

# Драйвер Exellio FP

Руководство пользователя

## Оглавление

Драйвер Exellio FP .....	5
Порядок работы с OLE – сервером .....	5
1. Инициализация работы .....	5
2. Завершение работы .....	5
3. Фискальный чек .....	5
4. Не фискальный чек (чек комментариев) .....	6
5. Чек возврата .....	6
6. X-отчет .....	6
7. Z-отчет .....	6
8. Отчеты за период .....	6
9. Печать копии чека .....	6
10. Служебный ввод/вывод .....	6
11. Отладка OLE – сервера .....	7
12. Возвращаемые значения функций библиотеки .....	7
13. Коды ошибок OLE – сервера .....	7
14. Значения переключателей на фискальном регистраторе .....	7
15. Работа регистратора Экселлио в терминальном режиме с 1С:Предприятие. ....	8
16. Состояние фискального регистратора .....	8
Инициализация .....	9
Открытие COM порта фискального регистратора (OpenPort) .....	9
Закрытие COM порта фискального регистратора (ClosePort) .....	10
Запись текущих настроек в энергонезависимую FLASH-память (SaveSettings) .....	10
Установка высоты штрих-кода (SetBarcodeHeight) .....	10
Установка контрастности печати (SetPrintDensity) .....	10
Включение/выключение автоматического открытия денежного ящика (EnableAutoOpenDrawer) .....	11
Установка названий дополнительных типов оплат (SetTaxName) .....	11
Программирование заголовков чека (SetHeaderFooter) .....	11
Чтение заголовков чека (GetHeader) .....	11
Включение/выключение автообрезки чека (EnableCutCheck) .....	12
Включение/выключение уменьшенного шрифта контрольной ленты (EnableSmallFont) .....	12
Включение/выключение запрета печати отчетов на контрольной ленте (EnableCRReport) .....	12
Включение/выключение печати логотипа на чеке (EnableLogo) .....	12
Установка даты и времени в фискальном регистраторе (SetDateTime) .....	12
Фискализация фискального регистратора (Fiscalise) .....	13
Установка десятичной точки, налоговых групп и ставок налогов (SetMulDecCurRF) .....	13
Установка режима продаж (SetTaxType) .....	13
Установка заводского номера фискального регистратора (SetSerialNum) .....	14
Установка фискального номера регистратора (SetFiscalNumber) .....	14
Установка налогового/идентификационного номера (SetTaxNumber) .....	14
Программирование паролей операторов (SetOperatorPassword) .....	14
Программирование имен операторов (SetOperatorName) .....	15
Обнуление сумм по операторам (ClearOperator) .....	15
Установка пароля администратора (SetAdminPassword) .....	15
Удаление пароля оператора (ClearOperatorPassword) .....	15
Продажа .....	15
Открыть не фискальный чек (OpenNonfiscalReceipt) .....	16
Печать не фискального текста (PrintNonfiscalText) .....	16
Закрыть не фискальный чек (CloseNonfiscalReceipt) .....	16
Открыть фискальный чек (OpenFiscalReceipt) .....	16

Регистрация продажи (Registrltem) .....	17
Регистрация продажи с изменением цены (RegistrltemEx) .....	17
Регистрация продажи и отображение на индикатор клиента продажи (RegistrAndDisplayltem) .....	18
Регистрация продажи, изменение цены и отображение на индикатор (RegistrAndDisplayltemEx) .....	18
Печать фискального текста (PrintFiscalText) .....	18
Подитог, скидка/надбавка на чек (SubTotal) .....	19
Оплата чека (Total) .....	19
Оплата чека и закрытие фискального чека (TotalEx) .....	20
Печать штрих-кода (PrintBarCode) .....	20
Печать разделительной линии (PrintLine) .....	20
Закрытие фискального чека (CloseFiscalReceipt) .....	21
Отмена чека (CancelReceipt) .....	21
Открыть возвратный чек (OpenReturnReceipt) .....	21
Печать копии чека (MakeReceiptCopy) .....	21
Печать нулевого чека (PrintNullCheck) .....	22
Печать суммовой скидки на группу товара (AbsDiscGrp) .....	22
Печать процентной скидки на группу товара (PerDiscGrp) .....	22
Печать суммовой скидки на налоговую группу товара (AbsDiscTax) .....	22
Печать процентной скидки на налоговую группу товара (PerDiscTax) .....	23
Регистрация продажи, с автоматическим программированием (Sale) .....	23
Регистрация продажи, с автоматическим программированием (SaleWC) .....	24
Программирование артикулов и получение информации об артикулах .....	24
Получение информации об артикулах (GetArticlesInfo) .....	24
Программирование артикула (SetArticle) .....	24
Удаление артикула/артикулов (DelArticle) .....	24
Чтение артикула (GetArticle) .....	25
Чтение первого артикула (GetFirstArticle) .....	25
Чтение следующего артикула (GetNextArticle) .....	25
Изменение цены артикула (ChangeArticlePrice) .....	26
Получение информации о первом свободном артикуле (GetFirstFreeArticle) .....	26
Получение информации о последнем свободном артикуле (GetLastFreeArticle) .....	26
Отчеты .....	27
Дневной отчет без обнуления, X - отчет (XReport) .....	27
Дневной отчет с обнулением, Z - отчет (ZReport) .....	27
Отчет налоговых ставок за период (PrintTaxReport) .....	27
Полный периодический отчет фискальной памяти по номерам Z - отчетов (PrintRepByNumFull) .....	27
Полный периодический отчет фискальной памяти по датам Z - отчетов (PrintRepByDateFull) .....	27
Краткий периодический отчет фискальной памяти по номерам Z - отчетов (PrintRepByNum) .....	28
Краткий периодический отчет фискальной памяти по датам Z - отчетов (PrintRepByDate) .....	28
Отчет по артикулам(PrintRepByArt) .....	28
Отчет по операторам(PrintRepByOperator) .....	28
Дневной отчет с обнулением, Z - отчет (ZReportWC) .....	29
Передача информации в прикладную программу .....	29
Получить дату и время (GetDateTime) .....	29
Получить информации о суммах фискальных или возвратных чеков (LastFiscalClosure) .....	29
Получить суммы с момента последнего Z - отчета (GetCurrentSums) .....	29

Получить суммы коррекций за день (GetCorectSums) .....	30
Получить размер фискальной памяти (GetFreeClosures) .....	30
Получить статус фискального регистратора (GetStatus) .....	30
Получить состояние фискальной транзакции (GetFiscalClosureStatus) .....	31
Получить диагностическую информацию (GetDiagnosticInfo) .....	31
Получить налоговые ставки регистратора (GetCurrentTaxRates) .....	31
Получить налоговый номер регистратора (GetTaxNumber) .....	32
Получить информацию о чеке (GetReceiptInfo) .....	32
Получить информацию о дневных суммах (GetDayInfo) .....	32
Получить суммы накопленные по оператору (GetOperatorInfo) .....	33
Получить номер последнего чека (GetLastReceiptNum) .....	33
Получить информацию о фискальности/нефискальности регистратора (isFiscalised) .....	33
Получить продолжительность смены (GetSmenLen) .....	33
Получить дату последнего Z-отчета (GetLastClosureDate) .....	33
Получить сумму в денежном ящике (GetSumInCash) .....	34
Получить время последнего обмена с налоговой (GetLastDPAExchangeTime) .....	34
Получить время последнего обмена с налоговой (GetLastDPAExchangeTimeEx) .....	34
Получения значений битов состояния ФР (GetStatusBit) .....	34
Получить информацию о модели фискального регистратора (FPMModel) .....	35
Команды принтера .....	35
Прогон ленты фискальный принтером (AdvancePaper) .....	35
Прогон ленты фискальный принтером расширенный(AdvancePaperEx) .....	35
Отрезать чек (CutReceipt) .....	35
Дисплей .....	35
Очистка дисплея (ClearDisplay) .....	35
Вывод текста в нижний ряд (DisplayTextLL) .....	36
Вывод текста в верхний ряд (DisplayTextUL) .....	36
Вывод даты и времени (DisplayDateTime) .....	36
Непосредственный вывод на дисплей (DisplayFreeText) .....	36
Отладка .....	36
Включить отображение ошибок OLE-сервером (ShowError) .....	36
Включить запись отладочной информации в текстовый лог-файл (Debugger) .....	37
Другие .....	37
Открыть денежный ящик (OpenDrawer) .....	37
Открыть денежный ящик с регулируемой длиной импульса (OpenDrawerEx) .....	37
Служебный ввод/вывод денег (InOut) .....	37
Печать диагностической информации (PrintDiagnosticInfo) .....	38
Звуковой сигнал (Sound) .....	38
Звуковой сигнал произвольный (SoundEx) .....	38
КСЕФ .....	38
Прочитать чеки из КСЕФ по номерам (ReadKSEFByNum) .....	38
Чтение следующей строки КСЕФ (ReadNextKSEFRec) .....	38
Прочитать чеки из КСЕФа по датам (ReadKSEFByDate) .....	39
Распечатать чеки из КСЕФ по номерам (PrintKSEFByNum) .....	39
Распечатать чеки из КСЕФ по датам (PrintKSEFByDate) .....	39
Прочитать чек в формате XML из КСЕФ (ReadKSEFByNumXML) .....	39

## Драйвер Exellio FP

Библиотека Exellio FP OLE – предназначена для работы прикладных программ с фискальными регистраторами Экселлио FPU-550, Экселлио FPU-260 и Экселлио LP1000 в операционных системах MS Windows 98 / ME / NT / 2000 / XP / 2003 / Vista / Windows 7 / Windows 8.

OLE сервер может использоваться с любыми языками программирования и средствами разработки, поддерживающими технологию COM. Например: 1С 7.7, 1С 8.x, Delphi, Visual C++, Visual Basic Script (VBS), Visual Basic for Applications (VBA), Visual Basic, CBuilder и др.

Имя объекта для создания: **ExellioFP.FiscalPrinter**

## Порядок работы с OLE – сервером

### 1. Инициализация работы

Для начала работы с библиотекой необходимо вызвать функцию [OpenPort](#). При выполнении команды происходит обращение к фискальному регистратору и поэтому он **ОБЯЗАТЕЛЬНО** должен быть включен. Выполнение функции приводит к открытию последовательного порта. Функция не выполняется если последовательный порт уже открыт другой программой или если фискальный регистратор не включен.

### 2. Завершение работы

Для завершения работы с фискальным регистратором необходимо дождаться завершения последней функции, а затем вызвать функцию [ClosePort](#). Завершение работы приводит к отключению последовательного порта, к которому подключен фискальный регистратор.

### 3. Фискальный чек

Для получения фискального чека необходимо выполнить следующее:

- Запрограммировать артикул в фискальный регистратор с помощью функции [SetArticle](#). Также вы можете изменить цену на уже запрограммированный артикул с помощью функции [ChangeArticlePrice](#). Для фискального регистратора Экселлио LP-1000 допустимо использовать функцию [ChangeArticlePrice](#) прямо в открытом фискальном чеке.
- Открыть фискальный чек вызовом функции [OpenFiscalReceipt](#).
- Выполнить необходимое количество продаж, снабдив их необходимым количеством комментариев, используя для этого функции [RegisterItem](#) или [RegistrAndDisplayItem](#) и [PrintFiscalText](#). Для фискального регистратора Экселлио LP-1000 удобно использовать функцию [RegistrItemEx](#), которая отображает на дисплей товар и позволяет продавать его по свободной цене.
- Выполнить функцию [SubTotal](#) для уточнения накопленной суммы по чеку, а также для осуществления скидки/надбавки на весь чек.
- Выполнить один или несколько раз (для комбинированной формы оплаты) функции [Total](#) таким образом, чтобы полностью покрыть сумму по чеку (возможно со сдачей). Для фискального регистратора Экселлио LP-1000 можно использовать функцию [TotalEx](#), которая позволит при полной оплате чека не использовать функцию [CloseFiscalReceipt](#).
- Выполнить функцию закрытия чека [CloseFiscalReceipt](#).

## 4. Не фискальный чек (чек комментариев)

Для получения нефискального чека необходимо выполнить следующее:

- Открыть не фискальный чек функцией [OpenNonfiscalReceipt](#).
- Напечатать необходимое количество комментариев функцией [PrintNonfiscalText](#).
- Закрыть нефискальный чек функцией [CloseNonfiscalReceipt](#).

## 5. Чек возврата

Для печати возвратного чека необходимо выполнить все действия, описанные в п.3 используя вместо функции открытия фискального чека ([OpenFiscalReceipt](#)) функцию открытия чека возврата ([OpenReturnReceipt](#)).

## 6. X-отчет

Для печати X-отчета достаточно вызвать функцию [XReport](#).

## 7. Z-отчет

Для печати Z-отчета достаточно вызвать функцию [ZReport](#).

## 8. Отчеты за период

Для печати периодических отчетов достаточно вызвать одну из функций:

- [PrintRepByNumFull](#) – полный отчет по номерам Z-отчетов
- [PrintRepByDateFull](#) – полный отчет по датам Z-отчетов
- [PrintRepByNum](#) – краткий отчет по номерам Z-отчетов
- [PrintRepByDate](#) – краткий отчет по датам Z-отчетов

## 9. Печать копии чека

Для печати копии чека достаточно вызвать функцию [MakeReceiptCopy](#).

## 10. Служебный ввод/вывод

Для служебного внесения/вынесения денег достаточно выполнить функцию [InOut](#).

## 11. Отладка OLE – сервера

Для включения ведения log-файла отладки вызовите функцию [Debugger](#) с входным параметром True, для остановки отладки вызовите функцию с параметром False.

## 12. Возвращаемые значения функций библиотеки

- OLE – сервер записывает код ошибки в свойство (переменную) с именем [LastError](#), а текст ошибки в свойство (переменную) [LastErrorText](#).
- Вы можете с помощью функции [ShowError](#) с параметром True (Истина) указать OLE – серверу о необходимости отображения описания и кода ошибки в окне сообщений.
- Уточнить состояние ФР можно прочитав значения битов состояния, функцией [GetStatusBit](#).

## 13. Коды ошибок OLE – сервера

0	Отсутствие ошибки
1	Ошибка открытия COM-порта
2	Соединение не установлено
3	Таймаут при чтении блока/нет связи с ФР
6	Ошибка NAK
7	Ошибка ФР, уточнение ошибки можно получить в свойстве <b>LastErrorText</b> , а также функцией <a href="#">GetStatusBit</a>
8	ФР заблокирован. Нет связи с сервером 72 часа
40	Ошибка получения информации по коду из ФР (для функции <a href="#">Sale</a> )
41	Ошибка программирования нового товара в ФР (для функции <a href="#">Sale</a> )
42	Налоговая группа не соответствует запрограммированному в ФР (для функции <a href="#">Sale</a> )
43	Наименование не соответствует запрограммированному в ФР (для функции <a href="#">Sale</a> )
44	Товарная группа не соответствует запрограммированному в ФР (для функции <a href="#">Sale</a> )
52	Нет лицензии на ФР

## 14. Значения переключателей на фискальном регистраторе

SW1...SW8 – переключатели на нижней крышке фискального регистратора.

Sw1 и Sw2 скорость передачи данных по COM порту:

Переключатель Sw1	Переключатель Sw2	Скорость
OFF (выключен)	OFF (выключен)	9600 бод
ON (включен)	OFF (выключен)	19200 бод
OFF (выключен)	ON (включен)	57600 бод
ON (включен)	ON (включен)	115200 бод

Sw3 OFF: Автообрезка запрещена.

ON: Автообрезка разрешена.

Sw4 OFF: Ограниченный доступ к индикатору клиента внутри открытого чека

- ON: Полный доступ к индикатору клиента из программы
- Sw5 OFF: Кодовая страница Windows 1251
- ON: Кодовая страница MS-DOS
- Sw6 Кодовая страница для отображения на индикатор клиента
- Sw7 OFF: Обычный шрифт контрольной ленты
- ON: Уменьшенный шрифт контрольной ленты
- Sw8 Не используется в Datecs FP3530T, в Экселлио:
  - OFF: Обычный шрифт чековой ленты
  - ON: Уменьшенный шрифт чековой ленты

## 15. Работа регистратора Экселлио в терминальном режиме с 1С:Предприятие.

Для работы фискального регистратора Экселлио с программой 1С:Предприятие в терминальном режиме предусмотрена функция **SetReadTimeout(long lMsec)**

**Назначение:** Функция предназначена для установки максимального времени ожидания прихода информации от регистратора к ПК (т.е. для установки таймаута). Данную функцию необходимо вызывать после создания объекта и перед открытием СОМ-порта.

### **Параметры:**

**long lMsec** – целое число, время таймаута в миллисекундах. По умолчанию 1000 мс (1 секунда), при плохих коммуникациях рекомендуется увеличивать (иногда до 10000 мс). Рекомендуется подбирать экспериментальным путем.

## 16. Состояние фискального регистратора

Описание байтов состояния фискального регистратора.

Значения битов можно получить функцией [GetStatusBit](#)

### **Байт S0 – общее назначение**

- 0.7 = 1 резерв
- 0.6 = 1 резерв
- 0.5 = 1 Общая ошибка – этот бит устанавливается всегда когда установлен один из битов, маркированный символом '#’.
- 0.4 = 1# Механизм печатающего устройства неисправен
- 0.3 = 1 Не подключен дисплей
- 0.2 = 1 Дата и время не были установлены с момента последнего аварийного обнуления RAM
- 0.1 = 1# Код полученной команды неверен
- 0.0 = 1# Полученные данные содержат синтаксическую ошибку

### **Байт S1 – общее назначение**

- 1.7 = 1 резерв
- 1.6 = 1 резерв
- 1.5 = 1 Открыта крышка принтера
- 1.4 = 1# Содержимое оперативной памяти было разрушено (RAM) при включении – аварийное обнуление
- 1.3 = 1 При установленном 2.3 означает, что открыт не фискальный чек, а чек возврата.
- 1.2 = 1# Совершено аварийное обнуление оперативной памяти
- 1.1 = 1 # Выполняемая команда не разрешена для текущего фискального режима принтера
- 1.0 = 1 При выполнении команды произошло переполнение операции суммирования – Состояние 1.1 если оно установлено указывает, на то что операция не может быть выполнена

### **Байт S2 – общее назначение**

- 2.7 = 1 резерв
- 2.6 = 1 Не используется
- 2.5 = 1 Открыт нефискальный чек
- 2.4 = 1 Заканчивается (но еще не закончилась) контрольная лента



- 2.3 = 1 Открыт фискальный чек
- 2.2 = 1 Нет контрольной ленты
- 2.1 = 1 Заканчивается (но еще не закончилась) чековая или контрольная лента
- 2.0 = 1 Закончилась чековая или контрольная лента

#### **Байт S3 – состояние переключателей**

- 3.7 = 1 резерв
- 3.6 = 1 переключатель Sw7 положение ON – уменьшенный шрифт на контрольной ленте
- 3.5 = 1 переключатель Sw6 положение ON – дисплей (кодированная таблица 1251)
- 3.4 = 1 переключатель Sw5 положение ON – кодовая страница для отправки данных на принтер DOS/Windows 1251
- 3.3 = 1 переключатель Sw4 положение ON – режим ”прозрачный дисплей”
- 3.2 = 1 переключатель Sw3 положение ON – автоматическая обрезка чека
- 3.1 = 1 переключатель Sw2 положение ON – скорость последовательного порта
- 3.0 = 1 переключатель Sw1 положение ON - скорость последовательного порта

#### **байт S4: фискальная память**

- 4.7 = 1 резерв
- 4.6 = 1 резерв
- 4.5 = 1 этот бит устанавливается всегда когда установлен один из битов, маркированный символом ‘\*’ в байтах 4 или 5
- 4.4 = 1 \* Фискальная память переполнена
- 4.3 = 1 В фискальной памяти есть место по крайней мере для 50 Z-отчетов
- 4.2 = 1 Нет модуля фискальной памяти
- 4.1 = 1 Не используется
- 4.0 = 1 \* Возникла ошибка при записи в фискальную память

#### **байт S5: фискальная память**

- 5.7 = 1 резерв
- 5.6 = 1 резерв
- 5.5 = 1 Фискальной и заводской номер запрограммированы
- 5.4 = 1 Налоговые ставки определены
- 5.3 = 1 Устройство фискализировано
- 5.2 = 1 \* Не используется
- 5.1 = 1 Фискальная память отформатирована
- 5.0 = 1 \* Фискальная память установлена в режим Read Only.

## Инициализация

### **Открытие COM порта фискального регистратора (OpenPort)**

#### **OpenPort(string sPortNumber, int iBaudRate)**

**Назначение:** Функция предназначена для открытия COM порта, к которому подключен фискальный регистратор Экселлио FPU-550. Эту функцию следует вызывать первой для начала работы OLE сервера с фискальным регистратором. Фискальный регистратор в момент вызова функции **ОБЯЗАТЕЛЬНО** должен быть включен, иначе COM порт открыт не будет.

#### **Параметры:**

**sPortNumber** – Строка с именем порта, например “COM1”, “COM2”.

**iBaudRate** – целое число, скорость COM порта, к которому подключен фискальный регистратор. Может принимать значения 9600, 19200, 57600, 115200 бод. Если указанное значение не отвечает стандартным значениям, открытие COM порта происходит на скорости по умолчанию (19200 бод). Фискальный регистратор Экселлио FPU-260 может функционировать только на скорости 19200.

В фискальном регистраторе необходимо установить соответствующую скорость с помощью переключателей, которые находятся на нижней крышке фискального регистратора. Это необходимо делать при выключенном аппарате.

#### **Положение переключателей скорости:**

Переключатель Sw1 и Sw2 определяют скорость работы COM порта:

ПереключательSw1	ПереключательSw2	Скорость
OFF (выключен)	OFF (выключен)	9600 бод
ON (включен)	OFF (выключен)	19200 бод
OFF (выключен)	ON (включен)	57600 бод
ON (включен)	ON (включен)	115200 бод

**Возвращаемое значение:** Нет

## Закрытие COM порта фискального регистратора (ClosePort)

ClosePort()

**Назначение:** Функция предназначена для закрытия COM порта, открытого функцией [OpenPort](#).

**Возвращаемое значение:** Нет

## Запись текущих настроек в энергонезависимую FLASH-память (SaveSettings)

SaveSettings()

**Назначение:** Функция предназначена для сохранения настроек и данных в энергонезависимую память:

- # Плотность печати
- # Высота штрих-кода
- # Длина импульса на открытие денежного ящика
- # Графический логотип
- # Заголовки чека
- # Программируемые типы оплат

Команда запрещена, если открыт чек.

При сбросе памяти RAM принтер считывает настройки с флэш-памяти.

## Установка высоты штрих-кода (SetBarcodeHeight)

SetBarcodeHeight(int Height)

**Назначение:** Функция предназначена установки высоты штрих-кода в пикселях (0.125 мм), который печатается функцией [PrintBarCode\(\)](#). Данная функция предназначена для использования только с регистратором Экселлио FPU-550.

**Параметры:**

**Height** – целое число, высота штрих-кода. Может принимать значение от 24 (3 мм) до 240 (30 мм).

**Возвращаемое значение:** Нет

## Установка контрастности печати (SetPrintDensity)

SetPrintDensity(int Density)

**Назначение:** Функция предназначена установки контрастности печати.

**Параметры:**

**Density** – целое число, контрастность. Может принимать значения:

1. Сильно бледное
2. Бледное
3. Нормальное
4. Контрастное
5. Сильно контрастное

**Возвращаемое значение:** Нет

## Включение/выключение автоматического открытия денежного ящика (EnableAutoOpenDrawer)

EnableAutoOpenDrawer(VARIANT\_BOOL bEnabled)

**Назначение:** Функция предназначена для разрешения или запрета автоматического открытия денежного ящика по командам [Total\(\)](#) или [InOut\(\)](#).

**Параметры:**

**bEnabled** – логическое значение, включить/выключить открытие ящика. Может принимать значения: True (Истина или цифровое значение -1) – включить автооткрытие, False (Ложь или цифровое значение 0) – выключить.

**Возвращаемое значение:** Нет

## Установка названий дополнительных типов оплат (SetTaxName)

SetTaxName(int Tax, BSTR Name)

**Назначение:** Функция предназначена для установки наименований дополнительных типов оплат. Выбор типа оплат осуществляется с помощью функции [Total\(\)](#).

**Параметры:**

**Tax** – целое число, номер дополнительного типа оплаты. Может принимать значение от 1 до 4.

**Name** – BSTR строка, наименование типа оплат (до 24 символов).

**Возвращаемое значение:** Нет

## Программирование заголовков чека (SetHeaderFooter)

SetHeaderFooter(int iLine, BSTR Text)

**Назначение:** Функция предназначена для установки заголовка чека и рекламной информации в чеке.

**Параметры:**

**iLine** – целое число, номер ряда в чеке. Может принимать значения: от 1 до 6 – для заголовка чека, 7-8 – для рекламной информации в конце чека.

**Text** – BSTR строка, соответствующий текст для строки заголовка или рекламного текста (до 36 символов).

**Возвращаемое значение:** Нет

## Чтение заголовков чека (GetHeader)

GetHeader(int iLine)

**Назначение:** Функция предназначена для чтения заголовка чека и рекламной информации чека.

**Параметры:**

**iLine** – целое число, номер ряда в чеке. Может принимать значения: от 1 до 6 – для заголовка чека, 7-8 – для рекламной информации в конце чека.

**Возвращаемое значение:** Нет

**Выходной параметр:** Прочитанный из фискального регистратора заголовок чека содержится в свойстве **s1**.

## Включение/выключение автообрезки чека (EnableCutCheck)

EnableCutCheck(VARIANT\_BOOL bEnabled)

**Назначение:** Функция предназначена для разрешения или запрета автоматического отрезания чека. Действует до выключения регистратора, после включения по умолчанию параметр устанавливается в соответствии с 3-им выключателем, который находится снизу регистратора.

**Параметры:**

**bEnabled** – логическое значение, включить/выключить автообрезку. Может принимать значения: True (Истина) – включить автообрезку, False (Ложь) – выключить.

**Возвращаемое значение:** Нет

## Включение/выключение уменьшенного шрифта контрольной ленты (EnableSmallFont)

EnableSmallFont(VARIANT\_BOOL bEnabled)

**Назначение:** Функция предназначена для разрешения или запрета использования уменьшенного шрифта на контрольной ленте. Действует до выключения регистратора, после включения по умолчанию параметр устанавливается в соответствии с 7-ым выключателем, который находится внизу регистратора.

**Параметры:**

**bEnabled** – логическое значение, включить/выключить уменьшенный шрифт. Может принимать значения: True (Истина) – уменьшенный шрифт, False (Ложь) – нормальный шрифт.

**Возвращаемое значение:** Нет

## Включение/выключение запрета печати отчетов на контрольной ленте (EnableCRReport)

EnableCRReport(VARIANT\_BOOL bEnabled)

**Назначение:** Функция предназначена для разрешения или запрета печати отчетов (кроме Z-отчета) на контрольной ленте.

**Параметры:**

**bEnabled** – логическое значение, включить/выключить. Может принимать значения: True (Истина) – разрешить печать отчетов на контрольной ленте шрифт, False (Ложь) – запретить.

**Возвращаемое значение:** Нет

## Включение/выключение печати логотипа на чеке (EnableLogo)

EnableLogo(VARIANT\_BOOL bEnabled)

**Назначение:** Функция предназначена для разрешения или запрета печати логотипа завода изготовителя регистратора.

**bEnabled** – логическое значение, включить/выключить печать логотипа. Может принимать значения: True (Истина) – включить печать логотипа, False (Ложь) – отключить печать.

**Возвращаемое значение:** Нет

## Установка даты и времени в фискальном регистраторе (SetDateTime)

SetDateTime(BSTR Date, BSTR Time)

**Назначение:** Функция предназначена для установки даты и времени в фискальном регистраторе. Данная команда не будет выполнена фискальным регистратором, если указана дата, предшествующая дате последнего Z-отчета. Величина года должна быть

до 2089. Изменение даты допускается только после аварийного обнуления ОЗУ фискального регистратора. Изменение времени допускается только после снятия Z-отчета.

**Параметры:**

**Date** – BSTR строка, определяющая дату в формате ДДММГГ

**Time** – BSTR строка, определяющая время в формате ЧЧ:ММ

**Возвращаемое значение:** Нет

## Фискализация фискального регистратора (Fiscalise)

Fiscalise(BSTR Pass, BSTR SerNum, BSTR TaxNum, int iTaxNumType)

**Назначение:** Функция предназначена для фискализации фискального регистратора. Если до выполнения функции фискальный номер не был задан, то необходимо предварительно выполнить функцию [SetTaxNumber](#). После проведения фискализации невозможно фискальный регистратор вновь перевести в учебный режим.

**Параметры:**

**Pass** – BSTR строка, пароль (оператор 14) от 4 до 8 цифр. По умолчанию равен строке «0000».

**SerNum** – BSTR строка, заводской номер фискального регистратора (например: «ИП00004100»).

**TaxNum** – BSTR строка, строка из 12 цифр, содержащая налоговый/идентификационный номер.

**iTaxNumType** – целое число, тип налогового номера. 0 – ПН, 1 – ИД.

**Возвращаемое значение:** Нет

## Установка десятичной точки, налоговых групп и ставок налогов (SetMulDecCurRF)

SetMulDecCurRF(BSTR Pass, int iDec, BSTR TaxEnable, double dTaxA, double dTaxB, double dTaxC, double dTaxD)

**Назначение:** Функция предназначена для установки десятичной точки, используемых налоговых групп и ставок налогов

**Параметры:**

**Pass** – BSTR строка, пароль (оператор 14), строка от 4 до 8 цифр. По умолчанию равен строке «0000».

**iDec** – целое число, место десятичной точки. Для Украины значение равно 2.

**TaxEnable** – BSTR строка, четыре символа, каждый из которых имеет значение 0 – запрещен налог или 1 – разрешен налог. Первый символ соответствует налогу А, второй – Б, третий – В, четвертый Г. Пример: разрешить налоги А, Б, В, Г – «1111».

**dTaxA** – вещественное число, размер налоговой ставки А. Принимает значения от 0 до 99.99

**dTaxB** – вещественное число, размер налоговой ставки Б. Принимает значения от 0 до 99.99

**dTaxC** – вещественное число, размер налоговой ставки В. Принимает значения от 0 до 99.99

**dTaxD** – вещественное число, размер налоговой ставки Г. Принимает значения от 0 до 99.99

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – десятичная запятая;

**s2** – разрешенные налоги (см. TaxEnable);

**s3** – налог А;

**s4** – налог Б;

**s5** – налог В;

**s6** – налог Г.

## Установка режима продаж (SetTaxType)

SetTaxType(int iType)

**Назначение:** Функция предназначена для установки режима продаж. Функция может быть выполнена при изменении или установке налогового номера или после снятия Z-отчета.

**Параметры:**

**iType** – целое число, может принимать значения: 0 – обычный режим, 1 – режим продаж наложенного типа.

**Возвращаемое значение:** Нет

## Установка заводского номера фискального регистратора (SetSerialNum)

SetSerialNum(BSTR SerialNumber)

**Назначение:** Функция предназначена для программирования заводского номера аппарата. Выполняется только после форматирования фискальной памяти.

**Параметры:**

**SerialNumber** - BSTR строка, заводской номер 10 символов, 2 буквы и 8 цифр.

**Возвращаемое значение:** Нет

## Установка фискального номера регистратора (SetFiscalNumber)

SetFiscalNumber(BSTR FN)

**Назначение:** Функция предназначена для программирования фискального номера регистратора.

**Параметры:**

**FN** - BSTR строка, фискальный номер, строка из 10 цифр.

**Возвращаемое значение:** Нет

## Установка налогового/идентификационного номера (SetTaxNumber)

SetTaxNumber(BSTR TaxNumber, int iType)

**Назначение:** Функция предназначена для установки налогового/идентификационного номера. Данная функция используется только для учебного режима фискального регистратора. После назначения номера необходимо использовать функцию **Fiscalise**.

**Параметры:**

**TaxNumber** - BSTR строка, строка из 12 цифр, содержащая налоговый/идентификационный номер.

**iType** – целое число, тип налогового номера. Может принимать значения: 0 – ПН, 1 - ИД

**Возвращаемое значение:** Нет

## Программирование паролей операторов (SetOperatorPassword)

SetOperatorPassword(int iOperNum, BSTR OldPass, BSTR NewPass)

**Назначение:** Функция предназначена для программирования паролей любого из 16 операторов фискального регистратора. Вслед за инициализацией и обнулением оперативной памяти значения паролей операторов равны строке 0000. Указанный пароль используется в команде открытия фискального чека и др. Если при выполнении команды открытия фискального чека были совершены три попытки (подряд) неверного ввода пароля, то фискальный регистратор блокируется и прекращает работу вплоть до выключения и затем включения его вновь.

**Параметры:**

**iOperNum** – целое число, номер оператора. Может принимать значения от 1 до 16

**OldPass** – BSTR строка, старый пароль оператора – от 4 до 8 цифр

**NewPass** – BSTR строка, новый пароль оператора – от 4 до 8 цифр

**Возвращаемое значение:** Нет

## Программирование имен операторов (SetOperatorName)

SetOperatorName(int iOperatorNum, BSTR Password, BSTR Name)

**Назначение:** Функция предназначена для программирования имени любого из 16 операторов фискального регистратора. Вслед за инициализацией и обнулением оперативной памяти значения паролей операторов равны строке 0000. Имя оператора печатается при открытии фискального чека. Если при выполнении этой команды уже был ранее открыт фискальный чек или был указан неверный пароль, то команда не выполняется.

**Параметры:**

iOperatorNum – целое число, номер оператора. Может принимать значения от 1 до 16

Password – BSTR строка, пароль оператора, строка от 4 до 8 цифр

Name – BSTR строка, имя оператора (24 символов включительно).

**Возвращаемое значение:** Нет

## Обнуление сумм по операторам (ClearOperator)

ClearOperator(int iOperNum, BSTR Pass)

**Назначение:** Функция предназначена для обнуления информации о суммах накопленных при продажах для данного оператора. Если при выполнении этой команды уже был ранее открыт фискальный чек или был указан неверный пароль, то команда не выполняется.

**Параметры:**

iOperNum – целое число, номер оператора. Может принимать значения от 1 до 16

Pass – BSTR строка, пароль оператора, строка от 4 до 8 цифр

**Возвращаемое значение:** Нет

## Установка пароля администратора (SetAdminPassword)

SetAdminPassword(BSTR OldPass, BSTR NewPass)

**Назначение:** Функция предназначена для программирования пароля администратора, с помощью которого можно обнулять пароли любого из 16 операторов. По умолчанию запрограммированный пароль администратора – строка «12345678».

**Параметры:**

OldPass – BSTR строка, старый пароль администратора, строка до 8 цифр. По умолчанию равен строке «12345678».

NewPass – BSTR строка, новый пароль администратора, строка до 8 цифр.

**Возвращаемое значение:** Нет

## Удаление пароля оператора (ClearOperatorPassword)

ClearOperatorPassword(int iOperNum, BSTR AdmPass);

**Назначение:** Функция предназначена для обнуления паролей конкретного оператора администратором.

**Параметры:**

iOperNum – целое число, номер оператора. Может принимать значение от 1 до 16.

AdmPass – BSTR строка, пароль администратора, строка из 8 цифр. По умолчанию равен «12345678».

**Возвращаемое значение:** Нет

## Продажа

## Открыть не фискальный чек (OpenNonfiscalReceipt)

### OpenNonfiscalReceipt()

**Назначение:** Функция предназначена для открытия нефискального чека (чека комментариев). Команда не выполняется, если уже открыт нефискальный или фискальный чек.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – счетчик всех нефискальных чеков от последнего Z - отчета;

**s2** – счетчик всех фискальных чеков от последнего Z – отчета;

**s3** – счетчик всех возвратных чеков от последнего Z – отчета.

## Печать не фискального текста (PrintNonfiscalText)

### PrintNonfiscalText(BSTR text)

**Назначение:** Функция предназначена для печати нефискального текста в нефискальном чеке.

**Параметры:**

**text** - BSTR строка, тестовая информация длиной до 1024 символов. В одной строке печатается 34 символа.

**Возвращаемое значение:** Нет

## Закреть не фискальный чек (CloseNonfiscalReceipt)

### CloseNonfiscalReceipt()

**Назначение:** Функция предназначена для закрытия нефискального чека.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – счетчик всех нефискальных чеков от последнего Z - отчета;

**s2** – счетчик всех фискальных чеков от последнего Z – отчета;

**s3** – счетчик всех возвратных чеков от последнего Z – отчета.

## Открыть фискальный чек (OpenFiscalReceipt)

### OpenFiscalReceipt(int iOperator, BSTR Password, int iPlaceNumber)

**Назначение:** Функция предназначена для открытия фискального чека.

Команда не выполняется если:

# уже был открыт фискальный или нефискальный чек

# переполнен счетчик чеков

# фискальная память переполнена

# фискальная память повреждена

# ошибка в параметрах команды

# заголовок чека содержит менее двух рядов

# неверно задан пароль оператора (при троекратном неправильном задании пароля фискальный регистратор блокируется, пока не будет проведено его выключение и затем включение)

# дата и время не установлены

# не подключен индикатор покупателя

**Параметры:**

**iOperator** – целое число, номер оператора. Может принимать значения от 1 до 16.



**Password** - BSTR строка, пароль оператора, номер которого задан **iOperator** – строка от 4 до 6 цифр. По умолчанию равен строке «0000».

**iPlaceNumber** – целое число, номер кассового места. Может принимать значение от 1 до 65535.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – счетчик всех нефискальных чеков от последнего Z - отчета;

**s2** – счетчик всех фискальных чеков от последнего Z – отчета;

**s3** – счетчик всех возвратных чеков от последнего Z – отчета.

## Регистрация продажи (Registrltem)

**Registrltem(int iArtNum, double dQnty, double dPercentDisc, double dSumDisc)**

**Назначение:** Функция предназначена для регистрации продажи товара в фискальном чеке, а также осуществления скидки или надбавки на товар. По данной функции из таблицы артикулов (в фискальном регистраторе) извлекается информация о названии артикула, его цене, налоговой группе. Имя артикула печатается в одной строке с ценой и кодом налоговой группы. Цена товара добавляется к соответствующим регистрам оперативной памяти. Скидка/надбавка печатается в отдельной строке.

**Функция не будет выполнена успешно если:**

- Не открыт фискальный чек
- Нет запрограммированного артикула с указанным в функции номером
- Переполнено количество продаж в одном чеке (500)
- Функция [Total](#) уже была выполнена
- Сумма по какой-либо налоговой группе может стать отрицательной в результате выполнения функции
- Сумма для конкретного артикула в пределах чека может стать отрицательной
- Сумма скидок/надбавок приведет к общей отрицательной сумме по чеку
- Не подключен дисплей покупателя

**Параметры:**

**iArtNum** – целое число, номер артикула запрограммированного в фискальный регистратор. Может принимать значение от 1 до 999999999.

**dQnty** – вещественное число, количество для продажи. Произведение количества на цену не должно превышать 9 знаков (2 после запятой).

**dPercentDisc** – вещественное число, процент скидки или надбавки на товар/услугу в процентном значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Может принимать значения от -99 до 99. Общие суммы по скидкам и надбавкам печатаются в дневном отчете. Если нет необходимости осуществлять скидку/надбавку – значение **dPercentDisc** необходимо передавать равным нулю.

**dSumDisc** – вещественное число, процент скидки или надбавки на товар/услугу в суммовом значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Если нет необходимости осуществлять скидку/надбавку – значение **dSumDisc** необходимо передавать равным нулю.

**Внимание:** при указании **dPercentDisc** и **dSumDisc** одновременно неравными нулю – возникает ошибка "Неверные параметры функции".

**Возвращаемое значение:** Нет

## Регистрация продажи с изменением цены (RegistrltemEx)

RegistrItemEx(int iArtNum, double dQty, double dPrice, double dPercentDisc, double dSumDisc)

**Назначение:** Функция предназначена для регистрации продажи товара в фискальном чеке, а также осуществления скидки или надбавки на товар.

**Внимание:** Основное отличие от функций [RegistrItem](#) и [RegistrAndDisplayItem](#) – возможность указывать новую цену товара без предварительного программирования ее с помощью [ChangeArticlePrice](#).

**Параметры:**

**iArtNum** – целое число, номер артикула запрограммированного в фискальный регистратор. Может принимать значение от 1 до 999999999.

**dQty** – вещественное число, количество для продажи.

**dPrice** – вещественное число, новая цена товара.

**dPercentDisc** – вещественное число, процент скидки или надбавки на товар/услугу в процентном значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Может принимать значения от -99 до 99. Общие суммы по скидкам и надбавкам печатаются в дневном отчете. Если нет необходимости осуществлять скидку/надбавку – значение **dPercentDisc** необходимо передавать равным нулю.

**dSumDisc** – вещественное число, процент скидки или надбавки на товар/услугу в суммовом значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Если нет необходимости осуществлять скидку/надбавку – значение **dSumDisc** необходимо передавать равным нулю.

**Внимание:** при указании **dPercentDisc** и **dSumDisc** одновременно неравными нулю – возникает ошибка "Неверные параметры функции".

**Возвращаемое значение:** Нет

### Регистрация продажи и отображение на индикатор клиента продажи (RegistrAndDisplayItem)

RegistrAndDisplayItem(int iArtNum, double dQty, double dPercentDisc, double dSumDisc)

**Назначение:** Функция предназначена для регистрации и отображения на индикатор клиента продажи товара в фискальном чеке, а также осуществления скидки или надбавки на товар.

**Параметры:** аналогичны функции [RegistrItem](#)

**Возвращаемое значение:** Нет

### Регистрация продажи, изменение цены и отображение на индикатор (RegistrAndDisplayItemEx)

RegistrAndDisplayItemEx(int iArtNum, double dQty, double dPrice, double dPercentDisc, double dSumDisc)

**Назначение:** Функция предназначена для регистрации товара с возможностью изменения цены и отображения на индикатор клиента продажи товара в фискальном чеке, а также осуществления скидки или надбавки на товар. Данная функция предназначена для использования только с регистратором Экселлио FPU-550.

**Параметры:** аналогичны функции [RegistrItemEx](#)

**Возвращаемое значение:** Нет

### Печать фискального текста (PrintFiscalText)

PrintFiscalText(BSTR text)

**Назначение:** Функция предназначена для печати фискального текста в фискальном чеке.

**Параметры:**

`text` - BSTR строка, текстовая информация длиной до 1024 символов. В одной строке чека печатается 34 символа.  
**Возвращаемое значение:** Нет

## Подитог, скидка/надбавка на чек (SubTotal)

SubTotal(double dPercentDisc, double dSumDisc)

**Назначение:** Функция предназначена для выполнения подитога по фискальному чеку, а также для выполнения скидки/надбавки на всю сумму, накопленную в настоящий момент по всему чеку.

**Параметры:**

`dPercentDisc` – вещественное число, процент скидки или надбавки на товар/услугу в процентном значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Может принимать значения от -99 до 99. Общие суммы по скидкам и надбавкам печатаются в дневном отчете. Если нет необходимости осуществлять скидку/надбавку – значение `dPercentDisc` необходимо передавать равным нулю.

`dSumDisc` – вещественное число, скидка или надбавка на товар/услугу в суммовом значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Если нет необходимости осуществлять скидку/надбавку – значение `dSumDisc` необходимо передавать равным нулю.

**Внимание:** при указании `dPercentDisc` и `dSumDisc` одновременно неравными нулю – возникает ошибка "Неверные параметры функции".

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

`s1` – подсумма на текущий момент фискального чека;

`s2` – сумма налогов по налоговой группе А;

`s3` – сумма налогов по налоговой группе Б;

`s4` – сумма налогов по налоговой группе В;

`s5` – сумма налогов по налоговой группе Г;

`s6` – сумма налогов по налоговой группе Д;

## Оплата чека (Total)

Total(BSTR Text, int iPayMode, double dSum)

**Назначение:** Функция предназначена для оплаты текущего фискального чека. Допускается несколько таких команд, например, для использования различных видов оплаты.

**Параметры:**

`Text` – BSTR строка, текстовый комментарий длиной до 48 символов.

`iPayMode` – целое число, тип оплаты. Типы оплат 5-8 используются только с регистратором Экселлио FPU-550.

Может принимать значения:

- 1 – оплата наличными;
- 2 – оплата в кредит;
- 3 – оплата чеком;
- 4 – оплата с помощью платежной карты;
- 5 – Программируемый тип оплаты 1;
- 6 – Программируемый тип оплаты 2;
- 7 – Программируемый тип оплаты 3;
- 8 – Программируемый тип оплаты 4.

**dSum** – вещественное число, сумма, которая вносится клиентом. Если сумма, полученная от клиента, превышает сумму чека, то будет начислена сдача. Может принимать нулевое значение, тогда оплата будет произведена на всю не оплаченную сумму чека указанным типом оплаты.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – результат выполнения функции

**s2** – зависит от предыдущего параметра:

Если **s1** равен строке “F” – ошибка

Если **s1** равен строке “E” – вычисленная сумма отрицательна. Оплата не совершается.

Если **s1** равен строке “D” – сумма оплаты меньше суммы чека (остаток в **s2**)

Если **s1** равен строке “R” – сумма оплаты больше суммы чека (остаток в **s2**)

Если **s1** равен строке “I” – сумма по некоторой налоговой группе отрицательна (текущая подсумма в **s2**).

## Оплата чека и закрытие фискального чека (TotalEx)

TotalEx(BSTR Text, int iPayMode, double dSum)

**Назначение:** Функция предназначена для оплаты текущего фискального чека. Допускается несколько таких команд, например, для использования различных вид оплаты.

**Внимание:** Единственное отличие от функции [Total](#) – при сумме оплаты чека равной или большей суммы чека, чек закрывается автоматически, т.е. нет необходимости в вызове функции [CloseFiscalReceipt](#).

**Параметры:** аналогичны функции [Total](#)

**Возвращаемое значение:** Нет

**Выходные параметры:** аналогичны функции [Total](#)

## Печать штрих-кода (PrintBarcode)

PrintBarcode(int Type, BSTR Text)

**Назначение:** Функция предназначена для печати штрих-кода в чеке. Высота штрих-кода задается функцией [SetBarcodeHeight](#). Штрих-код печатается центрированным.

**Параметры:**

**Type** – целое число, тип штрих-кода. Может принимать значения:

1. EAN8
2. EAN13
3. Code 128
4. ITF (Interleaved 2 of 5).
5. ITF (Interleaved 2 of 5).

**Text** – строка, данные штрих-кода:

1. Для EAN8 – 7 или 8 символов, только цифры. В любом случае контрольная сумма рассчитывается автоматически.
2. Для EAN13 – 12 или 13 символов, только цифры. В любом случае контрольная сумма рассчитывается автоматически.
3. Для Code 128 – от 22 или 42 символов, цифры и буквы. Контрольная сумма рассчитывается автоматически.
4. Для ITF (Interleaved 2 of 5) – только цифры.
5. Для ITF (Interleaved 2 of 5) – только цифры. Контрольная сумма рассчитывается и печатается автоматически.

**Возвращаемое значение:** Нет

**Выходные параметры:** Нет

## Печать разделительной линии (PrintLine)

## PrintLine(int Type)

**Назначение:** Функция предназначена для печати разделительной строки в чеке.

**Параметры:**

Type – целое число, тип строки. Может принимать значения:

1 - 36 символов «←»

2 - 36 символов «⇒»

3 – 18 символов «←» с интервалом

**Возвращаемое значение:** Нет

**Выходные параметры:** Нет

## Закрытие фискального чека (CloseFiscalReceipt)

### CloseFiscalReceipt()

**Назначение:** Функция предназначена для закрытия фискального чека или чека возврата. Команда не выполняется, если фискальный чек не был полностью оплачен с помощью команды [Total](#).

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – счетчик всех нефискальных чеков от последнего Z - отчета;

s2 – счетчик всех фискальных чеков от последнего Z – отчета;

s3 – счетчик всех возвратных чеков от последнего Z – отчета.

## Отмена чека (CancelReceipt)

### CancelReceipt()

**Назначение:** Функция предназначена для аннулирования текущего открытого фискального чека или чека возврата. Функция не выполнится, если чек не открыт или если была сделана полная или частичная оплата по чеку.

**Возвращаемое значение:** Нет

## Открыть возвратный чек (OpenReturnReceipt)

### OpenReturnReceipt(int iOperator, BSTR Pass, int iPlaceNum)

**Назначение:** Функция предназначена для открытия чека возврата.

**Параметры:** идентичны функции [OpenFiscalReceipt](#)

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – счетчик всех нефискальных чеков от последнего Z - отчета;

s2 – счетчик всех фискальных чеков от последнего Z – отчета;

s3 – счетчик всех возвратных чеков от последнего Z – отчета.

## Печать копии чека (MakeReceiptCopy)

### MakeReceiptCopy(int iCount)

**Назначение:** Функция предназначена для печати копии последнего чека.

**Параметры:**

iCount – целое число, количество копий последнего чека. Может принимать значение 1 или 2.

**Возвращаемое значение:** Нет

## Печать нулевого чека (PrintNullCheck)

PrintNullCheck()

**Назначение:** Функция предназначена для печати нулевого чека, для проверки работоспособности фискального регистратора. Функция автоматически выполняет следующие действия:

- Открывает фискальный чек (см. функцию [OpenFiscalReceipt](#))
- Выполняет печать фискального текста «НУЛЬОВИЙ ЧЕК» (см. функцию [PrintFiscalText](#))
- Выполняет подитог с нулевыми параметрами (см. функцию [SubTotal](#))
- Выполняет оплату наличными чека на сумму 0 грн (см. функцию [Total](#))
- Выполняет закрытие фискального чека (см. функцию [CloseFiscalReceipt](#))

Внимание! Функция PrintNullCheck открывает чек с параметрами: iOperator=1, Password="0000", iPlaceNumber=1. (См. подробное описание функции Открытие фискального чека [OpenFiscalReceipt](#)) Если в фискальном регистраторе запрограммированы пароли отличные от принятых по умолчанию – необходимо выполнить самостоятельно указанную выше последовательность, используя пароли, запрограммированные в фискальный регистратор.

**Параметры:** Нет

**Возвращаемое значение:** Нет

## Печать суммовой скидки на группу товара (AbsDiscGrp)

AbsDiscGrp(int Grp, double Dis)

**Назначение:** Функция предназначена для печати абсолютной (суммовой) скидки на группу товара. Группа товара задается при программировании артикула функцией [SetArticle](#).

**Параметры:**

Grp – целое число, группа товара. Может принимать значения от 1 до 99.

Dis – вещественное число, сумма скидки. Если сумма отрицательная – осуществляется надбавка.

**Возвращаемое значение:** Нет

## Печать процентной скидки на группу товара (PerDiscGrp)

PerDiscGrp(int Grp, double Dis)

**Назначение:** Функция предназначена для печати процентной скидки на группу товара. Группа товара задается при программировании артикула функцией [SetArticle](#).

**Параметры:**

Grp – целое число, группа товара. Может принимать значения от 1 до 99.

Dis – вещественное число, процент скидки. Может принимать значения от -99 до 99. Если значение меньше нуля – осуществляется надбавка.

**Возвращаемое значение:** Нет

## Печать суммовой скидки на налоговую группу товара (AbsDiscTax)

AbsDiscTax(int Grp, double Dis)

**Назначение:** Функция предназначена для печати абсолютной (суммовой) скидки на товар с определенной налоговой группой.

Налоговая группа товара задается при программировании артикула функцией [SetArticle](#).

**Параметры:**

**Grp** – целое число, налоговая группа товара. Может принимать значения от 1 до 5.

**Dis** – вещественное число, сумма скидки. Если сумма отрицательная – осуществляется надбавка.

**Возвращаемое значение:** Нет

## Печать процентной скидки на налоговую группу товара (PerDiscTax)

PerDiscTax(int Grp, double Dis)

**Назначение:** Функция предназначена для печати процентной скидки на товар с определенной налоговой группой. Налоговая группа товара задается при программировании артикула функцией [SetArticle](#).

**Параметры:**

**Grp** – целое число, налоговая группа товара. Может принимать значения от 1 до 5.

**Dis** – вещественное число, процент скидки. Может принимать значения от –99 до 99. Если значение меньше нуля – осуществляется надбавка.

**Возвращаемое значение:** Нет

## Регистрация продажи, с автоматическим программированием (Sale)

Sale(int Code, BSTR Name, int Tax, int Grp, double Price, double Qnty, double PercentDisc, double SumDisc, bool Display, BSTR Pass)

**Назначение:** Функция предназначена для регистрации продажи товара в фискальном чеке, а также осуществления скидки или надбавки на товар. Функция совмещает в себе функции [RegistrAndDisplayItemEx](#), [SetArticle](#) и [GetArticle](#). При первой продаже товар автоматически программируется в ФР, а при последующих продажах проверяется соответствие передаваемой информации о товаре, той, которая содержится в ФР. В случае несовпадения одного или нескольких полей (наименование, налоговая группа, группа товара), функция вернет ошибку. Если длина наименования больше той, которую поддерживает ФР, то оставшаяся часть наименования будет распечатана в качестве комментариев.

**Параметры:**

**Code** – целое число, код артикул. Может принимать значение от 1 до 999999999.

**Name** – BSTR строка, наименование артикула. Длина не ограничена.

**Tax** – целое число, налоговая группа. Может принимать значения от 1 до 7 (А ...Д - соответственно).

**Grp** – целое число, группа товара. Может принимать значение от 1 до 99.

**Price** – вещественное число, новая цена товара.

**Qnty** – вещественное число, количество для продажи. **Если количество < 0, то будет произведена аннуляция товара.**

**PercentDisc** – вещественное число, процент скидки или надбавки на товар/услугу в процентном значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Может принимать значения от –99 до 99. Общие суммы по скидкам и надбавкам печатаются в дневном отчете. Если нет необходимости осуществлять скидку/надбавку – значение **PercentDisc** необходимо передавать равным нулю.

**SumDisc** – вещественное число, процент скидки или надбавки на товар/услугу в суммовом значении. Если значение отрицательное – осуществляется скидка, если положительное – надбавка. Если нет необходимости осуществлять скидку/надбавку – значение **SumDisc** необходимо передавать равным нулю.

**Внимание:** при указании **PercentDisc** и **SumDisc** одновременно неравными нулю – возникает ошибка "Неверные параметры функции".

**Display** – Если = True, то будет выведена информация на индикатор, аналогично функции [RegistrAndDisplayItemEx](#).

**Pass** – BSTR строка, пароль для программирования (оператор 14). По умолчанию пароль равен строке «0000».

**Возвращаемое значение:** см. [Коды ошибок](#)

Расширенный код ошибки можно получить в свойстве `LastError`.

## Регистрация продажи, с автоматическим программированием (SaleWC)

`SaleWC(BSTR Code, BSTR Name, int Tax, int Grp, double Price, double Qnty, double PercentDisc, double SumDisc, bool Display, BSTR Pass)`

**Назначение:** Функция аналогична [Sale](#). Отличие заключается в том, что в данном варианте можно передавать строковый код любой длины. Таблица соответствия данных строковых кодов и числовых кодов ФР ведется внутри библиотеки. Для каждого ФР создается отдельная таблица. Хранятся данные таблицы по умолчанию в папке с библиотекой, в файлах с расширением `.art`. Т.е. учетной системе данную таблицу вести не нужно.

## Программирование артикулов и получение информации об артикулах

### Получение информации об артикулах (GetArticlesInfo)

`GetArticlesInfo()`

**Назначение:** Функция предназначена для получения информации из фискального регистратора об общем количестве артикулов, максимальной длине имени артикула, количестве запрограммированных артикулов.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

`s1` – максимальная длина артикула (размещаемая в одной строке);

`s2` – всего доступных для программирования артикулов;

`s3` – запрограммировано артикулов.

### Программирование артикула (SetArticle)

`SetArticle(int iCode, int iTax, int iGrp, double dPrice, BSTR Pass, BSTR Name)`

**Назначение:** Функция предназначена для программирования артикулов в фискальный регистратор. Невозможно запрограммировать артикул, если артикул с таким же значением кода товара `iCode` имеет не нулевой оборот, т.е. уже продавался в течение дня.

**Параметры:**

`iCode` – целое число, номер артикула.

Может принимать значение от 1 до 999999999.

Внимание: Значение номера артикула не связано с максимальным объемом памяти в 28140 артикулов для Экселлио FPU-550 и 59200 для Экселлио FPU-260, это разные понятия.

`iTax` – целое число, налоговая группа. Может принимать значения от 1 до 7.

`iGrp` – целое число, группа товара. Может принимать значение от 1 до 99.

`dPrice` – вещественное число, цена за единицу товара. Может принимать значение от 0.01 до 9999999.99

`Pass` – BSTR строка, пароль для программирования (оператор 14). По умолчанию пароль равен строке «0000».

`Name` – BSTR строка, наименование артикула длиной до 36 символов.

**Возвращаемое значение:** Нет

### Удаление артикула/артикулов (DelArticle)

`DelArticle(BSTR Pass, int iCode)`



**Назначение:** Функция предназначена для удаления артикула с указанным номером или всех артикулов с нулевыми оборотами по продаже. Функция выполняется успешно только в том случае, если оборот по продажам, данного артикула, равен нулю.

**Параметры:**

**Pass** – BSTR строка, пароль для программирования (оператор 14). По умолчанию равен строке «0000».

**iCode** – целое число, номер артикула. Может принимать значение от 1 до 999999999. Если параметр **iCode** равен 0, удаляются все артикулы с нулевыми оборотами по продаже.

**Возвращаемое значение:** Нет

## Чтение артикула (GetArticle)

### GetArticle(int iCode)

**Назначение:** Функция предназначена для чтения информации об артикуле, запрограммированном в фискальный регистратор.

**Параметры:**

**iCode** – целое число, номер артикула. Может принимать значение от 1 до 999999999, но не более чем в ответе функции

[GetArticlesInfo](#).

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1** – результат выполнения функции;
- s2** – номер артикула (код);
- s3** – налоговая группа;
- s4** – товарная группа;
- s5** – цена артикула;
- s6** – проданное количество;
- s7** – стоимость проданного товара;
- s8** – проданное количество в пределах чека;
- s9** – стоимость проданного товара в пределах чека;
- s10** – наименование артикула.

## Чтение первого артикула (GetFirstArticle)

### GetFirstArticle()

**Назначение:** Функция предназначена для чтения информации об первом запрограммированном артикуле. Возвращает информацию такого же типа, как и [GetArticle](#). Способ использования – первой выполняется функция [GetFirstArticle](#), а затем выполняется многократно функция [GetNextArticle](#) до возвращения ошибки выполнения.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1** – результат выполнения функции;
- s2** – номер артикула (код);
- s3** – налоговая группа;
- s4** – товарная группа;
- s5** – цена артикула;
- s6** – проданное количество;
- s7** – стоимость проданного товара;
- s8** – проданное количество в пределах чека;
- s9** – стоимость проданного товара в пределах чека;
- s10** – наименование артикула.

## Чтение следующего артикула (GetNextArticle)

## GetNextArticle()

**Назначение:** Функция предназначена для чтения информации о следующем запрограммированном артикуле. Возвращает информацию такого же типа как и [GetArticle](#).

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1 – результат выполнения функции;
- s2 – номер артикула (код);
- s3 – налоговая группа;
- s4 – товарная группа;
- s5 – цена артикула;
- s6 – проданное количество;
- s7 – стоимость проданного товара;
- s8 – проданное количество в пределах чека;
- s9 – стоимость проданного товара в пределах чека;
- s10 – наименование артикула.

## Изменение цены артикула (ChangeArticlePrice)

### ChangeArticlePrice(BSTR Pass, int iCode, double dPrice)

**Назначение:** Функция предназначена для изменения цены запрограммированного артикула. Изменение цены невозможно, если уже открыт фискальный чек. Допустимо изменение цены в открытом фискальном чеке.

**Параметры:**

Pass – BSTR строка, пароль для программирования (оператор 14). По умолчанию пароль равен строке «0000».

iCode – целое число, номер артикула. Может принимать значение от 1 до 999999999.

dPrice – вещественное число, новая цена за единицу товара.

**Возвращаемое значение:** Нет

## Получение информации о первом свободном артикуле (GetFirstFreeArticle)

### GetFirstFreeArticle()

**Назначение:** Функция предназначена для получения первого свободного (незапрограммированного) артикула.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах.

- s1 – результат выполнения функции;
- s2 – номер артикула (код);

## Получение информации о последнем свободном артикуле (GetLastFreeArticle)

### GetLastFreeArticle()

**Назначение:** Функция предназначена для получения последнего свободного (незапрограммированного) артикула.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах.

- s1 – результат выполнения функции;
- s2 – номер артикула (код);

## Отчеты

### Дневной отчет без обнуления, X - отчет (XReport)

XReport(BSTR Pass)

**Назначение:** Функция предназначена для печати дневного отчета фискального регистратора без обнуления (X - отчет). В отчете также выводится информация о служебном вносе/выносе денег.

**Параметры:**

Pass – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

**Возвращаемое значение:** Нет

### Дневной отчет с обнулением, Z - отчет (ZReport)

ZReport(BSTR Pass)

**Назначение:** Функция предназначена для печати фискального отчета (Z - отчета) с обнулением оперативной памяти и занесением сумм в фискальную память. В отчете также выводится информация о служебном вводе/выводе денег.

**Параметры:**

Pass – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

**Возвращаемое значение:** Нет

### Отчет налоговых ставок за период (PrintTaxReport)

PrintTaxReport(BSTR Pass, BSTR BegDate, BSTR EndDate)

**Назначение:** Функция предназначена для печати отчета налоговых ставок за период

**Параметры:**

Pass – BSTR строка, пароль для отчета (оператор 15). По умолчанию пароль равен строке «0000».

BegDate – BSTR строка, начальная дата отчета в формате ДДММГГ.

EndDate – BSTR строка, конечная дата отчета в формате ДДММГГ.

**Возвращаемое значение:** Нет

### Полный периодический отчет фискальной памяти по номерам Z - отчетов (PrintRepByNumFull)

PrintRepByNumFull(BSTR Pass, int iBegNum, int iEndNum)

**Назначение:** Функция предназначена для печати полного периодического отчета из фискальной памяти по номерам Z - отчетов.

**Параметры:**

Pass – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

iBegNum – целое число, начальный номер Z - отчета.

iEndNum – целое число, конечный номер Z - отчета.

**Возвращаемое значение:** Нет

### Полный периодический отчет фискальной памяти по датам Z - отчетов (PrintRepByDateFull)

PrintRepByDateFull(BSTR Pass, BSTR BegDate, BSTR EndDate)

**Назначение:** Функция предназначена для печати полного периодического отчета из фискальной памяти по номерам диапазону дат.

**Параметры:**

**Pass** – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

**BegDate** – BSTR строка, конечная дата Z отчета, в формате ДДММГГ.

**EndDate** – BSTR строка, конечная дата Z отчета, в формате ДДММГГ.

**Возвращаемое значение:** Нет

### Краткий периодический отчет фискальной памяти по номерам Z - отчетов (PrintRepByNum)

PrintRepByNum(BSTR Pass, int iBegNum, int iEndNum)

**Назначение:** Функция предназначена для печати сокращенного периодического отчета из фискальной памяти по номерам Z - отчетов.

**Параметры:**

**Pass** – BSTR строка, пароль для отчета (оператор 15). По умолчанию пароль равен строке «0000».

**iBegNum** – целое число, начальный номер Z - отчета.

**iEndNum** – целое число, конечный номер Z - отчета.

**Возвращаемое значение:** Нет

### Краткий периодический отчет фискальной памяти по датам Z - отчетов (PrintRepByDate)

PrintRepByDate(BSTR Pass, BSTR BegDate, BSTR EndDate)

**Назначение:** Функция предназначена для печати сокращенного периодического отчета из фискальной памяти по номерам диапазону дат.

**Параметры:**

**Pass** – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

**BegDate** – BSTR строка, конечная дата Z отчета, в формате ДДММГГ.

**EndDate** – BSTR строка, конечная дата Z отчета, в формате ДДММГГ.

**Возвращаемое значение:** Нет

### Отчет по артикулам(PrintRepByArt)

PrintRepByArt(BSTR Pass, int iType)

**Назначение:** Функция предназначена для печати различных отчетов по артикулам.

**Параметры:**

**Pass** – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

**iType** – целое число, тип отчета. Может принимать значения:

1 – отчет реализованных товаров

2 – отчет запрограммированных товаров

3 – отчет по товарным группам

**Возвращаемое значение:** Нет

### Отчет по операторам(PrintRepByOperator)

PrintRepByOperator(BSTR Pass)

**Назначение:** Функция предназначена для печати отчета по операторам. В отчете распечатывается служебная информация по операторам: продажи, скидки, надбавки, коррекции.

**Параметры:**

**Pass** – BSTR строка, пароль для отчета (оператор 15). По умолчанию равен строке «0000».

**Возвращаемое значение:** Нет

## Дневной отчет с обнулением, Z - отчет (ZReportWC)

ZReportWC(BSTR Pass)

**Назначение:** Функция аналогична [ZReport](#). Отличие заключается в том, что в данном варианте после распечатки отчета очищается таблица внутренних артикулов ФР, а также таблица соответствия строковых кодов и числовых кодов ФР.

См. также [SaleWC](#).

## Передача информации в прикладную программу

### Получить дату и время (GetDateTime)

GetDateTime()

**Назначение:** Функция предназначена для чтения даты и времени из фискального регистратора.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – дата и время в формате DD-MM-YY HH:MM:SS;

### Получить информации о суммах фискальных или возвратных чеков (LastFiscalClosure)

LastFiscalClosure(int iParam)

**Назначение:** С помощью этой функции фискальный регистратор возвращает информацию о последнем Z-отчете. Данные можно получить как о суммах по чекам продаж, так и о суммах по чекам возврата.

**Параметры:**

**iParam** – целое число, параметр функции. Может принимать значения: 0 – функция возвращает информацию о суммах фискальных чеков, 1 – функция возвращает информацию о суммах возвратных чеков.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – номер последнего дневного отчета;

**s2** – Сумма по налоговой группе А;

**s3** – Сумма по налоговой группе Б;

**s4** – Сумма по налоговой группе В;

**s5** – Сумма по налоговой группе Г;

**s6** – Сумма по налоговой группе Д.

## Получить суммы с момента последнего Z - отчета (GetCurrentSums)

## GetCurrentSums(int iParam)

**Назначение:** С помощью этой функции фискальный регистратор печатает суммы от последнего дневного отчета до текущего момента времени.

### Параметры:

**iParam** – целое число, параметр функции.

Может принимать значения:

0 – функция возвращает информацию о продажах за день (от последнего Z – отчета)

1 – функция возвращает информацию о возвратах за день (от последнего Z – отчета)

2 – функция возвращает информацию о налогах по продажам за день (от последнего Z – отчета)

3 – функция возвращает информацию о налогах по возвратам за день (от последнего Z – отчета)

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – Сумма по налоговой группе А;

**s2** – Сумма по налоговой группе Б;

**s3** – Сумма по налоговой группе В;

**s4** – Сумма по налоговой группе Г;

**s5** – Сумма по налоговой группе Д.

## Получить суммы коррекций за день (GetCorectSums)

## GetCorectSums()

**Назначение:** С помощью этой функции вычисляются и возвращаются фискальным регистратором суммы коррекций, начитанные от последнего Z – отчета до момента выполнения функции.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – Сумма всех продаж;

**s2** – Сумма по внутричековым коррекциям;

**s3** – Сумма оплат по кредиту;

**s4** – Счетчик всех нефискальных чеков от последнего Z - отчета;

**s5** – Счетчик всех фискальных чеков от последнего Z - отчета;

**s6** – Счетчик всех возвратных чеков от последнего Z - отчета;

## Получить размер фискальной памяти (GetFreeClosures)

## GetFreeClosures()

**Назначение:** С помощью этой функции можно получить размер свободной фискальной памяти регистратора.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – размер свободной фискальной памяти;

**s2** – повторяет предыдущий параметр;

## Получить статус фискального регистратора (GetStatus)

## GetStatus(VARIANT\_BOOL bWait)

**Назначение:** Функция предназначена для чтения статуса фискального регистратора. Подробнее см. «Протокол обмена фискального регистратора Экселлио».

**Параметры:**

**BWait** – логический тип,

принимает значения:

True (Истина) – фискальный принтер вначале допечатывает содержимое буфера печати,

False (Ложь) – фискальный принтер не допечатывает буфер печати перед возвратом результата.

**Возвращаемое значение:** Нет

## Получить состояние фискальной транзакции (GetFiscalClosureStatus)

GetFiscalClosureStatus(VARIANT\_BOOL bCurrent)

**Назначение:** Функция возвращает состояние фискальной транзакции фискального регистратора.

**Параметры:**

**bCurrent** – логический тип,

принимает значения:

True (Истина) – возвращается информация о суммах оплат в последнем чеке

False (Ложь) – не возвращается информация о суммах оплат в последнем чеке (свойство s4)

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – 1 – открыт фискальный или нефискальный чек, 0 – нет открытого чека;

**s2** – количество артикулов проданных в чеке;

**s3** – сумма по чеку;

**s4** – сумма, которую заплатил клиент в последнем чеке.

## Получить диагностическую информацию (GetDiagnosticInfo)

GetDiagnosticInfo(VARIANT\_BOOL bCalcCRC)

**Назначение:** Функция возвращает диагностическую информацию фискального регистратора.

**Параметры:**

**bCalcCRC** – логический тип,

принимает значения:

True (Истина) – вычислять контрольную сумму фискальной памяти

False (Ложь) – не вычислять контрольную сумму фискальной памяти

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – модель ФР (LP1000, FPU550)

**s2** – версия программного обеспечения;

**s3** – контрольная сумма программного обеспечения;

**s4** – состояние переключателей фискального регистратора (0 – выкл., 1 – вкл.);

**s5** – код страны;

**s6** – заводской номер фискального регистратора;

**s7** – фискальный номер.

## Получить налоговые ставки регистратора (GetCurrentTaxRates)

GetCurrentTaxRates()

**Назначение:** Функция возвращает текущие налоговые ставки фискального регистратора.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1 – ставка налоговой группы А;
- s2 – ставка налоговой группы Б;
- s3 – ставка налоговой группы В;
- s4 – ставка налоговой группы Г.

## Получить налоговый номер регистратора (GetTaxNumber)

### GetTaxNumber()

**Назначение:** Функция возвращает налоговый/идентификационный номер и тип номера (ПН или ИД).

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1 – налоговый номер;
- s2 – тип номера (0 – ПН, 1 – ИД).

## Получить информацию о чеке (GetReceiptInfo)

### GetReceiptInfo()

**Назначение:** Функция возвращает информацию о чеке.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1 – 0 - продажа с отрицательным знаком невозможна, 1 – возможна. Фактически означает была ли сделана хоть одна оплата по данному чеку;
- s2 – общая сумма по налоговой группе А;
- s3 – общая сумма по налоговой группе Б;
- s4 – общая сумма по налоговой группе В;
- s5 – общая сумма по налоговой группе Г;
- s6 – общая сумма по налоговой группе Д;
- s7 – 1 – открыт расширенный чек.

## Получить информацию о дневных суммах (GetDayInfo)

### GetDayInfo()

**Назначение:** Функция возвращает информацию о распределении сумм за день по способам оплаты.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

- s1 – Счетчик наличных платежей;
- s2 – Счетчик оплаты в кредит;
- s3 – Счетчик оплаты по дебетной карте;
- s4 – Счетчик оплаты по чеку;
- s5 – Номер последнего дневного Z - отчета;
- s6 – Номер следующего фискального чека от последнего Z - отчета;
- s7 – Глобальный счетчик чеков.
- s8 – Программируемый тип оплаты 1
- s9 – Программируемый тип оплаты 2
- s10 – Программируемый тип оплаты 3
- s11 – Программируемый тип оплаты 4



## Получить суммы накопленные по оператору (GetOperatorInfo)

GetOperatorInfo(int iOperNum)

**Назначение:** Функция возвращает информацию о суммах накопленных для указанного оператора.

**Параметры:**

iOperNum – целое число, номер оператора от 1 до 16.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – Номер оператора (от 1 до 16);

s2 – Счетчик чеков;

s3 – Количество продаж и сумма продаж;

s4 – Количество скидков и сумма скидков;

s5 – Количество надбавок и сумма надбавок;

s6 – Количество чеков возврата и сумма чеков возврата;

s7 – Имя оператора.

## Получить номер последнего чека (GetLastReceiptNum)

GetLastReceiptNum()

**Назначение:** Функция возвращает номер последнего закрытого чека

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – Номер последнего закрытого чека.

## Получить информацию о фискальности/нефискальности регистратора (isFiscalised)

int isFiscalised()

**Назначение:** Функция читает информацию из фискального регистратора о том зафискализирован он или нет.

**Возвращаемое значение:** возвращаемое значение целого типа, может принимать значения:

1 – фискальный регистратор фискализирован

0 – фискальный регистратор не фискализирован

## Получить продолжительность смены (GetSmenLen)

GetSmenLen()

**Назначение:** Функция читает продолжительность смены.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – Может принимать значения:

Z – нет чеков за день

P – продолжительность смены менее 24 часов

F – продолжительность смены более 24 часов

s2 – продолжительность смены в минутах (до 1440 мин)

## Получить дату последнего Z-отчета (GetLastClosureDate)

## GetLastClosureDate()

**Назначение:** Функция предназначена для чтения даты последнего Z-отчета.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – дата последнего Z-отчета в формате ДД-ММ-ГГГГ

## Получить сумму в денежном ящике (GetSumInCash)

### GetSumInCash()

**Назначение:** Функция предназначена для чтения суммы денег в денежном ящике.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – сумма наличных в денежном ящике.

## Получить время последнего обмена с налоговой (GetLastDPAExchangeTime)

### GetLastDPAExchangeTime()

**Назначение:** Функция предназначена для чтения времени последнего обмена с налоговой и состояния модема.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s2** – дата и время последнего успешного обмена с налоговой

**s6** – код ошибки/состояния:

**OK** - Модем работает в штатном режиме.

**MAX\_72H** - Нет связи с сервером 72 часа.

**ERR\_CON** - Последний сеанс обмена был неуспешным.

**MAX\_EXC** - Получены три и более телеграмм исключительных состояний.

## Получить время последнего обмена с налоговой (GetLastDPAExchangeTimeEx)

### GetLastDPAExchangeTimeEx(string sPortNumber, int iBaudRate)

**Назначение:** Функция предназначена для чтения времени последнего обмена с налоговой и состояния модема, в ситуации когда ФР уже заблокировался из-за превышения периода в 72 часа, и не отвечает на другие команды. Для вызова данной функции не нужно предварительно вызывать функцию **OpenPort**, а также **ClosePort** после нее.

**Параметры:** Аналогичны функции [OpenPort\(\)](#).

**Возвращаемое значение:** Нет

**Выходные параметры:** Аналогичны функции [GetLastDPAExchangeTime\(\)](#).

## Получения значений битов состояния ФР (GetStatusBit)

### GetStatusBit(byte ByteNum, byte BitNum)

**Назначение:** Функция возвращает значение бита состояния фискального регистратора, по его номеру.

**Параметры:**

**ByteNum** – номер байта состояния, согласно протокола.

**BitNum** – номер бита, в указанном выше байте.

**Возвращаемое значение:** Целое число, значение бита 1 или 0.

## Получить информацию о модели фискального регистратора (FPModel)

string FPModel

**Назначение:** Свойство возвращает строковый идентификатор модели фискального регистратора. Доступно после вызова [OpenPort](#).

**Возвращаемое значение:** возможные значения FPU550, LP1000.

## Команды принтера

### Прогон ленты фискальный принтером (AdvancePaper)

AdvancePaper(int Lines)

**Назначение:** Функция предназначена для прогона ленты принтером фискального регистратора.

**Назначение:**

Lines – целое число, количество строк для прогона ленты. Может принимать значение от 1 до 99.

**Возвращаемое значение:** Нет

### Прогон ленты фискальный принтером расширенный(AdvancePaperEx)

AdvancePaperEx(int Lines, int Type)

**Назначение:** Функция предназначена для прогона ленты принтером фискального регистратора.

**Назначение:**

iLines – целое число, количество строк для прогона ленты. Может принимать значение от 1 до 99.

Type – целое число, определяет тип ленты. Может принимать значение от 0 до 3.

0. Нет эффекта

1. Продвижение чековой ленты

2. Продвижение контрольной ленты

3. Продвижение чековой и контрольной ленты.

**Возвращаемое значение:** Нет

## Отрезать чек (CutReceipt)

CutReceipt()

**Назначение:** Функция предназначена для отрезки чека.

**Возвращаемое значение:** Нет

## Дисплей

### Очистка дисплея (ClearDisplay)

ClearDisplay()

**Назначение:** Функция предназначена для очистки дисплея.

**Возвращаемое значение:** Нет

## Вывод текста в нижний ряд (DisplayTextLL)

DisplayTextLL(BSTR text)

**Назначение:** Команда предназначена для вывода информации в нижний ряд дисплея.

**Параметры:**

text – BSTR строка, текстовая информация длиной до 20 символов.

**Возвращаемое значение:** Нет

## Вывод текста в верхний ряд (DisplayTextUL)

DisplayTextUL(BSTR text)

**Назначение:** Функция предназначена для вывода информации в верхний ряд дисплея.

**Параметры:**

text – BSTR строка, текстовая информация длиной до 20 символов.

**Возвращаемое значение:** Нет

## Вывод даты и времени (DisplayDateTime)

DisplayDateTime()

**Назначение:** Функция предназначена для вывода даты и времени, установленных в регистраторе, на дисплей.

**Возвращаемое значение:** Нет

## Непосредственный вывод на дисплей (DisplayFreeText)

DisplayFreeText(BSTR text)

**Назначение:** Функция предназначена для непосредственного вывода на дисплей.

**Параметры:**

text – BSTR строка, текстовая информация длиной до 40 символов.

**Возвращаемое значение:** Нет

## Отладка

### Включить отображение ошибок OLE-сервером (ShowError)

ShowError(VARIANT\_BOOL bShow)

**Назначение:** Функция предназначена для отображения ошибок, возникающих в процессе работы, самим OLE-сервером.

**Параметры:**

bShow – логический тип, принимает значения:

True (Истина) – отображать ошибки в виде окна сообщений

False (Ложь) – не отображать ошибки

**Возвращаемое значение:** Нет

## Включить запись отладочной информации в текстовый лог-файл (Debugger)

Debugger(BOOL bDebug)

**Назначение:** Функция предназначена для ведения текстового файла отладки OLE-сервера. Рекомендуется вызывать эту функцию самой первой при работе с OLE – сервером (даже перед [OpenPort](#)).

**Параметры:**

bDebug - логический тип, если равен True – включается отладка OLE-сервера, если False – отключается отладка.

**Возвращаемое значение:** Нет

## Другие

### Открыть денежный ящик (OpenDrawer)

OpenDrawer()

**Назначение:** Функция предназначена для открытия денежного ящика.

**Возвращаемое значение:** Нет

### Открыть денежный ящик с регулируемой длиной импульса (OpenDrawerEx)

OpenDrawerEx(int iMsc)

**Назначение:** Функция предназначена для открытия денежного ящика. Функция автоматически вызывается после закрытия чека, служебного внесения/вынесения денег, X-отчета, Z-отчета.

**Параметры:**

iMsc – целое число, длина импульса в миллисекундах (5 - 150).

**Возвращаемое значение:** Нет

### Служебный ввод/вывод денег (InOut)

InOut(double dSum)

**Назначение:** Функция предназначена для служебного ввода/вывода денег. (Например: внос разменной монеты, служебный вынос при инкассации). Команда не выполняется, если сумма служебного вывода денег превышает сумму выторга с момента последнего Z отчета вместе с суммой служебного внесения.

**Функция не выполняется:**

# При выплате суммы большей, чем денежная наличность в кассе,

# При вызове функции в незавершенном фискальном или служебном чеке.

**Параметры:**

DSum – вещественное число, если сумма положительная – выполняется служебное внесение денег, если отрицательная – вынесение.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – Наличность в фискальном регистраторе;

s2 – Общая сумма всех служебно внесенных сумм в фискальный регистратор за день;

s3 – Общая сумма всех служебно выплаченных сумм за день;

## Печать диагностической информации (PrintDiagnosticInfo)

PrintDiagnosticInfo()

**Назначение:** Функция предназначена для печати диагностической информации фискального регистратора.

**Возвращаемое значение:** Нет

## Звуковой сигнал (Sound)

Sound()

**Назначение:** Функция предназначена для вывода звукового сигнала фискальным регистратором.

**Возвращаемое значение:** Нет

## Звуковой сигнал произвольный (SoundEx)

SoundEx(int Hz, int Ms)

**Назначение:** Функция предназначена для вывода звукового сигнала фискальным регистратором с произвольной частотой и длительностью.

**Параметры:**

**Hz** – целое число, частота звука в Гц. Может принимать значение от 100 до 5000.

**Ms** – целое число, продолжительность звукового сигнала в миллисекундах. Может принимать значение от 50 до 2000.

**Возвращаемое значение:** Нет

## КСЕФ

### Прочитать чеки из КСЕФ по номерам (ReadKSEFByNum)

ReadKSEFByNum(int RecNum1, int RecNum2)

**Назначение:** Функция запускает цикл чтения содержимого чеков в текстовом виде, по указанному диапазону номеров. При этом возвращает первую строку чека с номером RecNum1. Для чтения всех последующих строк используется функция

[ReadNextKSEFRec](#).

**Параметры:**

**RecNum1, RecNum2** – номера чеков для чтения.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – результат выполнения операции: **P** - данные есть. **F** - данных больше нет, можно прерывать цикл чтения.

**s2** – первая строка чека с номером RecNum1.

### Чтение следующей строки КСЕФ (ReadNextKSEFRec)

ReadNextKSEFRec()

**Назначение:** Функция возвращает следующую строку из КСЕФ. Вызывается в цикле после [ReadKSEFByNum](#).

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – результат выполнения операции: **P** - данные есть. **F** - данных больше нет, можно прерывать цикл чтения.

**s2** – следующая строка чека из КСЕФ.

## Прочитать чеки из КСЭФ по датам (ReadKSEFByDate)

ReadKSEFByDate(string Date1, string Date2)

**Назначение:** Функция запускает цикл чтения содержимого чеков в текстовом виде, по указанному диапазону дат. При этом возвращает первую строку первого чека на дату Date1. Для чтения всех последующих строк используется функция [ReadNextKSEFRec](#).

**Параметры:**

Date1, Date2 – диапазон дат для чтения чеков.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

s1 – результат выполнения операции: **P** - данные есть. **F** - данных больше нет, можно прерывать цикл чтения.

s2 – первая строка чека.

## Распечатать чеки из КСЭФ по номерам (PrintKSEFByNum)

PrintKSEFByNum(int RecNum1, int RecNum2)

**Назначение:** Функция распечатывает чеки из КСЭФ на ленте, по указанному диапазону номеров.

**Параметры:**

RecNum1, RecNum2 – номера чеков для печати.

**Возвращаемое значение:** Нет

## Распечатать чеки из КСЭФ по датам (PrintKSEFByDate)

PrintKSEFByDate(string Date1, string Date2)

**Назначение:** Функция распечатывает чеки из КСЭФ на ленте, по указанному диапазону дат.

**Параметры:**

Date1, Date2 – диапазон дат для печати.

**Возвращаемое значение:** Нет

## Прочитать чек в формате XML из КСЭФ (ReadKSEFByNumXML)

ReadKSEFByNumXML(int DocType, int DocNum)

**Назначение:** Функция запускает цикл чтения содержимого чеков в XML-виде, по указанному номеру документа. При этом возвращает первую строку чека. Для чтения всех последующих строк используется функция [ReadNextKSEFRec](#).

**Параметры:**

DocType – тип документа для чтения. 0 - чек по его номеру, 1 - Z-отчет по его номеру

DocNum – номер документа для чтения.

**Возвращаемое значение:** Нет

**Выходные параметры:** Прочитанные из фискального регистратора с помощью этой функции данные будут записаны в свойствах:

**s1** – результат выполнения операции: **P** - данные есть. **F** - данных больше нет, можно прерывать цикл чтения.  
**s2** – первая строка чека.