STRING	Счет депо	
6	OPENBAL	DOUBLE	Входящий	
7	CURRENTPOS	DOUBLE	Текущий	
8	PLANNEDPOSBUY	DOUBLE	Плановая покупка	
9	PLANNEDPOSSELL	DOUBLE	Плановая продажа	
10	PLANBAL	DOUBLE	Контрольный	
11	USQTYB	DOUBLE	Куплено	
12	USQTYS	DOUBLE	Продано	
13	PLANNED	DOUBLE	Плановый	

## 8.10 Функции для обращения к спискам доступных параметров

### 8.10.1 GET\_CLASSES\_LIST

Функция предназначена для получения списка кодов классов, полученных с сервера в ходе текущего сеанса связи. Разделителем в списке кодов классов выступает запятая «,».

GET\_CLASSES\_LIST ()

Пример:

```
\
ClassesList = GET_CLASSES_LIST ()
\
```

Переменной «ClassesList» присваивается список доступных классов, например «TQBR, TQBS, TQNL, TQOB, TQOS, TQNO».

### 8.10.2 GET\_CLASS\_SECURITIES

Функция предназначена для получения списка кодов инструментов для списка классов, заданного списком кодов. Разделителем в списке кодов инструментов выступает запятая «,».

GET\_CLASS\_SECURITIES (STRING)

Пример:

```
\
SecuritiesList = GET_CLASS_SECURITIES ("TQBR,GKO")
\
```

Переменной «SecuritiesList» присваивается список кодов всех доступных инструментов для классов «A1-Акции» и «ГКО».

### 8.10.3 GET\_SECURITY\_INFO

Функция возвращает информацию по инструменту с кодом «sec\_code» из класса с кодом «class\_code». Если «class\_code» равен «», то ищет по всем классам до первого нахождения.

MAP GET\_SECURITY\_INFO (STRING class\_code, STRING sec\_code)

Параметры:

№	Параметр	Тип	Описание
1	CODE	STRING	Код инструмента
2	NAME	STRING	Наименование инструмента
3	SHORT_NAME	STRING	Краткое наименование
4	CLASS_CODE	STRING	Код класса
5	CLASS_NAME	STRING	Наименование класса
6	FACE_VALUE	DOUBLE	Номинал
7	FACE_UNIT	STRING	Код валюты номинала
8	SCALE	DOUBLE	Количество значащих цифр после запятой
9	MAT_DATE	STRING	Дата погашения
10	LOT_SIZE	DOUBLE	Размер лота

Пример:

```
\n\nSecInfo = GET_SECURITY_INFO ("","YUKO")\nLot = GET_VALUE (SecInfo, "LOT_SIZE")\n\n
```

Переменной «SecInfo» присваиваются значения параметров инструмента «YUKO». Переменная «Lot» содержит количество инструментов в одном лоте «YUKO».

## 8.11 Функции для работы с программируемой таблицей

Следующий набор функций оперирует только с таблицей «OWN», создаваемой при расчете программы. Данная таблица доступна не только для чтения с помощью стандартных функции «GET\_ITEM» и «GET\_NUMBER\_OF», но также и для модификации с помощью следующих функций.

### 8.11.1 ADD\_ITEM

Данная функция вставляет строку с индексом «Index» в таблицу «OWN». Ассоциативный массив «table\_string» используется для инициализации колонок таблицы значениями. «Table\_string» должен содержать элементы с ключами равными названиям колонок.

ADD\_ITEM (DOUBLE Index, MAP table\_string)

### 8.11.2 MODIFY\_ITEM

Функция модифицирует уже существующую строку с индексом «Index» с помощью массива «table\_string».

MODIFY\_ITEM (DOUBLE Index, MAP table\_string)

### 8.11.3 DELETE\_ITEM

Функция предназначена для удаления строки с индексом «Index».

DELETE\_ITEM (DOUBLE Index)

### 8.11.4 DELETE\_ALL\_ITEMS

Функция полностью очищает таблицу «OWN».

DELETE\_ALL\_ITEMS()

Параметры:

№	Параметр	Тип	Описание
1	INDEX	DOUBLE	Индекс для обращения к коллекции. Должен содержать целочисленное значение
2	TABLE_STRING	MAP	Массив, содержащий значения для колонок модифицируемой строки

Пример:

```
\n
st=CREATE_MAP()
st=SET_VALUE(st, "Value",10)
ADD_ITEM(1,st)
st=GET_ITEM ("OWN",1)
value=GET_VALUE(st,"Value")
DELETE_ALL_ITEMS()
```

```
`
```

Данный пример может работать только с таблицей, содержащей единственную колонку с именем «Value». Сначала создается массив, содержащий элемент со значением «10» и ключом «Value». Затем, в таблице создается строка с индексом «1» и значением колонки «Value» равным «10». В дальнейшем строка с индексом «1» считывается обратно в массив и значение элемента с ключом «Value» попадает в переменную «value». И, наконец, все строки таблицы удаляются вызовом «DELETE\_ALL\_ITEMS()».

### 8.11.5 SET\_ROW\_COLOR

Функция назначения цвета фона указанной строки таблицы. Данная функция относится к предыдущей версии языка и сохранена для совместимости.

```
SET_ROW_COLOR (STRING client_code, STRING background_color,  
STRING selected_background_color)
```

Параметры:

№	Параметр	Тип	Описание
1	CLIENT_CODE	STRING	Код клиента, при установке этого параметра в «ROWNAME» цвет подсветки устанавливается на текущего рассчитываемого клиента
2	BACKGROUND_COLOR	STRING	Основной цвет подсветки строки
3	SELECTED_BACKGROUND_COLOR	STRING	Цвет подсветки строки, когда строка выделена

Цвет фона (подсветки) задается макросом «RGB(<red>, <green>, <blue>)». Например, если в качестве параметра цвета передадите в функцию строку «RGB(255, 0, 0)», то цвет подсветки установится в красный. Для использования системного цвета по умолчанию используйте строку «DEFAULT\_COLOR».

Пример:

```
`  
  
SET_ROW_COLOR (ROWNAME, "RGB(0,255,0)", "DEFAULT_COLOR")  
  
`
```

Основной цвет строки таблицы для текущего рассчитываемого клиента здесь устанавливается в зеленый, цвет выделенной строки – по умолчанию.

#### 8.11.6 SET\_ROW\_COLOR\_EX

Функция назначения цвета фона и шрифта указанной строки таблицы.

SET\_ROW\_COLOR\_EX (DOUBLE row\_number, STRING background\_color, STRING selected\_background\_color, STRING font\_color, STRING selected\_font\_color)

Параметры:

№	Параметр	Тип	Описание
1	ROW_NUMBER	DOUBLE	Номер строки для подсветки
2	BACKGROUND_COLOR	STRING	Основной цвет фона строки
3	SELECTED_BACKGROUND_COLOR	STRING	Цвет фона строки, когда строка выделена курсором
4	FONT_COLOR	STRING	Основной цвет шрифта строки
5	SELECTED_FONT_COLOR	STRING	Цвет шрифта, когда строка выделена курсором

Цвет фона (подсветки) задается макросом «RGB(<red>, <green>, <blue>)». Например, если в качестве параметра цвета передадите в функцию строку «RGB(255, 0, 0)», то цвет подсветки установится в красный. Для использования системного цвета по умолчанию используйте строку «DEFAULT\_COLOR».

Пример:

```
\n\nSET_ROW_COLOR_EX (10, "DEFAULT_COLOR", "DEFAULT_COLOR", "RGB(0,255,0)",\n"RGB(0,0,255)")\n\n
```

Основной цвет шрифта для 10-й строки таблицы устанавливается в зеленый. При выделении её курсором цвет шрифта станет синим.

## 8.12 Функции для получения значений Таблицы текущих торгов

### 8.12.1 GET\_PARAM

Функция предназначена для получения значения параметров биржевой информации. С помощью этой функции можно получить данные из Таблицы текущих торгов для заданных кодов класса и инструмента.

GET\_PARAM (STRING classcode\_list, STRING seccode, STRING param\_name)

Параметры:

№	Параметр	Тип	Описание
1	CLASSCODE_LIST	STRING	Список кодов классов инструментов, в которых будет искаться инструмент через запятую. Например, если в функцию передается строка «TQBR,TQBS,TQNL», то инструмент будет искаться в классах «А1-Акции», «А2-Акции» и «Б-Акции»
2	SECCODE	STRING	Код инструмента
3	PARAM_NAME	STRING	Идентификатор параметра

Пример:

```
\nLastPrice = GET_PARAM ("TQBR", "HYDR", "last")\n
```

Переменной «LastPrice» присваивается значение цены последней сделки для «РусГидро» из класса «А1-Акции Московской Биржи».

### 8.12.2 GET\_PARAM\_EX

Функция предназначена для получения значений всех параметров биржевой информации из Таблицы текущих торгов. С помощью этой функции можно получить любое из значений Таблицы текущих торгов для заданных кодов класса и инструмента.

MAP GET\_PARAM\_EX (STRING classcode, STRING seccode, STRING param\_name)

Если в коде программы используется команда USE\_CASE\_SENSITIVE\_CONSTANTS (см. п. [8.3.2](#)), то значения параметров «classcode» и «seccode» должны быть указаны в регистре, которому

они соответствуют в Таблице текущих торгов, а значение параметра «param\_name» должно быть указано в верхнем регистре.

Параметры:

№	Параметр	Тип	Описание
1	CLASSCODE	STRING	Код класса
2	SECCODE	STRING	Код инструмента
3	PARAM_NAME	STRING	Идентификатор параметра

«MAP» имеет структуру:

№	Параметр	Тип	Описание
1	RESULT	DOUBLE	Результат выполнения операции. Возможные значения: _ «0» – ошибка; _ «1» – параметр найден
2	PARAM_TYPE	DOUBLE	Тип данных параметра, используемый в Таблице текущих торгов. Возможные значения: _ «1» – DOUBLE; _ «2» – LONG; _ «3» – CHAR; _ «4» – перечислимый тип; _ «5» – время; _ «6» – дата
3	PARAM_VALUE	DOUBLE	Значение параметра. Для строк (param_type = 3) значение равно «0». Для перечислимых типов значение равно порядковому значению перечисления
4	PARAM_IMAGE	STRING	Строковое значение параметра, аналогичное его представлению в таблице. В строковом представлении учитываются разделители разрядов, разделители целой и дробной части. Для перечислимых типов выводятся соответствующие им строковые значения

Пример:

```
\n\nPARMAP=GET_PARAM_EX("TQBR","RTKM","WAPRICE")\nWAPRICE=GET_VALUE(PARMAP,"PARAM_VALUE")\n\n
```



Переменной «WAPRICE» присваивается значение средневзвешенной цены по обыкновенным акциям «Ростелекома» из класса «А1-Акции Московской Биржи».

### 8.12.3 Значения параметров функций

Список возможных кодов классов инструментов «classcode\_list»:

Код класса	Название	Код класса	Название
TQBR	МБ ФР: Т+ А1-Акции, паи и РДР	GAZP	Облигации Газпром
TQBS	МБ ФР: Т+ А2-Акции и паи	INDX	МБ ФР: Индексы
TQNL	МБ ФР: Т+ Б-Акции и паи	GTS	PTC: СГК
TQOB	МБ ФР: Т+ А1-Облигации	GAZ	PTC: Акции Газпрома
TQOS	МБ ФР: Т+ А2-Облигации	QUADRO	PTC-СГК (Торги в валюте)
TQNO	МБ ФР: Т+ Б-Облигации	RTS10	PTC: Рынок «Десяти»
PSEQ	МБ ФР: РПС: А1-Акции и паи	RTSIDX	Индексы PTC
PSES	МБ ФР: РПС: А2-Акции и паи	RTSIND	PTC (Индексы)
PSNL	МБ ФР: РПС: Б-Акции и паи	SES2	ГЦБ: Крупные лоты
PSOB	МБ ФР: РПС: А1-Облигации	SPBFUT	Фьючерсы FORTS
PSNO	МБ ФР: РПС: Б-Облигации	SPBOPT	Опционы FORTS
PSAU	МБ ФР: РПС: Первичное размещение (облигации)	SPBSPT	Спот Санкт-Петербург
AUCT	МБ ФР: Аукцион (Акции)	SPBGKO	МКО
MAIN	МБ ГЦБ	SPBCEX	ГГКО на Петербургской валютной бирже
MAIC	ГЦБ: период закрытия	SPCGKO	Тестовая система на СПББ
BOBR	МБ ГЦБ: Облигации ЦБ	EQBREMU	Акции 1-го уровня (Эмулятор)
FUOP	Фьючерсы ММВБ	USDRUB	Курс доллара на FORTS

Список возможных идентификаторов параметров:

№	Параметр	Тип	Описание
1	STATUS	STRING	Статус
2	LOTSIZE	NUMERIC	Размер лота

№	Параметр	Тип	Описание
3	BID	NUMERIC	Лучшая цена спроса
4	BIDDEPTH	NUMERIC	Спрос по лучшей цене
5	BIDDEPTHT	NUMERIC	Суммарный спрос
6	NUMBIDS	NUMERIC	Количество заявок на покупку
7	OFFER	NUMERIC	Лучшая цена предложения
8	OFFERDEPTH	NUMERIC	Предложение по лучшей цене
9	OFFERDEPTHT	NUMERIC	Суммарное предложение
10	NUMOFFERS	NUMERIC	Количество заявок на продажу
11	OPEN	NUMERIC	Цена открытия
12	HIGH	NUMERIC	Максимальная цена сделки
13	LOW	NUMERIC	Минимальная цена сделки
14	LAST	NUMERIC	Цена последней сделки
15	CHANGE	NUMERIC	Разница цены последней к предыдущей сессии
16	QTY	NUMERIC	Количество инструментов в последней сделке
17	TIME	STRING	Время последней сделки
18	VOLTODAY	NUMERIC	Количество инструментов в обезличенных сделках
19	VALTODAY	NUMERIC	Оборот в деньгах
20	TRADINGSTATUS	STRING	Состояние сессии
21	VALUE	NUMERIC	Оборот в деньгах последней сделки
22	WAPRICE	NUMERIC	Средневзвешенная цена
23	HIGHBID	NUMERIC	Лучшая цена спроса сегодня
24	LOWOFFER	NUMERIC	Лучшая цена предложения сегодня
25	NUMTRADES	NUMERIC	Количество сделок за сегодня
26	PREVPRICE	NUMERIC	Цена закрытия
27	PREVWAPRICE	NUMERIC	Предыдущая оценка
28	CLOSEPRICE	NUMERIC	Цена периода закрытия
29	LASTCHANGE	NUMERIC	% изменения от закрытия
30	PRIMARYDIST	STRING	Размещение

№	Параметр	Тип	Описание
31	ACCRUEDINT	NUMERIC	Накопленный купонный доход
32	YIELD	NUMERIC	Доходность последней сделки
33	COUPONVALUE	NUMERIC	Размер купона
34	YIELDATPREVWAPRI	NUMERIC	Доходность по предыдущей оценке
35	YIELDATWAPRICE	NUMERIC	Доходность по оценке
36	PRICEMINUSPREVWAPRICE	NUMERIC	Разница цены последней к предыдущей оценке
37	CLOSEYIELD	NUMERIC	Доходность закрытия
38	CURRENTVALUE	NUMERIC	Текущее значение индексов Московской Биржи
39	LASTVALUE	NUMERIC	Значение индексов Московской Биржи на закрытие предыдущего дня
40	LASTTOPREVSTLPRC	NUMERIC	Разница цены последней к предыдущей сессии
41	PREVSETTLEPRICE	NUMERIC	Предыдущая расчетная цена
42	PRICEMVTLIMIT	NUMERIC	Лимит изменения цены
43	PRICEMVTLIMITT1	NUMERIC	Лимит изменения цены T1
44	MAXOUTVOLUME	NUMERIC	Лимит объема активных заявок (в контрактах)
45	PRICEMAX	NUMERIC	Максимально возможная цена
46	PRICEMIN	NUMERIC	Минимально возможная цена
47	NEGVALTODAY	NUMERIC	Оборот внесистемных в деньгах
48	NEGNUMTRADES	NUMERIC	Количество внесистемных сделок за сегодня
49	NUMCONTRACTS	NUMERIC	Количество открытых позиций
50	CLOSETIME	STRING	Время закрытия предыдущих торгов (для индексов РТС)
51	OPENVAL	NUMERIC	Значение индекса РТС на момент открытия торгов
52	CHNGOPEN	NUMERIC	Изменение текущего индекса РТС по сравнению со значением открытия
53	CHNGCLOSE	NUMERIC	Изменение текущего индекса РТС по сравнению со значением закрытия
54	BUYDEPO	NUMERIC	Гарантийное обеспечение продавца
55	SELLDEPO	NUMERIC	Гарантийное обеспечение покупателя
56	CHANGETIME	STRING	Время последнего изменения

№	Параметр	Тип	Описание
57	SELLPROFIT	NUMERIC	Доходность продажи
58	BUYPROFIT	NUMERIC	Доходность покупки
59	TRADECHANGE	NUMERIC	Разница цены последней к предыдущей сделки (FORTS, ФБ СПб, СПбБ)
60	FACEVALUE	NUMERIC	Номинал (для инструментов СПбБ)
61	MARKETPRICE	NUMERIC	Рыночная цена вчера
62	MARKETPRICETODAY	NUMERIC	Рыночная цена
63	NEXTCOUPON	NUMERIC	Дата выплаты купона
64	BUYBACKPRICE	NUMERIC	Цена оферты
65	BUYBACKDATE	NUMERIC	Дата оферты
66	ISSUESIZE	NUMERIC	Объем обращения
67	PREVDATE	NUMERIC	Дата предыдущего торгового дня
68	DURATION	NUMERIC	Дюрация
69	LOPENPRICE	NUMERIC	Официальная цена открытия
70	LCURRENTPRICE	NUMERIC	Официальная текущая цена
71	LCLOSEPRICE	NUMERIC	Официальная цена закрытия
72	QUOTEBASIS	STRING	Тип цены
73	PREVADMITTEDQUOT	NUMERIC	Признаваемая котировка предыдущего дня
74	LASTBID	NUMERIC	Лучшая цена спроса на момент завершения периода торгов
75	LASTOFFER	NUMERIC	Лучшая цена предложения на момент завершения торгов
76	PREVLEGALCLOSEPR	NUMERIC	Цена закрытия предыдущего дня
77	COUPONPERIOD	NUMERIC	Длительность купона
78	MARKETPRICE2	NUMERIC	Рыночная цена 2
79	ADMITTEDQUOTE	NUMERIC	Признаваемая котировка
80	BGOP	NUMERIC	БГО по покрытым позициям
81	BGONP	NUMERIC	БГО по непокрытым позициям
82	STRIKE	NUMERIC	Цена страйк

№	Параметр	Тип	Описание
83	STEPPRICET	NUMERIC	Стоимость шага цены
84	STEPPRICE	NUMERIC	Стоимость шага цены (для новых контрактов FORTS)
85	SETTLEPRICE	NUMERIC	Расчетная цена
86	OPTIONTYPE	STRING	Тип опциона
87	OPTIONBASE	STRING	Базовый актив
88	VOLATILITY	NUMERIC	Волатильность опциона
89	THEORPRICE	NUMERIC	Теоретическая цена
90	PERCENTRATE	NUMERIC	Агрегированная ставка
91	ISPERCENT	STRING	Тип цены фьючерса
92	CLSTATE	STRING	Статус клиринга
93	CLPRICE	NUMERIC	Котировка последнего клиринга
94	STARTTIME	STRING	Начало основной сессии
95	ENDTIME	STRING	Окончание основной сессии
96	EVNSTARTTIME	STRING	Начало вечерней сессии
97	EVNENDTIME	STRING	Окончание вечерней сессии
98	MONSTARTTIME	STRING	Начало утренней сессии
99	MONENDTIME	STRING	Окончание утренней сессии
100	CURSTEPPRICE	STRING	Валюта шага цены
101	REALVMPRICE	NUMERIC	Текущая рыночная котировка
102	MARG	STRING	Маржируемый
103	EXPDATE	NUMERIC	Дата исполнения инструмента
104	CROSSRATE	NUMERIC	Курс
105	BASEPRICE	NUMERIC	Базовый курс
106	HIGHVAL	NUMERIC	Максимальное значение (RTSIND)
107	LOWVAL	NUMERIC	Минимальное значение (RTSIND)
108	ICHANGE	NUMERIC	Изменение (RTSIND)
109	IOPEN	NUMERIC	Значение на момент открытия (RTSIND)
110	PCHANGE	NUMERIC	Процент изменения (RTSIND)

№	Параметр	Тип	Описание
111	OPENPERIODPRICE	NUMERIC	Цена предторгового периода
112	MIN_CURR_LAST	NUMERIC	Минимальная текущая цена
113	SETTLECODE	STRING	Код расчетов по умолчанию
114	STEPPRICECL	DOUBLE	Стоимость шага цены для клиринга
115	STEPPRICEPRCL	DOUBLE	Стоимость шага цены для промклиринга
116	MIN_CURR_LAST_TI	STRING	Время изменения минимальной текущей цены
117	PREVLLOTSIZE	NUMERIC	Предыдущее значение размера лота
118	LOTSIZECHANGEDAT	NUMERIC	Дата последнего изменения размера лота
119	AUCTPRICE	NUMERIC	Цена послеторгового аукциона
120	CLOSING_AUCTION_VOLUME	NUMERIC	Количество в сделках послеторгового аукциона

Список идентификаторов дополнительных параметров, доступных для функции GET\_PARAM\_EX:

№	Параметр	Тип	Описание
1	LONGNAME	STRING	Полное название инструмента
2	SHORTNAME	STRING	Краткое название инструмента
3	CODE	STRING	Код инструмента
4	CLASSNAME	STRING	Название класса
5	CLASS_CODE	STRING	Код класса
6	TRADE_DATE_CODE	DOUBLE	Дата торгов
7	MAT_DATE	DOUBLE	Дата погашения
8	DAYS_TO_MAT_DATE	DOUBLE	Число дней до погашения
9	SEC_FACE_VALUE	DOUBLE	Номинал инструмента
10	SEC_FACE_UNIT	STRING	Валюта номинала
11	SEC_SCALE	DOUBLE	Точность цены
12	SEC_PRICE_STEP	DOUBLE	Минимальный шаг цены
13	SECTYPE	STRING	Тип инструмента

## 8.13 Функции для получения значений Окна котировок

### 8.13.1 GET\_QUOTES\_II\_LEVEL\_DATA

Функция предназначена для получения значений котировок инструментов. С помощью этой функции можно получить данные из Таблицы котировок для заданных кодов класса и инструмента.

MAP GET\_QUOTES\_II\_LEVEL\_DATA (STRING ClassCode, STRING SecCode)

Возвращаемый «MAP» имеет следующую структуру:

№	Параметр	Тип	Описание
1	BID_COUNT	DOUBLE	Количество котировок покупки
2	OFFER_COUNT	DOUBLE	Количество котировок продажи
3	BID	COLLECTION	Котировки спроса (покупки)
4	OFFER	COLLECTION	Котировки предложений (продажи)

Коллекции «BID» и «OFFER» имеют следующую структуру:

№	Параметр	Тип	Описание
1	PRICE	DOUBLE	Цена покупки / продажи
2	QUANTITY	DOUBLE	Количество в лотах

## 8.14 Функции для получения значений таблицы «Позиции по инструментам»

Функции предназначены для получения значений таблицы для заданного кода клиента, кода фирмы, кода инструмента и счета депо.

### 8.14.1 DEPO\_OPEN\_BALANCE

Возвращает значение «Входящего остатка по инструментам».

DEPO\_OPEN\_BALANCE (STRING client\_code, STRING firmid, STRING seccode, STRING account)

#### **8.14.2 DEPO\_OPEN\_LIMIT**

Возвращает значение «Входящего лимита по инструментам».

DEPO\_OPEN\_LIMIT (STRING client\_code, STRING firmid, STRING seccode, STRING account)

#### **8.14.3 DEPO\_CURRENT\_BALANCE**

Возвращает значение «Текущего остатка по инструментам».

DEPO\_CURRENT\_BALANCE (STRING client\_code, STRING firmid, STRING seccode,  
STRING account)

#### **8.14.4 DEPO\_CURRENT\_LIMIT**

Возвращает значение «Текущего лимита по инструментам».

DEPO\_CURRENT\_LIMIT (STRING client\_code, STRING firmid, STRING seccode, STRING account)

#### **8.14.5 DEPO\_LIMIT\_AVAILABLE**

Возвращает значение доступного количества инструментов.

DEPO\_LIMIT\_AVAILABLE (STRING client\_code, STRING firmid, STRING seccode, STRING  
account)

#### **8.14.6 DEPO\_LIMIT\_LOCKED**

Возвращает значение «Заблокированного количества инструментов».

DEPO\_LIMIT\_LOCKED (STRING client\_code, STRING firmid, STRING seccode, STRING account)

#### **8.14.7 DEPO\_LIMIT\_LOCKED\_BUY**

Возвращает значение «Заблокированного на покупку количества лотов».

DEPO\_LIMIT\_LOCKED\_BUY (STRING client\_code, STRING firmid, STRING seccode, STRING  
account)

#### **8.14.8 DEPO\_LIMIT\_LOCKED\_BUY\_VALUE**

Возвращает значение «Стоимости инструментов, заблокированных на покупку».



DEPO\_LIMIT\_LOCKED\_BUY\_VALUE (STRING client\_code, STRING firmid, STRING seccode, STRING account)

Параметры:

№	Параметр	Тип	Описание
1	CLIENT_CODE	STRING	Код клиента
2	FIRMID	STRING	Код фирмы
3	SECCODE	STRING	Код инструмента
4	ACCOUNT	STRING	Счет депо

**Параметр «account» должен указываться с соблюдением регистра символов (верхнего/нижнего).**

Пример:

```
\nClDepoOB = DEPO_OPEN_BALANCE ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoOL = DEPO_OPEN_LIMIT ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoCB = DEPO_CURRENT_BALANCE ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoCL = DEPO_CURRENT_LIMIT ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoAV = DEPO_LIMIT_AVAILABLE ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoLCK = DEPO_LIMIT_LOCKED ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoLCKBuy = DEPO_LIMIT_LOCKED_BUY ("1075", "NC0080000000", "HYDR", "L01-00000F00")\nClDepoLCKBuyValue = DEPO_LIMIT_LOCKED_BUY_VALUE ("1075", "NC0080000000", "HYDR",\n"L01-00000F00")\n\
```

В примере показано присвоение переменным значений таблицы «Позиции по инструментам» для клиента с кодом «1075» по обыкновенным акциям «РусГидро»:

- переменной «ClDepoOB» присваивается значение входящего остатка;
- переменной «ClDepoOL» присваивается значение входящего лимита;
- переменной «ClDepoCB» присваивается значение текущего остатка;
- переменной «ClDepoCL» присваивается значение текущего лимита;
- переменной «ClDepoAV» присваивается значение доступного количества инструментов;
- переменной «ClDepoLCK» присваивается значение заблокированного количества инструментов;

- переменной «ClDepoLCKBuy» присваивается значение заблокированного на покупку количества лотов;
- переменной «ClDepoLCKBuyValue» присваивается значение стоимости инструментов, заблокированных на покупку.

## 8.15 Функции для получения значений таблицы «Позиции по деньгам»

Функции предназначены для получения значений таблицы для заданного кода клиента, кода фирмы, кода позиции и кода валюты.

### 8.15.1 MONEY\_OPEN\_BALANCE

Возвращает значение «Входящего остатка по денежным средствам».

MONEY\_OPEN\_BALANCE (STRING client\_code, STRING firmid, STRING tag, STRING curr\_code)

### 8.15.2 MONEY\_OPEN\_LIMIT

Возвращает значение «Входящего лимита по денежным средствам».

MONEY\_OPEN\_LIMIT (STRING client\_code, STRING firmid, STRING tag, STRING curr\_code)

### 8.15.3 MONEY\_CURRENT\_BALANCE

Возвращает значение «Текущего остатка по денежным средствам».

MONEY\_CURRENT\_BALANCE (STRING client\_code, STRING firmid, STRING tag, STRING curr\_code)

### 8.15.4 MONEY\_CURRENT\_LIMIT

Возвращает значение «Текущего лимита по денежным средствам».

MONEY\_CURRENT\_LIMIT (STRING client\_code, STRING firmid, STRING tag, STRING curr\_code)

### 8.15.5 MONEY\_LIMIT\_AVAILABLE

Возвращает значение «Доступного количества денежных средств».

MONEY\_LIMIT\_AVAILABLE (STRING client\_code, STRING firmid, STRING tag, STRING curr\_code)

### 8.15.6 MONEY\_LIMIT\_LOCKED

Возвращает значение «Заблокированного количества денежных средств».

MONEY\_LIMIT\_LOCKED (STRING client\_code, STRING firmid, STRING tag, STRING curr\_code)

Пример:

```
\nClMoneyOB = MONEY_OPEN_BALANCE ("1075", "NC0080000000", "EQTV", "SUR")\nClMoneyOL = MONEY_OPEN_LIMIT ("1075", "NC0080000000", "EQTV", "SUR")\nClMoneyCB = MONEY_CURRENT_BALANCE ("1075", "NC0080000000", "EQTV", "SUR")\nClMoneyCL = MONEY_CURRENT_LIMIT ("1075", "NC0080000000", "EQTV", "SUR")\nClMoneyAV = MONEY_LIMIT_AVAILABLE ("1075", "NC0080000000", "EQTV", "SUR")\nClMoneyLCK = MONEY_LIMIT_LOCKED ("1075", "NC0080000000", "EQTV", "SUR")\n\
```

В примере показано присвоение переменным значений таблицы «Позиции по деньгам» на Фондовой Московской Бирже для клиента с кодом «1075»:

- переменной «ClMoneyOB» присваивается значение входящего остатка по денежным средствам;
- переменной «ClMoneyOL» присваивается значение входящего лимита по денежным средствам;
- переменной «ClMoneyCB» присваивается значение текущего остатка по денежным средствам;
- переменной «ClMoneyCL» присваивается значение текущего лимита по денежным средствам;
- переменной «ClMoneyAV» присваивается значение доступного количества денежных средств;
- переменной «ClMoneyLCK» присваивается значение заблокированного количества денежных средств.

## 8.16 Функции расчета маржинальных позиций

Функции предназначены для получения значений маржинальных позиций для заданного кода клиента, кода фирмы, кода класса, счета депо и кода параметра цены, по которой идет расчет стоимости (например «OPEN» – цена открытия, «LAST» – цена последней сделки и т.п.).

### 8.16.1 SHORT\_VALUE

Возвращает значение «Стоимости всех коротких позиций».

SHORT\_VALUE (STRING client\_code, STRING firmid, STRING seccode, STRING class\_code, STRING account, STRING price\_param\_code)

### 8.16.2 LONG\_VALUE

Возвращает значение «Стоимости всех длинных позиций».

LONG\_VALUE (STRING client\_code, STRING firmid, STRING seccode, STRING class\_code, STRING account, STRING price\_param\_code)

Параметры:

№	Параметр	Тип	Описание
1	CLIENT_CODE	STRING	Код клиента
2	FIRMID	STRING	Код фирмы
3	SECCODE	STRING	Код инструмента
4	CLASS_CODE	STRING	Код класса
5	ACCOUNT*	STRING	Счет депо
6	PRICE_PARAM_CODE	STRING	Код параметра цены

**(\*) параметр чувствителен к верхнему/нижнему регистру символов.**

Пример:

```
\nClShortsValue = SHORT_VALUE ("1075", "NC0080000000", "HYDR", "TQBR", "L01-00000F00",\n"LAST")\nClLongsValue = LONG_VALUE ("1075", "NC0080000000", "HYDRR", "TQBR", "L01-00000F00",\n"OPEN")\n\
```

Переменной «ClShortsValue» присваивается значение стоимости всех коротких позиций для клиента с кодом «1075» по инструменту «РусГидро» из класса «A1-Акции» по счету «L01-00000F00», исходя из цены последней сделки.

Переменной `CLLongsValue` присваивается значение стоимости всех длинных позиций для клиента с кодом «1075» по инструменту «РусГидро» из класса «A1-Акции» по счету «L01-00000F00», исходя из цены открытия.

## 8.17 Функции получения значений таблиц «Клиентский портфель» и «Купить/Продать»

Функции предназначены для получения значений указанных таблиц. Значения этих таблиц вычисляются на рабочем месте клиента QUIK с периодичностью, установленной в настройках (пункт меню **Система/Настройки/Основные настройки...**, раздел «Торговля» / «Клиентский портфель», флажок «Обновлять через каждые .. секунд»).

### 8.17.1 GET\_CLIENT\_MARGINAL\_PORTFOLIO\_INFO

Функция возвращает ассоциативный массив (MAP) с параметрами таблицы «Клиентский портфель», соответствующих идентификатору участника торгов «`firmid`» и коду клиента «`client_code`».

MAP GET\_CLIENT\_MARGINAL\_PORTFOLIO\_INFO (STRING `firmid`, STRING `client_code`)

Параметры:

№	Параметр	Тип	Описание
1	IS_LEVERAGE	STRING (12)	Признак использования схемы кредитования с контролем текущей стоимости активов. Возможные значения: _ «МЛ» – используется схема ведения позиции «по плечу», «плечо» рассчитано по значению Входящего лимита; _ «МП» – используется схема ведения позиции «по плечу», «плечо» указано явным образом; _ «МОП» – используется схема ведения позиции «лимит на открытую позицию»; _ «МД» – используется схема ведения позиции «по дисконтам»; _ <пусто> – используется схема ведения позиции «по лимитам»
2	IN_ASSETS	DOUBLE	Оценка собственных средств клиента до начала торгов

№	Параметр	Тип	Описание	
3	LEVERAGE	DOUBLE	Плечо. Если не было задано явно, то отношение Входящего лимита к Входящим активам	Плечо
4	OPEN_LIMIT	DOUBLE	Оценка максимальной величины заемных средств до начала торгов	Вход. лимит
5	VAL_SHORT	DOUBLE	Оценка стоимости коротких позиций. Значение всегда отрицательное	Шорты
6	VAL_LONG	DOUBLE	Оценка стоимости длинных позиций	Лонги
7	VAL_LONG_MARGIN	DOUBLE	Оценка стоимости длинных позиций по маржинальным инструментам, принимаемым в обеспечение	Лонги МО
8	VAL_LONG_ASSET	DOUBLE	Оценка стоимости длинных позиций по немаржинальным инструментам, принимаемым в обеспечение	Лонги О
9	ASSETS	DOUBLE	Оценка собственных средств клиента по текущим позициям и ценам	Стоимость портфеля
10	CUR_LEVERAGE	DOUBLE	Текущее плечо	Тек.Плечо
11	MARGIN	DOUBLE	Уровень маржи, в процентах	Ур. Маржи
12	LIM_ALL	DOUBLE	Текущая оценка максимальной величины заемных средств	Тек. Лимит
13	AV_LIM_ALL	DOUBLE	Оценка величины заемных средств, доступных для дальнейшего открытия позиций	ДостТекЛимит
14	LOCKED_BUY	DOUBLE	Оценка стоимости активов в заявках на покупку	Блок. покупка
15	LOCKED_BUY_MARGIN	DOUBLE	Оценка стоимости активов в заявках на покупку маржинальных инструментов, принимаемых в обеспечение	Блок. пок. маржин.
16	LOCKED_BUY_ASSET	DOUBLE	Оценка стоимости активов в заявках на покупку немаржинальных инструментов, принимаемых в обеспечение	Блок.пок. обесп.
17	LOCKED_SELL	DOUBLE	Оценка стоимости активов в заявках на продажу маржинальных инструментов	Блок. продажа
18	LOCKED_VALUE_COEF	DOUBLE	Оценка стоимости активов в заявках на покупку немаржинальных инструментов	Блок. пок. немарж.

№	Параметр	Тип	Описание	
19	IN_ALL_ASSETS	DOUBLE	Оценка стоимости всех позиций клиента в ценах закрытия предыдущей торговой сессии, включая позиции по немаржинальным инструментам	ВходСредства
20	ALL_ASSETS	DOUBLE	Текущая оценка стоимости всех позиций клиента	ТекСредства
21	PROFIT_LOSS	DOUBLE	Абсолютная величина изменения стоимости всех позиций клиента	Прибыль/убытки
22	RATE_CHANGE	DOUBLE	Относительная величина изменения стоимости всех позиций клиента	ПроцИзмен
23	LIM_BUY	DOUBLE	Оценка денежных средств, доступных для покупки маржинальных инструментов	На покупку
24	LIM_SELL	DOUBLE	Оценка стоимости маржинальных инструментов, доступных для продажи	На продажу
25	LIM_NON_MARGIN	DOUBLE	Оценка денежных средств, доступных для покупки немаржинальных инструментов	НаПокупНеМаржин
26	LIM_BUY_ASSET	DOUBLE	Оценка денежных средств, доступных для покупки инструментов, принимаемых в обеспечение	НаПокупОбесп
27	VAL_SHORT_NET	DOUBLE	Оценка стоимости коротких позиций. При расчете не используется коэффициент дисконтирования*	Шорты (нетто)
28	VAL_LONG_NET	DOUBLE	Оценка стоимости длинных позиций. При расчете не используется коэффициент дисконтирования*	Лонги (нетто)
29	TOTAL_MONEY_BAL	DOUBLE	Сумма остатков по денежным средствам по всем позициям, без учета средств, заблокированных под исполнение обязательств, выраженная в выбранной валюте расчета	Сумма ден. остатков
30	TOTAL_LOCKED_MONEY	DOUBLE	Сумма заблокированных средств со всех денежных позиций клиента, пересчитанная в валюту расчетов через кросс-курсы на сервере	Суммарно заблок.

№	Параметр	Тип	Описание	
31	HAIRCUTS	DOUBLE	Сумма дисконтов стоимости длинных (только по инструментам обеспечения) и коротких позиций по инструментам, дисконтов корреляции между инструментами, а также дисконтов на задолженности по валютам, не покрытые обеспечением по инструментам в этих же валютах	Сумма дисконтов
32	ASSETS_WITHOUT_HC	DOUBLE	Суммарная величина денежных остатков, стоимости длинных позиций по инструментам обеспечения и стоимости коротких позиций, без учета дисконтирующих коэффициентов, без учета неттинга стоимости инструментов в рамках объединенной позиции по инструментам и без учета корреляции между инструментами	ТекАктБезДиск
33	STATUS_COEF	DOUBLE	Отношение суммы дисконтов к текущим активам без учета дисконтов	Статус счета
34	VARMARGIN	DOUBLE	Текущая вариационная маржа по позициям клиента, по всем инструментам	Вариационная маржа
35	GO_FOR_POSITIONS	DOUBLE	Размер денежных средств, уплаченных под все открытые позиции на срочном рынке	ГО поз.
36	GO_FOR_ORDERS	DOUBLE	Оценка стоимости активов в заявках на срочном рынке	ГО заяв.
37	RATE_FUTURES	DOUBLE	Отношение ликвидационной стоимости портфеля к ГО по срочному рынку	Активы/ГО
38	IS_QUAL_CLIENT	STRING	Признак «квалифицированного» клиента, которому разрешено кредитование заемными средствами с плечом 1:3. Возможные значения: «ПовышУрРиска» – квалифицированный, <пусто> – нет	ПовышУрРиска
39	IS_FUTURES	STRING	Счет клиента на FORTS, в случае наличия объединенной позиции, иначе поле остается пустым	Сроч. счет
40	CURR_TAG	STRING	Актуальные текущие параметры расчета для данной строки в формате «<Валюта>-<Идентификатор торговой сессии>». Пример: «SUR-EQTV»	Парам. расч.

**(\*) Подробнее о коэффициентах дисконтирования см. п. 7 Руководства по администрированию «Настройки Библиотеки расчета лимитов».**



Пример:

```
\nGET_CLIENT_MARGINAL_PORTFOLIO_INFO ("NC0080000000", "1")\n
```

### 8.17.2 GET\_CLIENT\_MARGINAL\_PORTFOLIO\_INFO\_EX

Функция возвращает ассоциативный массив (MAP) с параметрами таблицы «Клиентский портфель», соответствующих идентификатору участника торгов «firmid», коду клиента «client\_code» и сроку расчетов «limit\_kind».

MAP GET\_CLIENT\_MARGINAL\_PORTFOLIO\_INFO\_EX (STRING firmid, STRING client\_code, DOUBLE limit\_kind)

Возможные значения параметра «limit\_kind»:

- «0» – срок расчетов «T0»;
- «1» – срок расчетов «T1»;
- «2» – срок расчетов «T2»;
- «365» – срок расчетов «Тх».

Описание возвращаемых параметров см. в п. [8.17.1](#).

Дополнительно возвращаются следующие параметры:

№	Параметр	Тип	Описание	
1	INIT_MARGIN	DOUBLE	Значение начальной маржи. Заполняется для клиентов типа «МД»	Нач.маржа
2	MIN_MARGIN	DOUBLE	Значение минимальной маржи. Заполняется для клиентов типа «МД»	Мин.маржа
3	CORRECTED_MARGIN	DOUBLE	Значение скорректированной маржи. Заполняется для клиентов типа «МД»	Скор.маржа
4	CLIENT_TYPE	DOUBLE	Тип клиента	Тип клиента

№	Параметр	Тип	Описание
5	PORTFOLIO_VALUE	DOUBLE	Стоимость портфеля. Для клиентов типа «МД» возвращается значение для строк с максимальным сроком расчётов limit_kind

Пример:

```
\
GET_CLIENT_MARGINAL_PORTFOLIO_INFO_EX ("NC0080000000", "1", "0")
\
```

### 8.17.3 GET\_CLIENT\_MARGINAL\_BUY\_SELL\_INFO

Функция возвращает ассоциативный массив (MAP) с параметрами таблицы «Купить/Продать», означающими возможность купить либо продать указанный инструмент «sec\_code» класса «class\_code», указанным клиентом «client\_code» фирмы «firmid», по указанной цене «price». Если цена равна «0», то используются лучшие значения спроса/предложения.

```
MAP GET_CLIENT_MARGINAL_BUY_SELL_INFO (STRING firmid, STRING client_code, STRING
class_code, STRING sec_code, DOUBLE price)
```

Параметры:

№	Параметр	Тип	Описание
1	IS_MARGIN_SEC	DOUBLE	Признак маржинальности инструмента. Возможные значения: «1» – маржинальная, «0» – не маржинальная. Не заполняется для клиентов типа «МД»
2	IS_ASSET_SEC	DOUBLE	Принадлежность инструмента к списку инструментов, принимаемых в обеспечение. Возможные значения: «1» – принимается в обеспечение, «0» – не принимается в обеспечение. Не заполняется для клиентов типа «МД»
3	BALANCE	DOUBLE	Текущая позиция по инструменту, в лотах
4	CAN_BUY	DOUBLE	Оценка количества лотов, доступных на покупку по указанной цене *
5	CAN_SELL	DOUBLE	Оценка количества лотов, доступных на продажу по указанной цене *

№	Параметр	Тип	Описание
6	POSITION_VALUATION	DOUBLE	Денежная оценка позиции по инструменту по ценам спроса/предложения
7	VALUE	DOUBLE	Оценка стоимости позиции по цене последней сделки
8	OPEN_VALUE	DOUBLE	Оценка стоимости позиции клиента, рассчитанная по цене закрытия предыдущей торговой сессии
9	LIM_LONG	DOUBLE	Предельный размер позиции по данному инструменту, принимаемый в обеспечение длинных позиций
10	LONG_COEF	DOUBLE	Коэффициент дисконтирования, применяемый для длинных позиций по данному инструменту
11	LIM_SHORT	DOUBLE	Предельный размер короткой позиции по данному инструменту
12	SHORT_COEF	DOUBLE	Коэффициент дисконтирования, применяемый для коротких позиций по данному инструменту
13	VALUE_COEF	DOUBLE	Оценка стоимости позиции по цене последней сделки, с учетом дисконтирующих коэффициентов
14	OPEN_VALUE_COEF	DOUBLE	Оценка стоимости позиции клиента, рассчитанная по цене закрытия предыдущей торговой сессии с учетом дисконтирующих коэффициентов
15	SHARE	DOUBLE	Процентное отношение стоимости позиции по данному инструменту к стоимости всех активов клиента, рассчитанное по текущим ценам
16	SHORT_WA_PRICE	DOUBLE	Средневзвешенная стоимость коротких позиций по инструментам
17	LONG_WA_PRICE	DOUBLE	Средневзвешенная стоимость длинных позиций по инструментам
18	PROFIT_LOSS	DOUBLE	Разница между средневзвешенной ценой приобретения инструментов и их рыночной оценки
19	SPREAD_HC	DOUBLE	Коэффициент корреляции между инструментами
20	CAN_BUY_OWN	DOUBLE	Максимально возможное количество инструментов в заявке на покупку этого инструмента на этом классе на собственные средства клиента, исходя из лучшей цены предложения
21	CAN_SELL_OWN	DOUBLE	Максимально возможное количество инструментов в заявке на продажу этого инструмента на этом классе из собственных активов клиента, исходя из лучшей цены спроса

№	Параметр	Тип	Описание
22	IS_REST_SHORT_SEC	DOUBLE	Признак того, является ли инструмент разрешенным для продажи на заемные средства. Возможные значения: «1» – разрешена, «0» – не разрешена. Не заполняется для клиентов типа «МД»

**(\*) В зависимости от настроек сервера QUIK, величина может выражаться в лотах, либо в штуках. Уточните единицу измерения у обслуживающего брокера.**

Пример:

```
\
GET_CLIENT_MARGINAL_BUY_SELL_INFO ("NC0080000000", "1", "TQBR", "HYDR", 0)
\
```

#### 8.17.4 GET\_CLIENT\_MARGINAL\_BUY\_SELL\_INFO\_EX

Функция возвращает ассоциативный массив (MAP) с параметрами таблицы «Купить/Продать», означающими возможность купить либо продать указанный инструмент «sec\_code» класса «class\_code», указанным клиентом «client\_code» фирмы «firmid», по указанной цене «price». Если цена равна «0», то используются лучшие значения спроса/предложения.

```
MAP GET_CLIENT_MARGINAL_BUY_SELL_INFO_EX (STRING firmid, STRING client_code,
STRING class_code, STRING sec_code, DOUBLE price)
```

Описание возвращаемых параметров см. в п. [8.17.3](#).

Дополнительно возвращаются следующие параметры:

№	Параметр	Тип	Описание
1	LIMIT_KIND	DOUBLE	Срок расчётов. Возможные значения: – «0» – «T0»; – «1» – «T1»; – «2» – «T2»
2	D_LONG	DOUBLE	Эффективный начальный дисконт для длинной позиции. Заполняется для клиентов типа «МД»
3	D_MIN_LONG	DOUBLE	Эффективный минимальный дисконт для длинной позиции. Заполняется для клиентов типа «МД»

№	Параметр	Тип	Описание
4	D_SHORT	DOUBLE	Эффективный начальный дисконт для короткой позиции. Заполняется для клиентов типа «МД»
5	D_MIN_SHORT	DOUBLE	Эффективный минимальный дисконт для короткой позиции. Заполняется для клиентов типа «МД»
6	CLIENT_TYPE	DOUBLE	Тип клиента
7	IS_LONG_ALLOWED	DOUBLE	Признак того, является ли инструмент разрешенным для покупки на заемные средства. Возможные значения: «1» – разрешена, «0» – не разрешена. Заполняется для клиентов типа «МД»
8	IS_SHORT_ALLOWED	DOUBLE	Признак того, является ли инструмент разрешенным для продажи на заемные средства. Возможные значения: «1» – разрешена, «0» – не разрешена. Заполняется для клиентов типа «МД»

Пример:

```
\
GET_CLIENT_MARGINAL_BUY_SELL_INFO_EX ("NC0080000000", "1", "TQBR", "HYDR", 0)
\
```

## 8.18 Функции для работы с файлами

Функции предназначены для работы с текстовыми файлами и могут использоваться, например, для ведения журнала (log-файла) действий программы. Названия файлов могут содержать пути к ним, например «C:/QUIK/log/new.log».

### 8.18.1 CLEAR\_FILE

Очищает указанный файл.

```
MAP CLEAR_FILE (STRING target_file)
```

Функция возвращает ассоциативный массив (MAP), содержащий параметры:

№	Параметр	Тип	Описание
1	RESULT	DOUBLE	Результат выполнения операции. Возможные значения: «1» – выполнено успешно, «0» – произошла ошибка

№	Параметр	Тип	Описание
2	DESCRIPTION	STRING	Диагностика операционной системы в случае ошибки

### 8.18.2 WRITE

Записывает в конец файла «target\_file» строку «string\_to\_write».

MAP WRITE (STRING target\_file, STRING string\_to\_write)

Функция возвращает ассоциативный массив (MAP), содержащий параметры:

№	Параметр	Тип	Описание
1	RESULT	DOUBLE	Результат выполнения операции. Возможные значения: «1» – выполнено успешно, «0» – произошла ошибка
2	DESCRIPTION	STRING	Диагностика операционной системы в случае ошибки

### 8.18.3 WRITELN

Записывает в конец файла «target\_file» строку «string\_to\_write» с последующим переводом строки.

MAP WRITELN (STRING target\_file, STRING string\_to\_write)

Функция возвращает ассоциативный массив (MAP), содержащий параметры:

№	Параметр	Тип	Описание
1	RESULT	DOUBLE	Результат выполнения операции. Возможные значения: «1» – выполнено успешно, «0» – произошла ошибка
2	DESCRIPTION	STRING	Диагностика операционной системы в случае ошибки

Пример:

```
\
CLEAR_FILE ("new.log")
WRITE ("new.log","Hello, ")
WRITELN ("new.log","world")
```

`

#### 8.18.4 GET\_FILE\_LEN

Возвращает количество строк в файле «target\_file». Если такого файла не существует, возвращает «-1».

DOUBLE      GET\_FILE\_LEN (STRING target\_file)

#### 8.18.5 READ\_LINE

Функция считывает из файла «target\_file» и возвращает строку под номером «line». Длина строки не должна превышать 1000 символов, строки большей длины при считывании будут обрезаны.

STRING      READ\_LINE (STRING target\_file, DOUBLE line, DOUBLE error)

«error» – возвращаемый результат выполнения операции. Если значение равно «0», то чтение прошло успешно; «1» – произошла ошибка, «2» – достигнут конец файла.

Пример:

```
`  
WRITELN ("new.log","Hello, world")  
msg = READ_LINE ("new.log", GET_FILE_LEN("new.log"), error)  
MESSAGE (msg,1)  
`
```

## 8.19 Функции для работы со строками

#### 8.19.1 LEN

Возвращает длину строковой переменной «value».

DOUBLE      LEN (STRING value)

#### 8.19.2 TRIM

Обрезает пробелы в начале и в конце строки.

STRING      TRIM (STRING value)

Функция возвращает строку без пустых символов в начале и в конце строки.

### 8.19.3 SUBSTR

Возвращает подстроку из строки «value», начиная с символа с номером «start» и длиной «len».

STRING SUBSTR (STRING value, DOUBLE start, DOUBLE len)

### 8.19.4 FIND

Ищет вхождение подстроки в заданную строку.

DOUBLE FIND (STRING value, DOUBLE start, STRING sub)

Функция возвращает позицию первого вхождения подстроки «sub» в строке «value», начиная с позиции «start». Если подстрока не найдена, функция возвращает результат «-1».

Пример:

```
\nstroke="anymessage"\nstroke2=SUBSTR(stroke, FIND(stroke, 1, "message"), LEN("message"))\nMESSAGE (stroke2,2)\n\
```

## 8.20 Функции для работы с графиками

### 8.20.1 GET\_CANDLE

Функция для обращения к данным «свечек» на графике, а также к значениям индикаторов технического анализа.

MAP GET\_CANDLE (STRING class\_code, STRING sec\_code, STRING parameter\_name, STRING interval, STRING graph\_type, DOUBLE Date, DOUBLE Time)

Функция возвращает ассоциативный массив (MAP), содержащий информацию о ценах в момент времени «Date»«Time», для графика, построенного по инструменту с кодом «sec\_code» из класса с кодом «class\_code» с временным интервалом «interval». Если «class\_code» равен «», то функция ищет «sec\_code» по всем классам до первого нахождения.

Тип графика указывается следующим кодом:



Тип графика	Код	Тип графика	Код	Тип графика	Код
PRICE	1	PARABOLIC SAR	10	CHAIKIN OSCILLATOR	22
VOLUME	2	SROC	11	CUM WAD	23
MOVING AVERAGE	3	MOM	12	ELDER FI	24
PRICE OSCILLATOR	4	ROC	13	ELDER RAY	25
MACD	5	MFI	16	VERTICAL HORIZONTAL	26
STANDARD DEV	6	WILLIAMS %R	17	FILTER	
BOLLINGER LINES	7	ENVELOPS	18	CHAIKIN VOLATILITY	27
STOCHASTIC	8	VOLUME OSCILLATOR	19		
RSI	9	BALANCE VOLUME	20		
		CUM AD	21		

Значение временного интервала «interval» можно задавать следующими числовыми величинами:

Величина интервала	Значение	Величина интервала	Значение	Величина интервала	Значение
Месячный	month	1 минута	1	15 минут	15
	-3	2 минуты	2	20 минут	20
Недельный	week	3 минуты	3	30 минут	30
	-2	4 минуты	4	60 минут	60
Дневной	day	5 минут	5	2 часа	120
	-1	6 минут	6	4 часа	240
Тиковый	0	10 минут	10		

Значение «parameter\_name» должно соответствовать одному из значений имени параметра из Таблицы текущих торгов, перечень см. в п. [8.12](#). Если «parameter\_name» указан как «», то поиск осуществляется по данным Таблицы обезличенных сделок.

Значение даты «Date» нужно указывать в формате «ГГГГММДД». Например, «20050527» означает 27 мая 2005 г. Значение времени «Time» нужно указывать в формате «ЧЧММСС». Например, «163500» означает 16 ч. 35 мин.

Функция возвращает ассоциативный массив (MAP), содержащий параметры:

№	Параметр	Тип	Описание
1	OPEN	DOUBLE	Цена открытия (первой сделки) в данном интервале времени
2	CLOSE	DOUBLE	Цена закрытия (последней сделки) в интервале
3	HIGH	DOUBLE	Наибольшая цена сделки в интервале
4	LOW	DOUBLE	Наименьшая цена сделки в интервале
5	VOLUME	DOUBLE	Суммарный объем сделок в интервале

Пример:

```
\nmsg = GET_CANDLE("TQBR", "HYDR", "", "5", "PRICE", 20101130, 103500)\nMESSAGE(msg, 2)\n\
```

### 8.20.2 GET\_CANDLE\_EX

В некоторых случаях, например, когда индикатор технического анализа состоит из нескольких линий, применение функции GET\_CANDLE невозможно. Для доступа к таким индикаторам средствами языка QPILE применяется функция GET\_CANDLE\_EX.

MAP GET\_CANDLE\_EX (STRING Tag, DOUBLE Date, DOUBLE Time)

Функция возвращает ассоциативный массив (MAP) с данными для графика со строковым идентификатором Tag в момент времени «Date» и «Time». Таким образом, для того чтобы обратиться к данным на графике, необходимо его предварительно построить и присвоить ему уникальный строковый идентификатор Tag. Идентификатор Tag задается в диалоге «Редактирование настроек графика» на вкладке «Дополнительно» (см. п. 4.2.3 Раздела 4 «Работа с графиками»).

При вызове функции значение даты «Date» нужно указывать в формате «ГГГГММДД». Например, «20050527» означает 27 мая 2005 г. Значение времени «Time» нужно указывать в формате «ЧЧММСС». Например, «163500» означает 16 ч. 35 мин. Функция округляет параметр «Time» до ближайшего меньшего значения времени, соответствующего интервалу графика. Например, если интервал графика – «5 минут», а значение параметра «Time» – «163700», то функция округлит значения параметр «Time» до «163500», а если интервал графика – «60 минут», то «163700» округлится до «160000».

Возвращаемый функцией ассоциативный массив содержит следующие поля:

№	Параметр	Тип	Описание
1	COUNT	DOUBLE	Количество линий, образующих индикатор
2	TIME	DOUBLE	Точное время свечи
3	LINES	DOUBLE	Коллекция линий, в которой каждый элемент коллекции содержит ассоциативный массив (MAP)

Каждый элемент коллекции линий содержит ассоциативный массив (MAP) со следующими параметрами:

№	Параметр	Тип	Описание
1	NAME	STRING	Название линии (из легенды)
2	OPEN	DOUBLE	Цена открытия в интервале времени
3	CLOSE	DOUBLE	Цена закрытия в интервале времени
4	HIGH	DOUBLE	Наибольшее значение цены в интервале времени
5	LOW	DOUBLE	Наименьшее значение цены в интервале времени
6	VOLUME	DOUBLE	Суммарный объем сделок в интервале

Пример:

```
slice = Get_Candle_Ex ("all", 20070511, 170000)
LineCount = Get_Value (slice,"COUNT")
time = Get_Value (slice,"TIME")
lines = Get_Value (slice,"LINES")
FOR lineID FROM 0 TO LineCount-1
  line = Get_Collection_Item (lines, lineID)
  open = Get_Value (line,"OPEN")
  line_name = Get_Value (line,"NAME")
  volume = Get_Value (line,"VOLUME")
END FOR
```

## 8.21 Функции для работы с заявками

Функции предназначены для создания заявок и отправки их в торговую систему.

### 8.21.1 SEND\_TRANSACTION

Отправляет заявку с параметрами, указанными в массиве «trans\_params» и ожидает ответа торговой системы в течение «wait\_timeout\_for\_replay» (в секундах, не менее 5). Элементы массива «trans\_params» заполняются по правилам создания строки для импорта транзакций из файла (подробнее см. п.6.11 Раздела 6 «Совместная работа с другими приложениями»).

MAP SEND\_TRANSACTION (DOUBLE wait\_timeout\_for\_replay, MAP trans\_params)

Транзакции, выполняющие групповое снятие заявок, не поддерживаются:

- «KILL\_ALL\_ORDERS» – снять все заявки из торговой системы,
- «KILL\_ALL\_STOP\_ORDERS» – снять все стоп-заявки,
- «KILL\_ALL\_NEG\_DEALS» – снять все заявки на внебиржевые сделки и заявки на сделки РЕПО.

Функция возвращает ассоциативный массив (MAP), содержащий результат обработки заявки:

№	Параметр	Тип	Описание
1	RESULT	DOUBLE	Результат выполнения операции. Возможные значения: «1» – выполнено успешно, «0» – произошла ошибка
2	RESULT_EX	STRING	Расширенная диагностика выполнения операции. Может принимать значения, соответствующие полю «STATUS» в «.tro»-файле при импорте транзакций (см. п. 6.11.4 Раздела 6 «Совместная работа с другими приложениями»)
3	ORDER_NUMBER	DOUBLE	Регистрационный номер заявки в торговой системе
4	DESCRIPTION	STRING	Текстовый комментарий с ответом сервера QUIK или торговой системы

Пример:

```
\n\nnew_global("trans_params", "")\nnew_global("trans_result", "")\ntrans_params = CREATE_MAP ()\ntrans_params = set_value (trans_params, "TRANS_ID", "333")\ntrans_params = set_value (trans_params, "ACTION", "NEW_ORDER")
```

```

trans_params = set_value (trans_params, "CLASSCODE", "TQBR")
trans_params = set_value (trans_params, "SECCODE", "HYDR")
trans_params = set_value (trans_params, "ACCOUNT", "L01-00000F00")
trans_params = set_value (trans_params, "OPERATION", "B")
trans_params = set_value (trans_params, "PRICE", "7.561")
trans_params = set_value (trans_params, "QUANTITY", "1")
trans_params = set_value (trans_params, "CLIENT_CODE", "")
trans_params = set_value (trans_params, "TYPE", "L")
trans_result = SEND_TRANSACTION (30, trans_params)
WRITELN ("qpile_trans.log", get_value (curr_datetime, "DATETIME") & ": " & "Result: "
& get_value (trans_result, "RESULT") & ", Result_ex: " & get_value (trans_result,
"RESULT_EX") & ", OrderNum: " & get_value (trans_result, "ORDER_NUMBER") & ",
Description: " & get_value (trans_result, "DESCRIPTION"))
\

```

## 8.22 Функции для работы с метками

Функции предназначены для построения меток и установки их на график.

### 8.22.1 ADD\_LABEL

Добавляет метку с заданными параметрами.

DOUBLE      ADD\_LABEL (STRING tag, MAP label params)

В окно, где отображен график с идентификатором «tag» добавляется метка. В случае успешного добавления функция возвращает идентификатор метки или (-1), если добавить метку не удалось.

Формат массива MAP содержит следующие параметры метки:

№	Параметр	Тип	Описание
1	TEXT	STRING	Подпись метки (если подпись не требуется, то пустая строка)
2	IMAGE_PATH	STRING	Путь к картинке, которая будет отображаться в качестве метки (пустая строка, если картинка не требуется)
3	ALIGNMENT	STRING	Расположение картинки относительно текста (возможно 4 варианта: LEFT, RIGHT, TOP, BOTTOM)
4	YVALUE	DOUBLE	Значение параметра на оси Y, к которому будет привязана метка
5	DATE	DOUBLE	Дата в формате «ГГГГММДД», к которой привязана метка

№	Параметр	Тип	Описание
6	TIME	DOUBLE	Время в формате «ЧЧММСС», к которому будет привязана метка
7	R	DOUBLE	Красная компонента цвета в формате RGB. Число в интервале [0;255]
8	G	DOUBLE	Зеленая компонента цвета в формате RGB. Число в интервале [0;255]
9	B	DOUBLE	Синяя компонента цвета в формате RGB. Число в интервале [0;255]
10	TRANSPARENCY	DOUBLE	Прозрачность метки в процентах. Значение должно быть в промежутке [0; 100]
11	TRANSPARENT_BACKGROUND	DOUBLE	Прозрачность метки. Возможные значения: «0» – прозрачность отключена, «1» – прозрачность включена
12	FONT_FACE_NAME	STRING	Название шрифта (например «Arial»)
13	FONT_HEIGHT	DOUBLE	Размер шрифта
14	HINT	STRING	Текст всплывающей подсказки

Пример:

```
\
label_params=create_map()
label_params=set_value(label_params,"TEXT", "Текст метки")
label_params=set_value(label_params,"IMAGE_PATH", "image.bmp")
label_params=set_value(label_params,"ALIGNMENT", "LEFT")
label_params=set_value(label_params,"YVALUE", 2000)
label_params=set_value(label_params,"DATE", 20080616)
label_params=set_value(label_params,"TIME", 220000)
label_params=set_value(label_params,"R", 0)
label_params=set_value(label_params,"G", 0)
label_params=set_value(label_params,"B", 200)
label_params=set_value(label_params,"TRANSPARENCY", 10)
label_params=set_value(label_params,"FONT_FACE_NAME", "Tahoma")
label_params=set_value(label_params,"FONT_HEIGHT", 12)
label_params=set_value(label_params,"HINT", "Подсказка")
id=ADD_LABEL("ALL", label_params)
,
```

Результат представляется следующим образом: на графике с идентификатором "ALL" будет добавлена метка с параметрами, указанными в массиве label\_params.

### 8.22.2 DELETE\_LABEL

Удаляет метку с заданными параметрами.

DOUBLE DELETE\_LABEL(String tag, DOUBLE id)

Удаляет метку с графика с текстовым идентификатором «tag» по заданному идентификатору id. В случае успешного удаления возвращает 1, в случае неудачи 0.

Пример:

```
\nerr=DELETE_LABEL("ALL", id)\n'
```

Результат действия функции будет удаление метки на графике с идентификатором «ALL».

### 8.22.3 DELETE\_ALL\_LABELS

Команда удаляет все метки на диаграмме с указанным графиком.

DOUBLE DELETE\_ALL\_LABELS(String tag)

Удаляет все нанесенные метки с диаграммы, содержащей график с идентификатором tag. Если на этой диаграмме отображаются графики с разными идентификаторами, то этой командой будут удалены метки со всех графиков, а не только с указанного графика.

Пример:

```
\nerr=DELETE_ALL_LABELS("ALL")\n'
```

Результатом действия функции будет удаление всех меток на графике.

### 8.22.4 GET\_LABEL\_PARAMS

Команда позволяет получить параметры меток.

MAP GET\_LABEL\_PARAMS(String tag, DOUBLE id)

Возвращает параметры метки с указанным идентификатором; если метка с указанным идентификатором не существует – возвращается пустой MAP.

- «tag» – идентификатор любого индикатора на графике, в котором расположена метка,
- «id» – индекс или порядковый номер метки. Начинается с единицы.

Пример:

```
\nnew_params=GET_LABEL_PARAMS("ALL", id)\n'
```

Результатом действия является получение данных метки. Если такой метки не существует, то возвратившиеся значения будут пустыми.

### 8.22.5 SET\_LABEL\_PARAMS

Функция задает параметры для метки с указанным идентификатором.

DOUBLE SET\_LABEL\_PARAMS(String tag, DOUBLE id, MAP new label params)

Она позволяет задать новые параметры для метки. В случае успешного обновления параметров функция возвращает 1, в случае неудачи 0.

Пример:

```
\nerr=SET_LABEL_PARAMS("ALL", id, label_params)\n'
```

Результатом действия функции будет изменение существующих параметров на заданные функцией.

## 8.23 Сервисные функции

### 8.23.1 GET\_TRADE\_DATE

Возвращает дату текущей торговой сессии.

MAP GET\_TRADE\_DATE ()



Функция возвращает ассоциативный массив (MAP), содержащий параметры:

№	Параметр	Тип	Описание
1	DATE	STRING	Торговая дата в виде строки «ДД.ММ.ГГГГ»
2	YEAR	DOUBLE	Год
3	MONTH	DOUBLE	Месяц
4	DAY	DOUBLE	День

Пример:

```
\n\nwriteln(log_file_name, get_value(GET_TRADE_DATE(), "Date"))\n
```

Записывает в файл строку вида:

```
02.06.2004
```

### 8.23.2 GET\_DATETIME

Возвращает текущие дату и время.

MAP GET\_DATETIME ()

Функция возвращает ассоциативный массив (MAP), содержащий параметры:

№	Параметр	Тип	Описание
1	DATETIME	STRING	Торговая дата в виде строки «ДД.ММ.ГГГГ ЧЧ:ММ:СС.ссс», где «ссс» – миллисекунды
2	YEAR	DOUBLE	Год
3	MONTH	DOUBLE	Месяц
4	DAY	DOUBLE	День
5	DAYOFWEEK	DOUBLE	Порядковый номер дня недели (воскресенье – 0, понедельник – 1, вторник – 2, среда – 3, четверг – 4, пятница – 5, суббота – 6)
6	HOUR	DOUBLE	Час

№	Параметр	Тип	Описание
7	MIN	DOUBLE	Минута
8	SEC	DOUBLE	Секунда
9	MILLISEC	DOUBLE	Миллисекунда

Пример:

```
\
writeln(log_file_name, get_value(GET_DATETIME(), "Datetime"))
,
```

Записывает в файл строку вида:

```
02.06.2004 16:57:34.460
```

### 8.23.3 APPLY\_SCALE

Возвращает строку с числом, полученным округлением числа «without\_scale» до разрядности «scale».

STRING APPLY\_SCALE (DOUBLE without\_scale, DOUBLE scale)

### 8.23.4 IS\_CONNECTED

Функция предназначена для определения состояния подключения клиентского места к серверу. Возвращает 1, если клиентское место подключено и 0, если не подключено.

IS\_CONNECTED ()

### 8.23.5 GET\_INFO\_PARAM

Функция возвращает значения параметров информационного окна (пункт меню Система/О программе/Информационное окно...).

STRING GET\_INFO\_PARAM (STRING param\_name)

Параметр «param\_name» может принимать значения, представленные в таблице.

Значение параметра	Описание	Значение параметра	Описание
VERSION	Версия программы	MESSAGESENT	Передано сообщений
TRADEDATE	Дата торгов	ALLSENT	Передано всего байт
SERVERTIME	Время сервера	BYTESENT	Передано полезных байт
LASTRECORDTIME	Время последней записи	BYTESPERSECSSENT	Передано за секунду
NUMRECORDS	Число записей	MESSAGESRECV	Принято сообщений
LASTRECORD	Последняя запись	BYTESRECV	Принято полезных байт
LATERECORD	Отставшая запись	ALLRECV	Принято всего байт
CONNECTION	Соединение	BYTESPERSECRECV	Принято за секунду
IPADDRESS	IP-адрес сервера	AVGSENT	Средняя скорость передачи
IPPORT	Порт сервера	AVGRECV	Средняя скорость приема
IPCOMMENT	Описание соединения	LASTPINGTIME	Время последней проверки связи
SERVER	Описание сервера	LASTPINGDURATION	Задержка данных при обмене с сервером
SESSIONID	Идентификатор сессии	AVGPINGDURATION	Средняя задержка данных
USER	Пользователь	MAXPINGTIME	Время максимальной задержки
USERID	ID пользователя	MAXPINGDURATION	Максимальная задержка данных
ORG	Организация		
MEMORY	Занято памяти		
LOCALTIME	Текущее время		
CONNECTIONTIME	Время на связи		

### 8.23.6 BREAKPOINT

Функция предназначена для прерывания режима расчета и вывода окна «Отладка», в котором пользователь сможет наблюдать за дальнейшей работой программы.

BREAKPOINT ()

Команда не содержит никаких параметров и результатом ее действия является остановка выполнения программы, выделение красным маркером следующей за ней функции и вызов окна «Отладка», в котором пользователь сможет отслеживать выполнение кода скрипта. При необходимости, команда может содержаться в коде произвольное количество раз.

## 8.24 Отладка программ QPILE

Окно «Отладка» предназначено для пошагового контроля за исполнением кода скриптов. Оно позволяет отслеживать выполнение кода программ, написанных на языке QPILE.

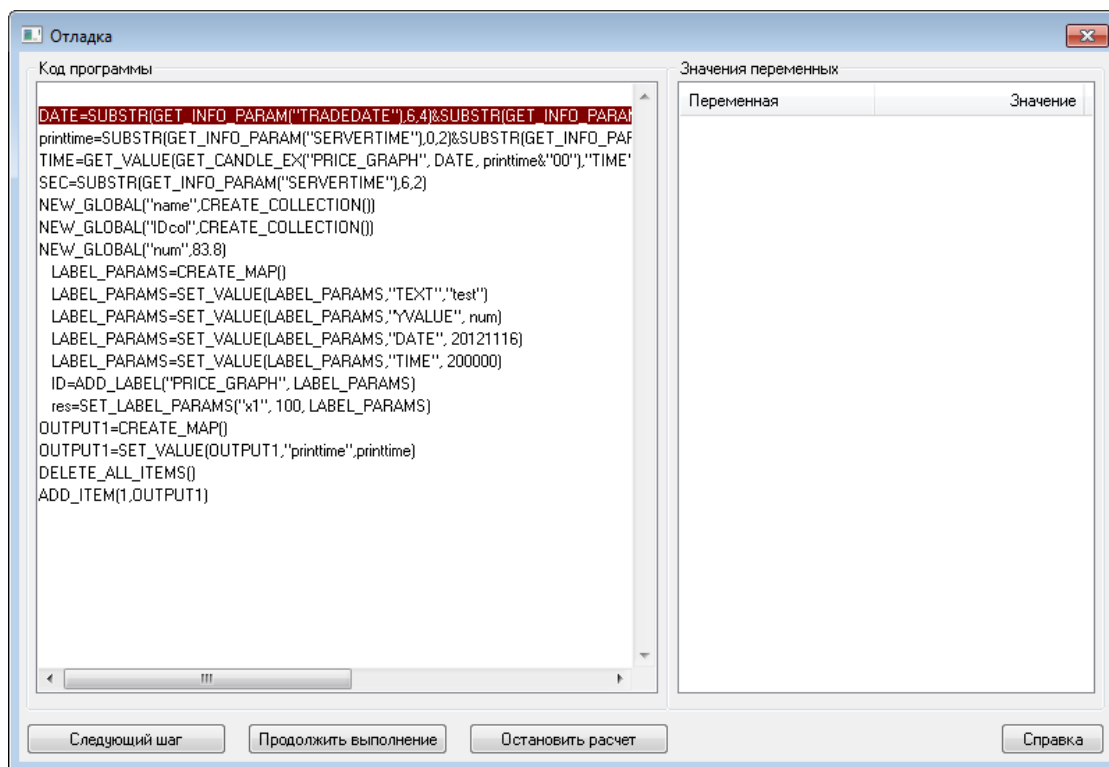
Окно может быть вызвано следующим образом:

- из таблицы QPILE, выбрав команду контекстного меню «Начать расчет в режиме отладки»;
- из кода программы функцией `breakpoint()`.

Окно содержит два больших поля:

- «Код программы» – выводит код языка QPILE;
- «Значения переменных» – выводятся переменные и их значения в виде двух столбцов.

При вызове окна с помощью оператора `breakpoint()` следующая за ним строка программы будет отмечена красным цветом.



Окно «Отладка» содержит следующие кнопки:

- «Следующий шаг» – выполняет один оператор в коде.
- «Продолжить выполнение» – продолжает выполнение программы, пока не будет нажата кнопка «Остановить расчет», найден еще один оператор `breakpoint()` или не будет достигнут конец программы.
- «Остановить расчет» – останавливает расчет на последнем выполненном операторе.

Доступные функции:

- «F5» – продолжить выполнение программы;
- «Shift»+«F5» – прекратить отладку программы и закрыть данное окно;
- «F10» – перейти к следующей строке.

## ПРИЛОЖЕНИЕ 1. Синтаксис команд языка QPILE

```
Program:
Statement_List

Statement_List:
Statement "\n"
Statement_List "\n" Statement

Statement:
NAME=Expression
IFOperator
FOROperator
FUNCDescr
CONTINUE      //пропускает выполнение операторов до конца текущего Statement_List
BREAK  //начинает выполнение оператора следующего за текущим Statement_List
RETURN //выход из текущего управляющего блока - из тела функции или из всей программы

IFOperator:
"IF" Condition "\n"
Statement_List
"ELSE" "\n"
Statement_List
"END IF"

Condition
Condition "OR" Condition
Condition "AND" Condition
"("Condition") "
PrimaryCondition

PrimaryCondition
Expression "==" Expression
Expression "=" Expression //с той же семантикой, что и "=="
Expression ">=" Expression
Expression "<=" Expression
Expression ">"Expression
Expression "<"Expression
Expression "!=" Expression
Expression "<>" Expression //с той же семантикой, что и "!="

FOROperator:
"FOR" NAME "IN" ArgList "\n"
    Statement_List
"END FOR"
```

```

"FOR" NAME "FROM" Expression "TO" Expression "\n"
    Statement_List
"END FOR"

ArgList
NAME    // в этом случае переменная с именем NAME должна содержать значение вида
ArgList1
ArgList1

ArgList1:
Expression
ArgList "," Expression

FUNCDescr
"FUNC" NAME "(" FargList ")" "\n"
Statement_List
"END FUNC"

FargList
NAME
FargList "," NAME

Expression:
Expression "+" Term
Expression "-" Term
Expression "&" Term //конкатенация строк
Term

Term:
Term "/" Primary
Term "*" Primary
Primary

Primary:
NUMBER
STRINGNAME    //значение переменной с именем NAME
"-" Primary
"(" Expression ")"
FunctionCall

FunctionCall
FNAME "(" ArgList1 ")"

NUMBER:
Digits

```

```
Digits "." DigitsSTRING: //определяется стандартно  
NAME: //определяется стандартно
```

```
Ключевые слова = {IF, ELSE, FOR, IN, TO, FROM, AND, OR, RESULT, FUNC, END FUNC, END  
FOR, END IF, CONTINUE, BREAK, RETURN}
```

## ПРИЛОЖЕНИЕ 2. Рекомендации к составлению программ на QPILE

1. Функции для работы со структурными переменными, такие как «SET\_VALUE», «ADD\_COLLECTION\_ITEM», «REMOVE\_COLLECTION\_ITEM» возвращают измененное значение коллекции или массива. Поскольку все параметры передаются в функцию по значению, то использование данных функций в качестве процедур приводит к потере сделанных в них изменений.

Например:

```
clientscol=INSERT_COLLECTION_ITEM(clientscol,0,initmap)  
'правильно  
  
INSERT_COLLECTION_ITEM(clientscol,0,initmap)  
'неправильно, в этом случае после вызова функции clientscol будет содержать то же  
значение, что и до ее вызова
```

2. Возвращаемые функциями значения могут оказаться строковыми, хотя и представлять вещественное число. В этих случаях нужно использовать сложение с «0» для преобразования результата к численному значению. Например, для корректной инициализации переменной в операторе присваивания вида «v=GET\_VALUE()» стоит написать «v=0+GET\_VALUE()». В последнем случае гарантированно достигается вещественное значение переменной.  
Аналогичным образом, для приведения вещественного значения к строковому может использоваться операция склейки с пустой строкой «v=""&GET\_VALUE()».
3. Произвольные пользовательские типы могут быть сформированы путем комбинации коллекций и ассоциативных массивов. Например, если нужен список структур типа:

```
c=struct{  
  openbal:double  
  closebal:double  
  clientcode:string}
```



то его можно представить как коллекцию ассоциативных массивов, в каждом из которых будет три ключа – «OPENBAL», «CLOSEBAL», «CLIENTCODE». Код для инициализации такой структуры:

```
initmap=CREATE_MAP()
initmap=SET_VALUE(initmap,"OPENBAL",0)
initmap=SET_VALUE(initmap,"CLOSEBAL",0)
initmap=SET_VALUE(initmap,"CLIENTCODE","")
clientscol=CREATE_COLLECTION()
FOR i FROM 0 TO 10
clientscol=INSERT_COLLECTION_ITEM(clientscol,0,initmap)
END FOR
```

После чего, для доступа к полю «OPENBAL» 5-го клиента надо написать:

```
openbal = GET_VALUE(GET_COLLECTION_ITEM(clientscol,5),"OPENBAL")
```

Если вместо коллекции таких записей использовать массив с ключом представляющим из себя код клиента, то можно будет обращаться к значениям структуры клиента без указания его индекса в массиве, а зная только клиентский код:

```
clientsmap=CREATE_MAP()
FOR i FROM 0 TO 10
clientsmap=SET_VALUE(clientsmap,"Q" & i, initmap)
END FOR
```

и последующее обращение к значению поля «OPENBAL» для клиента с кодом «Q5»:

```
openbal = GET_VALUE(GET_VALUE(clientsmap,"Q5"),"OPENBAL")
```

- 4. «MODIFY\_ITEM» не выполняет действий, если указанной строки в таблице «OWN» еще нет.** Поэтому надо предварительно проверять ее наличие. При построении пользовательской таблицы на каждой итерации изменяющей в себе строку полезным будет код:

```
new_global("first_time_flag",0)

if first_time_flag==0
add_item(1, SAMPLE)
first_time_flag=1
```

```
else  
  modify_item(1, SAMPLE)  
end if
```

При первом запуске создается строка номер 1 со значениями полей из переменной «SAMPLE» насчитанной ранее, а на последующих итерациях строка номер 1 модифицируется.