

Специфікація рішень API ПРРО Cashalot

Актуально починаючи з версії ПРРО Cashalot 1.00.35

Зміст

Поняття і визначення	5
Апаратні та програмні вимоги	5
Додаткові пристрої	6
<u>ВАРІАНТ 1 – 1С BAS ДРАЙВЕР ПРРО CASHALOT, CASHALOTAPIDRV32.DLL АБО CASHALOTAPIDRV64.DLL</u>	6
Ініціалізація драйверу:	7
Отримання налаштувань драйверу	9
Встановлення налаштувань драйверу	9
Підключення фіскального реєстратора ПРРО Cashalot	10
Відключення екземпляра фіскального реєстратора ПРРО Cashalot	11
Перевірка ПРРО Cashalot	11
Відкриття зміни	12
Реєстрація чеків	12
Відміна відкритого чеку	18
Отримання інформації про поточний стан ПРРО	18
Друк довільної текстової інформації	19
Повернення товару	20
Службове внесення	21
Службова видача	21
Формування X-звіту	21
Формування Z-звіту	22
Закриття зміни	22
Порядок виклику функцій при виконанні касових операцій	22
Приклад реалізації підключення драйверу до Розрахунково-облікової платформи (.erf)	23
<u>ВАРІАНТ 2А – API ПРРО CASHALOT З СОМ ІНТЕРФЕЙСОМ CASHALOTAPI32.DLL, CASHALOTAPI64.DLL</u>	25
Встановлення налаштувань API бібліотеки SetParameter	27
Встановлення налаштувань API бібліотеки SetParameterEx	29
GetVersion – Отримання номеру версії API бібліотеки	32
GetInterfaceRevision – Отримання номеру ревізії API бібліотеки	32
OpenShift – Відкриття зміни	32
FiscalizeCheck – Фіскалізація чека продажу товарів, послуг	33
FiscalizeReturnCheck – Фіскалізація чека повернення товарів, послуг	41
GetReceiptLocalNumberByFiscalNumber – Отримання локального номеру чеку за його фіскальним номером	44
GetReceiptXML – Отримання XML структури фіскального чеку	45
GetFirstReceiptByPeriod – Отримати перший чек з періоду	47
GetFirstReceiptByShift – Отримати перший чек з зміни	48

GetNextReceipt– Отримати наступний чек з періоду	49
GetCurrentStatus – Отримати інформацію по поточному стану каси	51
GetFirstZReportByPeriod – Отримати перший Z-звіт за період	53
GetNextZReport – Отримати наступний Z-звіт з періоду	54
GetZReportByShift– Отримати Z-звіт касової зміни	56
ShowReceipt – Вивести візуальне відображення чеку на екран	58
PrintTextDocument – Надрукувати довільний нефіскальний текст	59
ServiceInput – Службове внесення грошей в касу	60
ServiceInputEx – Службове внесення грошей в касу	60
ServiceOutput – Службова видача грошей з каси	61
ServiceOutputEx – Службова видача грошей з каси	62
GetLastError – Отримання опису останньої помилки	63
PrintXReport - Формування X-звіту	63
GetXReport - Формування X-звіту	63
GetLastZReport – Отримати останній Z-звіт	67
PrintZReport – Формування Z-звіту	71
CloseShift - Закриття зміни	71
Приклад роботи з кількома ПРРО одночасно в одній обліковій системі використовується новий метод SetParameterEx	75
Можна задавати параметри, які ПРРО використовувати для кожного методу окремо.	76
Ломбардні операції	77
Рекомендований порядок роботи з CashalotApi	81
ВСТАНОВЛЕННЯ ПАРАМЕТРІВ	81
ВІДКРИТТЯ ЗМІНИ	81
ПРОДАЖ ТОВАРІВ, ПОСЛУГ	81
ПОВЕРНЕННЯ ТОВАРІВ, РЕКОМПЕНСАЦІЯ ПОСЛУГ	81
ПОШУК ФІСКАЛЬНОГО ТА ЛОКАЛЬНОГО НОМЕРУ ЧЕКУ	82
ОТРИМАННЯ XML-СТРУКТУРИ ЧЕКУ	82
ВІДОБРАЖЕННЯ ЗАРЕЄСТРОВАНОВОГО ЧЕКУ	82
НАЛАШТУВАННЯ ЕЛЕКТРОННОЇ ПОШТИ	82
СЛУЖБОВЕ ВНЕСЕННЯ ГРОШЕЙ В КАСУ	82
СЛУЖБОВА ВИДАЧА ГРОШЕЙ З КАСИ	82
ФОРМУВАННЯ X-ЗВІТУ	82
ОТРИМАННЯ ЧЕКІВ	82
ОТРИМАННЯ ДОКУМЕНТІВ Z-ЗВІТІВ ЗАКРИТИХ КАСОВИХ ЗМІН	83
ОТРИМАННЯ ОСТАННЬОГО Z-ЗВІТУ	83
ЗАКРИТТЯ ЗМІНИ, Z-ЗВІТ	83
Приклад реалізації підключення SOM бібліотеки на C++	84
СТВОРЕННЯ ФАЙЛІВ ЗАГОЛОВКІВ *.h	84
ЗАУВАЖЕННЯ	84
ЛІСТИНГ	84
ВАРІАНТ 2Б – АРІ ПРРО CASHALOT З .NET ІНТЕРФЕЙСОМ CASHALOTAPIBRIDGE.DLL	102
Специфікація методів АРІ бібліотеки ПРРО Cashalot при підключенні до облікової системи	102
Порядок роботи обліково-розрахункової системи з АРІ бібліотекою ПРРО Cashalot	102
Специфікація методів бібліотеки CashalotApiBridge.dll	103

Відкриття зміни	103
Відкриття зміни	103
Службове внесення грошових коштів в касу	104
Службове вилучення (видача) грошових коштів з каси	105
Проведення розрахункових операцій продажу товарів/послуг	106
Визначення фіскального номеру чеку	113
Проведення операції повернення товару	114
Створення X-Звіту	116
Визначення локального номеру чека по фіскальному номеру	117
Отримання XML-структури чеку	118
Відображення фіскалізованого чеку	118
Закриття зміни та створення Z-Звіту	118
Ломбардні операції	119
Приклад використання API для ломбардних операцій	120
Приклад реалізації функції для виклику API методів ПРРО Cashalot	122

Загальна інформація

Розроблені API ПРРО Cashalot використовуються для інтеграції ПРРО з розрахунково-обліковими системами. В даному документі приведено специфікацію методів бібліотек для запропонованих варіантів підключення:

Додано можливість одночасної роботі декількох ПРРО Cashalot в одній обліковій системі використовуючи бібліотеку (**CashaLotApiBridgeHlp.dll** та метод **SetParameterEx**), що сумісні з усіма нижче перелічуваними варіантами.

Варіант 1 - драйвер ПРРО Cashalot для BAS призначений для інтеграції ПРРО Cashalot з розрахунково-обліковою системою BAS (API драйвер **CashalotApiDrv**).

Варіант 2 - API ПРРО Cashalot призначений для підключення до сторонніх розрахунково-облікових систем:

а) з використанням COM-методів (**COM API CashalotApi**);

б) на основі .NET-технології (**API CashalotApiBridge**).

Варіант 3 - сервер API ПРРО Cashalot призначений для взаємодії з фіскальним сервером програмних реєстраторів розрахункових операцій.

Поставляється у вигляді окремого застосунку, що розгортається локально на стороні користувача як **web-служба**.

Інтеграція надає користувачам можливість безпосередньо з розрахунково-облікової системи скористатися функціоналом програмного забезпечення Cashalot:

- Відкриття та закриття касової зміни;
- Створення та реєстрація чеків продажу/повернення;
- Створення видаткового чека на основі чека продажу та його реєстрація;
- Формування та друк Z-Звіту за даними фіскального сервера контролюючого органу (скорочено ФСКО)/X-Звіту за даними ПРРО;
- **Створення та реєстрація службових чеків:**
 - - Службове внесення;
 - - Службова видача;
- **Реєстрація чека:**
 - - Засвідчення документа кваліфікованим електронним підписом (КЕП);
 - - Відправка по API XML чека на фіскальний сервер контролюючого органу;
 - - Отримання та обробка позитивної відповіді від ФСКО (отримання фіскального номера);
 - - Отримання та обробка негативної відповіді;
- **Робота в режимі Офлайн:**
 - - Контроль зв'язку з фіскальним сервером контролюючого органу;
 - - Використання резервних номерів;
 - - Контроль таймера 36 (36 годин однієї офлайн-сесії);
 - - Контроль таймера 168 (168 годин офлайн-сесій на місяць);
 - - Збереження чеків у Тимчасовий ланцюжок;
 - - Реєстрація непереданих чеків (контрольна стрічка), створених в режимі Офлайн (після переходу в режим Онлайн);
 - - Авторизація по КЕП (вхід в програму);
 - - Відображення на екрані (візуалізація) зареєстрованого чека на пристрої продавця з QR-кодом після успішної оплати;
 - - Друк зареєстрованого чека покупцю за допомогою принтера чеків.
- **Робота з обладнанням:**
 - - Інтеграція зі сканером штрих-кодів;

- Інтеграція з принтером чеків.

1. Поняття і визначення

КЕП – кваліфікований електронний підпис.

Локальний номер чека – номер, призначений чеку системою виписки електронних касових чеків, що використовується продавцем.

Офлайн сесія – сукупність документів, створених в режимі офлайн, між припиненням і відновленням зв'язку ПРРО з ФСКО.

ПРРО – програмний реєстратор розрахункових операцій. Система виписки електронних касових чеків, що використовується продавцем.

Розрахунково-облікова платформа – платформа BAF (реалізована підтримка версій 7.7 і вище).

Розрахунково-облікова система – стороння система обліку розрахунків.

Фіскальний номер чека – номер, призначений чеку Сервером ФСКО.

ФСКО – фіскальний сервер контролюючого органу.

BAS – конфігурація Business Automation Software for Retail для розрахунково-облікової платформи BAF.

Обов'язкові Параметри* відповідають за ідентифікацію та виклик необхідних компонентів, функцій, параметрів то що.

Далі розглянуто варіанти роботи розрахунково-облікових систем або платформ з API бібліотеками ПРРО Cashalot:

Варіант 1. CashalotApiDrv32.dll, CashalotApiDrv64.dll

Варіант 2а. CashalotApi32.dll, CashalotApi64.dll

Варіант 2б. CashalotApiBridge.dll

Варіант 3. Ознайомитися з умовами підключення і можливостями роботи з серверним рішенням API ПРРО Cashalot можна за телефоном (044) 344-07-44.

2. Апаратні та програмні вимоги

Для оптимальної роботи ПРРО Cashalot, що є клієнтським десктопним програмним забезпеченням, рекомендовані системні та програмні вимоги:

Операційна система

Програмний комплекс коректно функціонує на комп'ютерах з сучасними операційними системами Microsoft та компонентами, що входять до її складу:

глобальні системні оновленнями, що презентує компанія Microsoft для систем безпеки та інших програмних компонентів в залежності від операційної системи яка використовується;

рекомендовано використовувати останню версію ОС, наприклад, Windows 10 або Windows Server 2019 відповідно до потреб.

Апаратне забезпечення:

- процесор з мінімальною тактовою частотою від 2 ГГц;
- оперативна пам'ять від 2 GB;
- вільне місце на жорсткому диску від 1,5 GB;
- кольоровий графічний дисплей (розподільна здатність від 1440 x 900);
- маніпулятор типу миша та клавіатура;
- встановлений український або російський мовний стандарт регіональних налаштувань операційної системи;
- підтримка кирилиці в операційній системі;
- доступ до поштового сервера;
- наявний зв'язок з фіскальним сервером контролюючого органу.

Мережа

Для роботи користувача необхідний доступ до мережі Інтернет. Необхідна пропускна здатність каналу залежить від інтенсивності роботи.

3. Додаткові пристрої

Можливий набір пристроїв для обладнання робочого місця касира:

- системний пристрій з встановленими програмними модулями комп'ютер/планшет);
- монітор касира;
- принтер чеків;
- платіжний пристрій (POS-термінал);
- ваги;
- сканер штрих-кодів.

Варіант 1 – 1С BAS драйвер ПРРО Cashalot, CashalotApiDrv32.dll або CashalotApiDrv64.dll

В конфігураторіBAS для драйверу фіскального реєстратора вказуються функції, призначені для розробки драйверів обладнання, що підключається. Ці функції відправляють запити і отримують відповіді від драйверу **CashalotApiDrv32.dll/ CashalotApiDrv64.dll** (далі - **CashalotApiDrv**).

Інструкція по підключенню і налаштуванню драйверу Cashalot для інтеграції в BAS детально описує підготовку до роботи в обліково-розрахунковій системі.

1. Ініціалізація драйверу:

Для використання драйверу необхідно зареєструвати бібліотеки **CashalotApiDrv32.dll/ CashalotApiDrv64.dll**.

Для цього в командному рядку Windows виконайте команду:

```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApiDrv32.dll
```

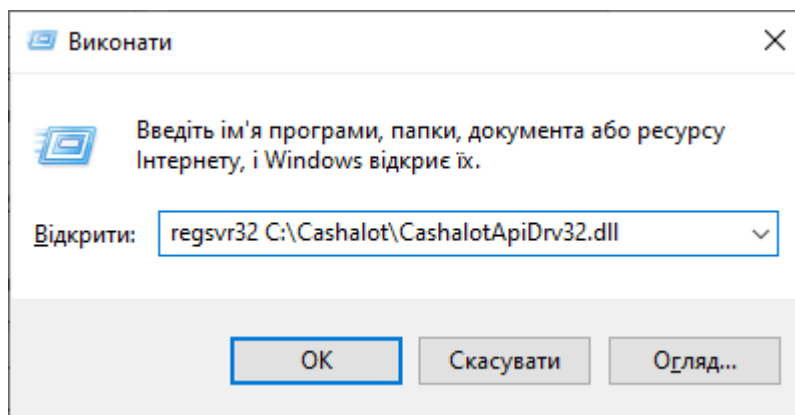
(для 32-бітної платформи)

```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApiDrv64.dll
```

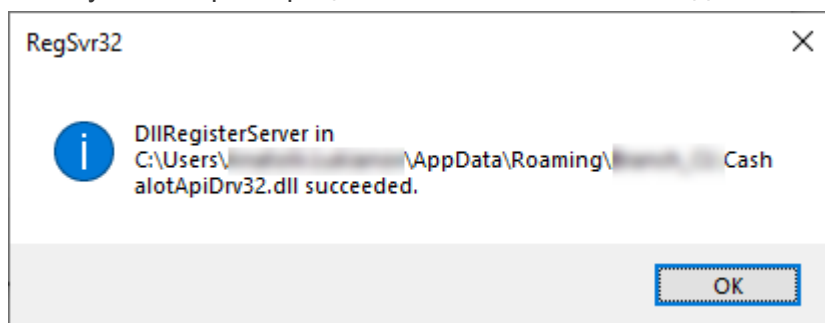
(для 64-бітної платформи)

Наприклад:

```
regsvr32 C:\Users\User1\AppData\Local\Cashalot\CashalotApiDrv32.dll
```



Після успішної реєстрації бібліотеки з'явиться повідомлення зразку:



Ініціалізація компонент відбувається шляхом створення об'єкту відповідно до синтаксису:

*ПідключитьВнешнююКомпоненту(<Местоположение>, <Имя>, ТипВнешнейКомпоненты.Native);
Компонента = Новый("AddIn. <Имя>.AddInNativeExtension");*

Драйвер ПРРО Cashalot ініціалізується наступним чином:

ПідключитьВнешнююКомпоненту("C:\Cashalot\CashalotApiDrv32.dll", "CashalotApiDrv ")

ОбъектДрайвера = Новый("AddIn.CashalotApiDrv.CashalotApiDrv");

В драйвері CashalotApiDrv реалізована обробка наступних функцій:

ПолучитьНомерВерсии

ПолучитьОшибку

ПолучитьПараметры

УстановитьПараметр
 Подключить
 Отключить
 ТестУстройства
 ОткрытьСмену
 ОткрытьЧек
 НапечататьФискальнуюСтроку
 НапечататьФискСтроку
 НапечататьПериодическийОтчетПоДатам
 УстановитьРасширенныйПорядокОплатыЧека
 ЗакрыватьЧек
 ОтменитьЧек
 НапечататьЧекВнесенияВыемки
 НапечататьОтчетБезГашения
 НапечататьОтчетСГашением
 ЗакрыватьСмену

Необхідно встановити налаштування в 1С КонфігураторіBAS таким чином щоб драйвер мав можливість розрізняти різні екземпляри ПРРО Cashalot, використовуючи їх одночасно для декількох екземплярів ПРРО Cashalot.

Користувач має можливість налаштувати власну конфігурацію 1С BAS для декількох екземплярів ПРРО одночасно, для цього необхідно вказати обов'язкові значень параметру.

Обов'язкові значення параметру*

Параметр: **DeviceIDFnRRO***, **“PathToCashalotDir”**.

Параметри функції зазначені в Таблиці 1.1.

Таблиця 1.1 Параметри функції опису помилки.

Ім'я	Опис
“DeviceIDFnRRO”	Фіскальний номер ПРРО
“PathToCashalotDir”	Шлях до каталогу з встановленим ПРРО Cashalot

виконує завдання опису властивостей, унікальне значення параметру («ИДУстройства») призначатися та відображається, для кожного екземпляру окремо.

За умови, що було виконано процедуру підключення відповідних бібліотек, що відповідальні за ініціалізацію компонент шляхом створення об'єкту, відповідно до синтаксису, встановлених функцією драйверу, за їхніми іменами.

Рекомендовано кожен операцію в BAS супроводжувати встановлювати обов'язкове значення параметру*; Фіскальний номер ПРРО* та (шлях до ПРРО Cashalot)*.

1. УстановитьПараметр
2. Операція (наприклад, реєстрація чеку)

Отримання номеру версії драйверу

Для отримання номеру версії драйверу використовується функція

СТРОКА **ПолучитьНомерВерсии()**

Приклад:

Версия = ОбъектДрайвера.ПолучитьНомерВерсии();

Отримання опису помилки

Для отримання опису помилки використовується функція

ЧИСЛО **ПолучитьОшибку(ОписаниеОшибки)**

Параметри функції зазначені в Таблиці 1.2.

Таблиця 1.2 Параметри функції опису помилки.

Ім'я	Тип	Опис
ОписаниеОшибки	СТРОКА [OUT]	Опис помилки

Приклад:

ИмяПараметра = "DeviceIDFnRRO";

ЗначениеПараметра = "4000001234";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

ИмяПараметра = "PathToCashalotDir";

ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

ОбъектДрайвера.ПолучитьОшибку(*ВыходныеПараметры*[1]);

2. Отримання налаштувань драйверу

Для отримання списку параметрів налаштувань драйверу та їх типів використовують функцію

БУЛЕВО ПолучитьПараметры (ТаблицаПараметров)

Параметри функції зазначені в Таблиці 2.1

Таблица 2.1 Параметри функції отримання параметрів драйверу.

Ім'я	Тип	Опис
ТаблицаПараметров	СТРОКА [OUT]	Опис параметрів.

Таблица списку параметрів представляє собою XML-рядок.

Зауваження. Функція не повертає поточні значення параметрів.

Приклад:

ИмяПараметра = "DeviceIDFnRRO";

ЗначениеПараметра = "4000001234";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

ИмяПараметра = "PathToCashalotDir";

ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

ОбъектДрайвера.ПолучитьПараметры(*ПараметрыДрайвера*);

3. Встановлення налаштувань драйверу

Сценарії користувача, що встановлюють значень параметрів налаштувань драйверу декількох екземплярів ПРРО Cashalot, кожен з яких має унікальним значенням параметрів, встановлених функцією драйверу, за їхніми іменами використовують функцію

БУЛЕВО УстановитьПараметр(*Имя*, *Значение*)

Параметри функції зазначені в Таблиці 3.1.

Таблица 3.1 Параметри функції установки параметрів драйверу.

Ім'я	Тип	Опис
Имя	СТРОКА	Ім'я параметру
Значение	СТРОКА	Значення параметру

При роботі з **CashalotApi**, необхідно зазначити такі параметри:

Ім'я	Опис	Обов'язкові
"DeviceIDFnRRO"	Фіскальний номер ПРРО	Так, для Всіх
"PathToCashalotDir"	Шлях до каталогу з встановленим ПРРО Cashalot	Так, для Всіх

"DEFAULTPRINTERNAME"	Визначає принтер для друку при натисканні кнопки "Друк" в діалогових вікнах: 1) "" – пустий рядок, виводиться діалог вибору принтера. 2) "DEFAULTWINDOWS" – використовувати принтер Windows за замовчуванням 3) "ім'я принтера чеків" - в параметр передається ім'я принтера чеків, який буде використовуватись для друку чеків в ПРРО Кашалот.	Так, для цієї функції
"AUTOPRINTMODE"	Параметр визначає автоматичну відправку на друк чеку або службового звіту без відображення інформаційного вікна "True" – відправляти на друк "False" – не відправляти на друк Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), то чек одразу друкується, в іншому випадку відображається вікно вибору принтера для друку.	Так, для цієї функції
OtherParametrs	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.	Так, для цієї функції

Приклад:

```

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);
Для Каждого Параметр Из Параметры Цикл
    Если Лев(Параметр.Ключ, 2) = "P_" Тогда
        ЗначениеПараметра = Параметр.Значение;
        ИмяПараметра = Сред(Параметр.Ключ, 3);
        Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра,
ЗначениеПараметра)
    КонецЕсли;
КонецЦикла;

```

4. Підключення фіскального реєстратора ПРРО Cashalot

Для підключення ПРРО Cashalot з поточними значенням параметрів, встановлених функцією «УстановитьПараметр», використовується функція

БУЛЕВО Підключить(ИДУстройства)

Параметри функції зазначені в Таблиці 4.1

Таблиця 4.1 Параметри функції підключення пристрою.

Ім'я	Тип	Опис
------	-----	------

ИДУстройства	СТРОКА [OUT]	Ідентифікатор пристрою
--------------	--------------	------------------------

При успішному підключенні, функція повертає ідентифікатор підключеного ПРРО.

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Ответ = ОбъектДрайвера.Подключить(*ПараметрыПодключения.ИДУстройства*);

5. Відключення екземпляра фіскального реєстратора ПРРО Cashalot

Для відключення ПРРО Cashalot(), в залежності від налаштувань користувача з одним чи декількома встановленими екземплярами ПРРО(), використовується функція «УстановитьПараметр», яка використовується функцією

БУЛЕВО Отключить(**ИДУстройства**)

Параметри функції зазначені в Таблиці 5.1.

Таблиця 5.1 Параметри функції відключення пристрою.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Функція повертає ідентифікатор відключеного ПРРО().

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "3000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Ответ = ОбъектДрайвера.Отключить(*ПараметрыПодключения.ИДУстройства*);

6. Перевірка ПРРО Cashalot

Сценарій в якому функція використовується користувачем для перевірки одного чи декількох ПРРО(), шляхом пробного підключення і опитування пристрою з поточними значеннями параметрів, встановлених для кожного екземпляру окремо, функцію «УстановитьПараметр», використовується функція

БУЛЕВО ТестУстройства (**Описание**, **ПризнакДемоРежима**)

Параметри функції зазначені в Таблиці 6.1.

Таблиця 6.1 Параметри функції

Ім'я	Тип	Опис
Описание	СТРОКА [OUT]	Опис результату виконання тесту
ПризнакДемоРежима	СТРОКА [OUT]	Опис обмежень демонстраційного режиму при його наявності, і пустий результат при його відсутності.

При успішній перевірці повертається повідомлення "Тест пристрою виконано вдало" та результат перевірки приймає значення ИСТИНА.

Увага! На даному етапі розробки результат виконання функції не містить значень вихідних параметрів.

Приклад:

```
ИмяПараметра = "DeviceIDFnRRO";  
ЗначениеПараметра = "4000001234";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
ИмяПараметра = "PathToCashalotDir";  
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
Ответ = ОбъектДрайвера.ТестУстройства("", "");
```

7. Відкриття зміни

Сценарій в якому функція використовується користувачем при відкритті зміни, в залежності від налаштувань для одного ПРРО чи для декількох екземплярами ПРРО, для кожного екземпляру окремо необхідно вказати обов'язкові значення параметру*-'ИДУстройства', щоб мати змогу застосувати функцію

БУЛЕВО ОткрытьСмену(ИДУстройства)

Параметри функції зазначені в Таблиці 7.1. При виклику цієї функції драйвер ПРРО Cashalot викликає вікно авторизації, в якому користувач зазначає папку з файловим ключом касира або старшого касира, пароль, після чого відбувається відкриття касової зміни.

Таблиця 7.1 Параметри функції відкриття зміни.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Приклад:

```
ИмяПараметра = "DeviceIDFnRRO";  
ЗначениеПараметра = "4000001234";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
ИмяПараметра = "PathToCashalotDir";  
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
Результат = ОткрытьСмену(ОбъектДрайвера, Параметры, ПараметрыПодключения,  
ВходныеПараметры, ВыходныеПараметры);
```

8. Реєстрація чеків

При реєстрації чеків дані передаються через ланцюжок функцій BAS, призначених для друку чеків на фіскальних реєстраторах:

БУЛЕВО ОткрытьЧек(ИДУстройства, ФискальныйЧек, ЧекВозврата, АннулироватьОткрытыйЧек, НомерЧека, НомерСмены)

БУЛЕВО НапечататьФискальнуюСтроку(ИДУстройства, Наименование*, КодУКТЗЕД, Артикул*, Штрихкод, ЕдиницаИзмерения*, НомерСекции, Количество*, Цена*, Сумма*, ПроцентСкидки, СуммаСкидки, НДС*, ПризнакСуммаВключаетНДС, Сумма НДС, ПризнакАкцизногоНалога НомерСтроки, ДопРеквизиты*);

або

БУЛЕВО НапечататьФискСтроку (ИДУстройства, Наименование*, Количество*, Цена*, Сумма*, НомерСекции, СтавкаНДС*, ДопРеквизиты*);

БУЛЕВО PrintNoFiscString (ИДУстройства, Коментарий);

БУЛЕВО ЗакретьЧек (ИДУстройства, НаличнаяОплата, ОплатаКартой, ОплатаКредитом, ОплатаСертификатом, СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления, ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат, НомерЧекаЧисловой, ФискальныйНомерСозданногоЧека);

До стандартного виразу функцій формування чека було додано новий логічний рядок:

ОткрытьЧек+НапечататьФискСтроку+НапечататьНефискСтроку+ЗакретьЧек, додана нова функцій:

УстановитьРасширенныйПорядокОплатыЧека/SetCheckExtendetPaymentOrderInfo

Новая цепочка:

ОткрытьЧек+НапечататьФискСтроку+НапечататьНефискСтроку+УстановитьРасширенныйПорядокОплатыЧека+ЗакретьЧек

Параметри, відмічені зірочкою (*), є обов'язковими при передачі даних товару.

Функція **ОткрытьЧек** формує заголовок чеку.

Функції **НапечататьФискальнуюСтроку** та **НапечататьФискСтроку** призначені для заповнення чеку рядками товарами, що купуються. Ці функції рівноцінні, проте, перша з них має повний перелік параметрів товару, а друга має скорочений перелік. Обидві функції мають параметр **ДопРеквизиты**, в якому передається сформований XML-рядок з переліком реквізитів товару. Реквізити, які зазначені в XML-рядку, мають пріоритет над явно заданими параметрами функції, тому рекомендується використовувати **ДопРеквизиты**, а також функцію **НапечататьФискСтроку** з використанням цього параметру. Крім того, у випадку використання функції **НапечататьФискСтроку**, в **ДопРеквизиты** необхідно обов'язково вказати значення для артикулу та одиниці вимірювання товару.

Функція **ЗакретьЧек** виконує завершення формування чеку, передає дані в ПРРО Cashalot, де чек підписується і відправляється на сервер ФСКО для реєстрації, та повертає в якості кінцевих параметрів фіскальний номер зареєстрованого чеку та номер для наступного чеку. Порушення послідовності функцій призведе до помилки реєстрації чеків.

ТОВ "Тест-гідрометцентр"	
Тест-гідромет	
УКРАЇНА, м. Київ, Адреса	
ІД 24234227	
ФН ПРРО 4000007619	
КАСІР Підромет П. П.	
Чек №297664	
НазваТовару1	
1 X 20000,03	20000,03 A
Знижка	-1000,00
НазваТовару2	
1 X 20000,03	20000,03 A
Знижка	-1000,00

Сума, грн	38000,06
ПДВ А=20%	6333,34

Заокруглення	0,04
До сплати	38000,10
ГОТІВКА, грн	20000,00
Решта, грн	999,90
КАРТКА	19000,00
297664	3/4/2021 3:04:15 PM



ФІСКАЛЬНИЙ ЧЕК ОНЛАЙН
Cashalot

[Друкувати звіт](#)

[Готово](#)

Заокруглення суми по чеку реалізовано наступним чином. В функцію **ЗакретьЧек** передається значення готівки, що надана клієнтом на касі, значення оплати картою, сертифікатом та кредитом, а також заокруглена сума, розрахована за певним алгоритмом користувача (наприклад, відповідно до Постанови НБУ¹, 1-4 коп заокруглюється до 0 коп, 5-9

¹ <https://zakon.rada.gov.ua/laws/main/v0025500-18> п.4.

коп заокруглюється до 10 коп). В чеку буде відображено заокруглення, надану готівку та інші форми оплати, заокруглену суму до сплати, решту:

Для друку нефіскальних рядків допускається один виклик функції **PrintNoFiscString()**. Якщо необхідно надрукувати декілька нефіскальних рядків, необхідно побудувати параметр *Коментарий* у вигляді:

"<Перший рядок коментаря> +Символы.ПС + < Другий рядок коментаря >
>+Символы.ПС +...+Символы.ПС +< Останній рядок коментаря >

де *Символы.ПС* – символ переходу на новий рядок. При побудові нефіскального рядку необхідно врахувати ширину чекової стрічки та допустиму кількість символів в одному рядку. У випадку, якщо параметр

СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления не визначений (null), або дорівнює нулю, то він замінюється сумою по всім формам оплати. В такому випадку заокруглення буде дорівнювати нулю.

Параметри функцій зазначені в Таблиці 8.1.

Таблиця 8.1 Параметри функцій для формування чеку.

Функція	Параметри функцій		
	Ім'я	Тип	Опис
ОткрытьЧек	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	ФискальныйЧек	БУЛЕВО	Ознака фіскального чеку
	ЧекВозврата	БУЛЕВО	Ознака чеку повернення
	АннулироватьОткрытыйЧек	БУЛЕВО	Ознака автоматичного анулювання раніше відкритого чеку
	НомерЧека	ЧИСЛО [OUT]	Номер нового чеку
	НомерСмены	ЧИСЛО [OUT]	Номер відкритої зміни
НапечататьФискальнуюСтроку	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	Наименование	СТРОКА	Найменування товару, відповідно до специфікації ФСКО, допускається довжина до 127 символів
	КодУКТЗЕД	СТРОКА	Код УКТЗЕД
	Артикул	СТРОКА	Артикул, відповідно до специфікації ФСКО, допускається довжина до 63 символів.
	Штрихкод	СТРОКА	Штрихкод
	ЕдиницаИзмерения	СТРОКА	Одиниця вимірювання
	НомерСекции	СТРОКА	Номер секції (не використовується)

	Количество	ЧИСЛО	Кількість товару
	Цена	ЧИСЛО	Ціна одиниці товару (в форматі <Гривні, Копійки>)
	Сумма	ЧИСЛО	Кінцева сума по позиції чеку (з врахуванням всіх націнок та знижок, в форматі <Гривні, Копійки>)
	ПроцентСкидки	ЧИСЛО	Процент знижки (ігнорується, якщо заповнена сума знижки)
	СуммаСкидки	ЧИСЛО	Сума знижки на позицію
	НДС	ЧИСЛО	ПДВ в процентах (від 0 до 99,99)
	ПризнакСуммаВключаетНДС	БУЛЕВО	Ознака, що сума включає ПДВ
	СуммаНДС	ЧИСЛО	ПДВ в процентах (від 0 до 99,99)
	ПризнакАкцизногоНалога	БУЛЕВО	Ознака, що сума включає ПДВ
	НомерСтроки	БУЛЕВО	Номер рядку (не використовується)
	ДопРеквизиты	СТРОКА	Додаткові параметри (XML таблиця)
НапечататьФискСтроку	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	Наименование	СТРОКА	Найменування товару, відповідно до специфікації ФСКО, допускається довжина до 127 символів
	Количество	ЧИСЛО	Кількість товару
	Цена	ЧИСЛО	Ціна одиниці товару (в форматі <Гривні, Копійки>)
	Сумма	ЧИСЛО	Кінцева сума по позиції чеку (з врахуванням всіх націнок та знижок, в форматі <Гривні, Копійки>)
	НомерСекции	СТРОКА	Номер секції (не використовується)
	СтавкаНДС	ЧИСЛО	ПДВ в процентах (від 0 до 99,99)
	ДопРеквизиты	СТРОКА	Додаткові параметри

PrintNoFiscString	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	Коментарий	СТРОКА	Нефіскальний рядок, коментар. Для побудови декількох рядків використовуйте +Символы.ПС +
ЗакрытьЧек	<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
	НаличнаяОплата	ЧИСЛО	Сума оплати готівкою
	ОплатаКартой	ЧИСЛО	Сума оплати картою
	ОплатаКредитом	ЧИСЛО	Сума оплати кредитом
	ОплатаСертификатом	ЧИСЛО	Сума оплати сертифікатом
	СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления	ЧИСЛО	Сума до оплати, що враховує всі форми оплати, знижки та заокруглення
	ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат	СТРОКА	Фіскальний номер чеку, за яким виконується повернення
	НомерЧекаЧисловой	ЧИСЛО [OUT]	Номер нового чеку
	ФискальныйНомерСозданногоЧека	СТРОКА [OUT]	Фіскальний номер нового чеку

*** обов'язкові параметри.**

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

ДопРеквизиты

Текст в форматі XML з кодуванням UTF-8, що передається за допомогою параметрів типу **СТРОКА**. Атрибути XML-рядку вказані в Таблиці 8.2.

Нагадаємо, що параметри рядку чеку, які вказані в XML-рядку мають пріоритет над параметрами, які явно передані в функції НапечататьФискальнуюСтроку або НапечататьФискСтроку, тому допускається передача цих параметрів рядку чеку через XML-рядок.

Таблица 8.2. Атрибути XML-рядку параметру ДопРеквизиты.

Атрибут XML-рядку	Опис атрибуту
NOMENCLATURE	Найменування товару, відповідно до специфікації ФСКО, допускається довжина до 127 символів
UKTZED	Код УКТЗЕД
VENDORCODE	Артикул, відповідно до специфікації ФСКО, допускається довжина до 63 символів
BARCODE	Штрихкод

NEW

DIMENSION	Одиниця вимірювання
DISCOUNT	Процент знижки (ігнорується, якщо заповнена сума знижки)
DISCOUNTAMOUNT	Сума знижки
VAT	Ставка ПДВ в процентах
PRICEINCLUDESVAT або PRICEINCLUDEVAT	Ознака, що сума включає ПДВ
AMOUNTVAT	Сума ПДВ, в форматі <Гривні, Копійки>
ISEXCISABLE	Ознака підакцизного товару, True/False
EXCISESTAMPBARCODE	Штрих-код марки акцизного податку. Поле ігнорується, якщо ознака підакцизного товару ISEXCISABLE = false.
OtherParameters	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

Параметр **OtherParameters** передається в форматі XML з кодуванням UTF-8. Атрибути XML-рядка вказані в таблиці нижче.

Зазначимо, що параметри XML-рядка OtherParameters мають пріоритет над параметрами, які явно передані в ReceiptLst.

Атрибути XML-рядка OtherParameters

XML атрибут	Опис атрибута
NOMENCLATURE *	Найменування товару
UKTZED	Код УКТЗЕД
VENDORCODE *	Артикул
BARCODE	Штрих-код
DIMENSION *	Одиниця вимірювання
ParentReceiptFiscalNumber	Фіскальний номер чеку по якому здійснюється доплата/повернення
SaleOrderNumber	Номер замовлення/договору
SumPreparePaid	Вказуємо суму передоплати, вказуємо загальну суму по чеку.
DISCOUNT	Процент знижки (ігнорується, якщо заповнена сума знижки)
DISCOUNTAMOUNT	Сума знижки

VAT *	Ставка ПДВ в процентах
PRICEINCLUDESVAT або PRICEINCLUDEVAT	Ознака, що ціна враховує ПДВ
AMOUNTVAT	Сума ПДВ, у форматі <Гривні, Копійки>
ISEXCISABLE	Ознака підакцизного товару, True/False

* обов'язкові параметри в рядку OtherParams, якщо не вказані відповідні явні параметри. Опціональний об'єкт Comment використовується для додавання нефіскальної інформації:

Comment	Опціональний рядок нефіскальної інформації, коментар. Допускається використання одного коментарю на чек. За необхідності сформулювати коментар з декількох рядків, для переходу на новий рядок використовується комбінація "\r" або "\n".
---------	---

Об'єкт DocType є обов'язковим при проведенні ломбардних операцій, призначений для визначення типу документа (чека). Ломбардні операції описані в розділі ["Ломбардні послуги"](#).

DocType	Тип документа 0 – чек продажу 4 – ломбардний чек За замовчуванням DocType = 0, якщо об'єкт не задано.
---------	--

Структура **JSONPayData** містить підсумки оплати по чеку, і має наступні поля:

Поле	Значення
SumCash	Сума оплати чеку готівкою
SumPayByCard	Сума оплати чеку банківською картою
SumPayByCredit	Сума оплати чеку в кредит
SumPayByCertificate	Сума оплати чеку сертифікатом
SumPayCheck	Сума чеку до оплати, з урахуванням всіх надбавок і знижок
SumPreparePayed	Сума попередньої оплати
CustomerEmail	Опціонально. Електронна адреса покупця, на яку можна відправити фіскалізований чек, передається в поле електронної адреси при натисканні кнопки "Надіслати чек на Email" екрану відображення чеку. Рекомендовано використовувати при встановленому параметрі відображення вікон інтерфейсу: SetParameter("NOINTERFACEMODE","False")
Необов'язкові параметри інформації по транзакції терміналу при безготівкових розрахунках:	
TerminalID	Код терміналу

ApprovalCode	Код підтвердження
RRN	PPH транзакції
IssuerName	Платіжна система
PAN	Номер картки
TransactionDate	Дата транзакції (в форматі dd.MM.yyyy HH:mm:ss)
SignVerif	Електроний підпис
AcquireName	Термінал
InvoiceNumber	Номер чеку
ParentRRN	PPH транзакції чека по якому здійснено повернення

Увага! Різниця між SumPayCheck та сумою по всіх видах оплати вважається значенням заокруглення. Тому розробник має самостійно реалізувати алгоритм заокруглення значення виданої готівки або суми оплати по чеку.

Відповідь

В якості відповіді повертається структура CashalotApiRetVal, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документу (див. нижче)
XML	String	XML-структура фіскалізованого чеку

Тип документу "*Type*" визначається числом, яке формується наступним чином:

1<Клас документу><Тип документу><Додатковий тип документу>

Де

Число	Кількість знаків	Значення
<Клас документу>	1	0 - Чек 1 - Z-звіт 2 - X-звіт
<Тип документу>	3	000 - Чек реалізації товарів/послуг 001 - Чек переказу коштів 002 - Чек операції обміну валюти 003 - Чек видачі готівки 100 - Відкриття зміни 101 - Закриття зміни

		102 - Початок офлайн сесії 103 - Завершення офлайн сесії
<Додатковий тип документу>	3	000 - Касовий чек (реалізація) 001 - Видатковий чек (повернення) 002 - Чек операції «службове внесення»/«отримання авансу» 003 - Чек операції «отримання підкріплення» 004 - Чек операції «служба видача»/«інкасація» 005 - Чек сторнування попереднього чека

9. Відміна відкритого чеку

Для відміни раніше відкритого чеку використовується функція

БУЛЕВО ОтменитьЧек(ИДУстройства)

Параметри функції зазначені в Таблиці 9.1.

Таблиця 9.1 Параметри функції відміни чеку.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Приклад:

Результат = ОтменитьЧек(ОбъектДрайвера, Параметры, ПараметрыПодключения, ВыходныеПараметры);

10. Розгорнутий порядок сплати поточного чека

Для встановлення розширеного порядку сплати поточного чеку, використовується функція:

БУЛЕВО

УстановитьРасширенныйПорядок(ОткрытьЧек+НапечататьФискСтроку+НапечататьНефискСтроку+УстановитьРасширенныйПорядокОплатыЧека+ЗакрытьЧек)

Для різних варіантів потрібно встановити наступний порядок оплати:

PaymentOrderType = 1/2/3, заповнити поле суму передоплати **SumPreparePayed**, сума післяплати буде перерахована автоматично від суми всіх рядків чека за вирахуванням суми передоплати.

У чеку післяплати (**PaymentOrderType** = 2) потрібно вказати:

фіскальний номер чека за яким була здійснена передоплата **ParentReceiptFiscalNumber**
фіскальний номер каси, на якій була передоплата:

- якщо він відрізняється
- якщо не відрізняється, встановити значення нулл або порожній рядок.

Таблиця 10.1 Параметри функції встановлення розширеного порядку поточного чеку ПРРО.

JsonVal	String	JSON-структура, що містить значення успішності операції, ОзнакаСкороченогоЗвіту, Base64 рядок XML структури чеку	
DeviceID	String	Ідентифікатор устрою	ФіскальнийНомерПРРО
PaymentOrderType	Bool	Спосіб оплати: 0 - звичайна 1 - передоплата	Для різних варіантів потрібно встановити спосіб оплати PaymentOrderType = 1/2/3,

		2 - післяплата 3 - інтернет продаж	заповнити поле суму передоплати SumPreparePayed У чеку післяплати PaymentOrderType == 2 потрібно вказати: ParentReceiptFiscalNumber фіскальний номер чека за яким було здійснено передоплату. Встановлюємо спосіб оплати "ПЕРЕДОПЛАТА" VARIANT PaymentOrderType, VARIANT SumPreparePayed, VARIANT SaleOrderNumber, VARIANT ParentReceiptFiscalNumber, VARIANT OtherParameters. PaymentOrderType = ; //Спосіб оплати (0-звичайна 1 - передоплата 2 - післяплата 3 - інтернет продаж)
SumPreparePayed	String	Сума передоплати	Сума готівки наданої покупцем(може бути більшою за суму до сплати, з неї вираховується решта).
SaleOrderNumber	String	Номер замовлення/договору	VARIANT ParentReceiptFiscalNumber присвоїти номер договору/замовлення в залежності від обраного параметру, що (друкується в чеці)
ParentReceiptFiscalNumber	String	Фіскальний номер чека по якому здійснюється	Фіскальний номер чека по якому здійснюється доплата/повернення
OtherParameters	String	Додаткові параметри XML	зарезервовано - не використовується

Використання драйвера 1С, для стандартного виразу функцій формування чека:

(ОткрытьЧек+НапечататьФискСтроку+НапечататьНефискСтроку+ЗакрытьЧек)

Додана нова функція:

УстановитьРасширенныйПорядокОплатыЧека/SetCheckExtendetPaymentOrderInfo

та новий функціональний вираз(ланцюжок):

ОткрытьЧек+НапечататьФискСтроку+НапечататьНефискСтроку+УстановитьРасширенныйПорядокОплатыЧека+ЗакрытьЧек

Опис функції та полів:

Встановити розширений порядок оплати для поточного чека:

bool **STDMETHODCALLTYPE** **SetCheckExtendetPaymentOrderInfo**(**BSTR** DeviceID, **VARIANT** PaymentOrderType, **VARIANT** SumPreparePayed, **VARIANT** SaleOrderNumber, **VARIANT** ParentReceiptFiscalNumber, **VARIANT** OtherParameters)

Приклади виконання функції FiscalizeCheck/FiscalizeReturnCheck для різних способів та видів оплати чеку

Чек предоплата

JSONGoodsLstData=

```
{ "ReceiptLst": [ { "VendorCode": "", "Article": "", "Name": "ТоварПредоплатыПослеплаты1", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "1", "Barcode": "", "UnitType": "шт", "Quantity": "1", "Price": "1000", "Amount": "1000", "DiscountPrc": "", "DiscountSum": "", "VATRate": "", "IsPriceIncludeVAT": true, "SumVAT": "", "IsExcise": false, "OtherParams": null, "VATLetter": "", "ExciseLetter": "", "ExciseStampBarcode": "", "DocNumber": "" }, { "VendorCode": "", "Article": "", "Name": "ТоварПредоплатыПослеплаты2", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "1", "Barcode": "", "UnitType": "шт", "Quantity": "2", "Price": "1000", "Amount": "2000", "DiscountPrc": "", "DiscountSum": "", "VATRate": "", "IsPriceIncludeVAT": true, "SumVAT": "", "IsExcise": false, "OtherParams": null, "VATLetter": "", "ExciseLetter": "", "ExciseStampBarcode": "", "DocNumber": "" } ], "Comment": null, "DocType": "0", "DocSubType": "0", "SaleOrderNumber": "555444333" }
```

JSONPayData =

```
{ "SumCash": "1500", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "1500", "CustomerEmail": null, "TerminalID": null, "ApprovalCode": null, "RRN": null, "IssuerName": null, "PAN": null, "TransactionDate": null, "SignVerif": null, "AcquireName": null, "InvoiceNumber": null, "ParentRRN": null, "PaymentOrderType": "1", "SumPreparePaid": "1500", "ParentReceiptFiscalNumber": null, "OtherParams": null }
```

Чек післяплати

JSONGoodsLstData =

```
{ "ReceiptLst": [], "Comment": null, "DocType": "0", "DocSubType": "0", "SaleOrderNumber": "555444333" }
```

JSONPayData =

```
{ "SumCash": "1600", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "1500", "CustomerEmail": null, "TerminalID": null, "ApprovalCode": null, "RRN": null, "IssuerName": null, "PAN": null, "TransactionDate": null, "SignVerif": null, "AcquireName": null, "InvoiceNumber": null, "ParentRRN": null, "PaymentOrderType": 2, "SumPreparePaid": "1500", "ParentReceiptFiscalNumber": "604779", "OtherParams": null }
```

Важливо: обов'язковим параметром є **ParentReceiptFiscalNumber**

Чек інтернет-продажу:

JSONGoodsLstData =

```
{ "ReceiptLst": [ { "VendorCode": "", "Article": "", "Name": "ТоварІнтернетПродажу1", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "1", "Barcode": "", "UnitType": "шт", "Quantity": "1", "Price": "1000", "Amount": "1000", "DiscountPrc": "", "DiscountSum": "", "VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "" }
```

```
VAT":"","IsExcise":false,"OtherParameters":null,"VATLetter":"","ExciseLetter":"","ExciseStampBarcode":"","DocNumber":"","VendorCode":"","Article":"","Name":"Товар Интернет Продажу2","CdUKDZED":"","CdDKPP":"","GoodsType":"1","Barcode":"","UnitType":"шт","Quantity":"2","Price":"1000","Amount":"2000","DiscountPrc":"","DiscountSum":"","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"","IsExcise":false,"OtherParameters":null,"VATLetter":"","ExciseLetter":"","ExciseStampBarcode":"","DocNumber":"","Comment":null,"DocType":"0","DocSubType":"0","SaleOrderNumber":"555444333555"}
```

JSONPayData =

```
{ "SumCash":null, "SumPayByCard":"1500", "SumPayByCredit":null, "SumPayByCertificate":null, "SumPayCheck":"3000", "CustomerEmail":null, "TerminalID":null, "ApprovalCode":null, "RRN":null, "IssuerName":null, "PAN":null, "TransactionDate":null, "SignVerif":null, "AcquireName":null, "InvoiceNumber":null, "ParentRRN":null, "PaymentOrderType":"3", "SumPreparePayed":"1500", "ParentReceiptFiscalNumber":null, "OtherParameters":null }
```

Важливо: Сума післяплати розраховується автоматично від різниці всіх позицій чеку, мінус сума передоплати.

9. Друк періодичних звітів

Для друку періодичних звітів, використовується функція

БУЛЕВО НапечататьПериодическийОтчетПоДатам (ИДУстройства, ВходныеПараметры, ВыходныеПараметры, Сокращенный)

Період це проміжок часу, що встановлюється між **DateBegin** та **DateEnd**.

Таблиця 10.1 Параметри функції поточного стану ПРРО.

Ім'я	Тип	Опис
ИДУстройства	String	Ідентифікатор пристрою
ВходныеПараметры	DATE	Початок періоду в форматі ДДММГГГГ
ВыходныеПараметры	DATE	Кінець періоду в форматі ДДММГГГГ
Сокращенный	BOOL	ОзнакаСкороченогоЗвіту

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення початку періоду та дату закінчення заданого періоду:

Таблиця 10.2 Параметри функції поточного стану ПРРО.

JsonVal	String	JSON-структура, що містить значення успішності операції, ОзнакаСкороченогоЗвіту, Base64 рядок XML структури чеку	
DeviceID	String	Ідентифікатор пристрою	ФіскальнийНомерПРРО
DateBegin	DATE	Дата початку періоду	dd.MM.yyyy
DateEnd	DATE	Дата кінця періоду	dd.MM.yyyy

IsShort	BOOL	Признак сокращенного отчета	TRUE/FALSE
---------	------	-----------------------------	------------

Приклад:

Функция осуществляет печать периодического отчёта по датам.

Функция НапечататьПериодическийОтчетПоДатам(ОбъектДрайвера, Параметры, ПараметрыПодключения, ВходныеПараметры, ВыходныеПараметры)

Результат = Истина;

ОбщегоНазначенияКлиентСервер.СообщитьПользователю(НСтр("ru='Печать периодического отчета по датам, с даты ' ;uk='Друк періодичного звіту по датах, з дати '" +

Формат(ВходныеПараметры[0], "ДФ=dd.MM.yyyy") +

НСтр("ru=' по дату ' ;uk=' за дату '" +

Формат(ВходныеПараметры[1], "ДФ=dd.MM.yyyy"));

Если ПараметрыПодключения.ТипОборудования = "ФискальныйРегистратор" Тогда

Ветка на стандарт 1.4. Поддержка команды драйвера

"НапечататьПериодическийОтчетПоДатам".

Если ПараметрыПодключения.РевизияИнтерфейса > 1004 Тогда

Попытка

Результат =

ОбъектДрайвера.НапечататьПериодическийОтчетПоДатам(ПараметрыПодключения.ИДУстройства, ВходныеПараметры[0],ВходныеПараметры[1],Результат);

Исключение

Результат = Ложь;

ВыходныеПараметры.Очистить();

ВыходныеПараметры.Добавить(999);

ВыходныеПараметры.Добавить(НСтр("ru='Ошибка вызова метода

<ОбъектДрайвера.НапечататьПериодическийОтчетПоДатам>.' ;uk='Помилка виклику методу

<ОбъектДрайвера.НапечататьПериодическийОтчетПоДатам>.'" + Символы.ПС +

ОписаниеОшибки());

Возврат Результат;

КонецПопытки;

КонецЕсли;

КонецЕсли;

Заполнение выходных параметров

ВыходныеПараметры.Добавить(1);

ВыходныеПараметры.Добавить(1);

ВыходныеПараметры.Добавить(0);

ВыходныеПараметры.Добавить(ТекущаяДата());

Возврат Результат;

КонецФункции

10. Отримання інформації про поточний стан ПРРО

Для отримання інформації про поточний стан використовується функція

БУЛЕВО ПолучитьТекущееСостояние (ИДУстройства, ВходныеПараметры, ВыходныеПараметры)

Параметри функції зазначені в Таблиці 11.1.

Таблица 11.1 Параметры функции поточного stanu ПРРО.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою
ВходныеПараметры	СТРОКА	Таблиця вхідних параметрів, параметр ігнорується.
ВыходныеПараметры	СТРОКА	XML-структура параметрів з поточними значеннями

Таблиця вихідних параметрів має вигляд XML-структури з переліком параметрів, які наведено в Таблиці 11.2., та їх значень.

Таблиця 11.2 Параметри XML-структури вихідних параметрів.

Ім'я параметру	Значення
CheckNumber	Локальний номер останнього чеку
CashBalance	Залишок готівки в касі
IsOfflineMode	Стан офлайн-режиму каси: 0 – каса в онлайн-режимі; 1 – каса в офлайн-режимі
NextLocalNumber	Наступний локальний номер чеку
LastCheckFiscalNumber	Фіскальний номер останнього чеку
ShiftNumber	Номер поточної відкритої зміни/ останньої закритої зміни
ShiftID	Ідентифікатор зміни
ShiftFiscalNumber	Фіскальний номер зміни
ShiftState	Стан зміни: 1 – закрита, 2 – відкрита.
DateTime	Дата і час формування фіскального документу

Приклад:

ИДУстройства = "4000001234";

ВыходныеПараметры = "";

Результат = *ОбъектДрайвера.ПолучитьТекущееСостояние (ИДУстройства, "" ,ВыходныеПараметры)*;

Приклад вихідних параметрів:

ВыходныеПараметры =

```
<?xml version = "1.0" encoding= "UTF-8" ?>< DocumentOutputParameters > <
Parameters CheckNumber = "0" CashBalance = "0" IsOfflineMode = "0"
NextLocalNumber = "2472" LastCheckFiscalNumber = "" ShiftNumber= "2471" ShiftID =
"96acfaa1-22bd-4d6c-9f42-b5dbe958f6d1" ShiftFiscalNumber = "510130" ShiftState=
"2" DateTime= "0001-01-01T00:00:00" /> </ DocumentOutputParameters >
```

11. Друк довільної текстової інформації

Для друку на чекову стрічку довільного тексту використовується функція

БУЛЕВО *НапечататьТекстовыйДокумент(ИДУстройства, Текст)*

Після виклику відображається вікно з зображенням тексту і можливістю роздрукувати.

У випадку, коли задано автоматичну відправку чеків на друк і принтер для друку за замовчуванням, друк заданого тексту відбувається без попереднього перегляду:

УстановитьПараметр("AUTOPRINTMODE", "true");

УстановитьПараметр("DEFAULTPRINTERNAME", "xprint80-name");

Функцію можливо викликати без відкриття касової зміни, але необхідно авторизуватися.

Параметри функції зазначені в Таблиці 12.1.

Таблиця 12.1 Параметри функції поточного стану ПРРО.

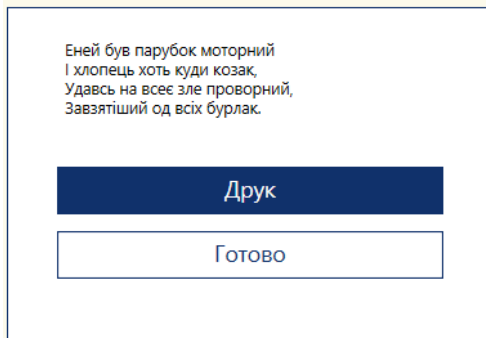
Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою
Текст	СТРОКА	Довільний текст, який передається на друк.

Приклад:

ИДУстройства = "4000001234";

Текст = "Еней був парубок моторний"+Символы.ПС+"І хлопець хоть куди козак,"+Символы.ПС+"Удавьсь на всеє зле проворний,"+Символы.ПС+"Завзятіший од всіх бурлак."";

Результат = *ОбъектДрайвера.НапечататьТекстовыйДокумент (ИДУстройства , Текст)*;



12. Повернення товару

Виконання операції повернення товару відбувається аналогічно операції формування чеку (див. Реєстрація чеків), але із зазначенням параметрів

Параметри функції зазначені в Таблиці 13.1.

Таблиця 13.1 Параметри функції службового внесення.

Ім'я параметру	Значення
ЧекВозврата	True
ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат	Фіскальний номер чеку, за яким виконується повернення

Приклад:

ИмяПараметра = "DeviceIDFnRRO";

ЗначениеПараметра = "4000001234";

Ответ = *ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра)*;

ИмяПараметра = "PathToCashalotDir";

ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";

Ответ = *ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра)*;

ЧекВозврата = *ВходныеПараметры[1]*;

ФискальныйЧек = *ВходныеПараметры[0]*;

Результат = *ОткрытьЧек(ОбъектДрайвера, ВходныеПараметры, ПараметрыПодключения,*

ЧекВозврата, ФискальныйЧек, ВыходныеПараметры);

ИДУстройства = "4000001234";

Наименование = "Товар1";

КодУКТЗЕД = "000001";

Артикул = "артикул1";

Штрихкод = "9999999999";

ЕдиницаИзмерения = "шт";

НомерСекции = 1;

Количество = 5;
Цена = 50,45;
Сумма = 239,64;
ПроцентСкидки = 5;
СуммаСкидки = 12,61;
НДС = 20;
ПризнакСуммаВключаетНДС = 1;
СуммаНДС = 50,45;
ПризнакАкцизногоНалога = 0;
Ответ = ОбъектДрайвера.НапечататьФискСтроку(*ИДУстройства, Наименование, Количество, Цена, Сумма, НомерСекции, СтавкаНДС, ДопРеквизиты*);
СуммаНаличнойОплаты = 239,60;
СуммаБезналичнойОплаты1 = 0;
СуммаБезналичнойОплаты2 = 0;
СуммаБезналичнойОплаты3 = 0;
Ответ = ОбъектДрайвера.ЗакрытьЧек(*ИДУстройства, СуммаНаличнойОплаты, СуммаБезналичнойОплаты1, СуммаБезналичнойОплаты2, СуммаБезналичнойОплаты3, СуммаЧека, ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат, НомерСозданногоЧека, ФискальныйНомерСозданногоЧека*);

13. Службове внесення

Для виконання операції службового внесення використовується функція

БУЛЕВО НапечататьЧекВнесенияВыемки (**ИДУстройства, Сумма**)

Параметри функції зазначені в Таблиці 14.1.

Таблица 14.1 Параметри функції службового внесення.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою
Сумма	ЧИСЛО	Сума внесення у форматі <Грн, Коп>

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра, ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра, ЗначениеПараметра*);
Результат = ОбъектДрайвера.НапечататьЧекВнесенияВыемки
 (*ПараметрыПодключения.ИДУстройства, ?(ТипИнкассации = 1, Сумма, -Сумма)*);

14. Службова видача

Для виконання операції службової видачі використовується функція

БУЛЕВО НапечататьЧекВнесенияВыемки (**ИДУстройства, Сумма**)

з від'ємним значенням **Сумма** (див. розділ Службове внесення).

Параметри функції зазначені в Таблиці 15.1.

Таблица 15.1 Параметри функції службового внесення.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою
Сумма	ЧИСЛО	(від'ємна) сума видачі у форматі <Грн, Коп>

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Результат = ОбъектДрайвера.НапечататьЧекВнесенияВыемки
(*ПараметрыПодключения*.ИДУстройства, ?(*ТипИнкассации* = 0, *Сумма*, -*Сумма*));

15. Формування X-звіту

Для виконання операції формування звіту без гасіння (X-звіту) використовується функція **БУЛЕВО НапечататьОтчетБезГашения(ИДУстройства)**

Параметри функції зазначені в Таблиці 16.1.

Таблиця 16.1 Параметри функції X-звіту.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Результат =
ОбъектДрайвера.НапечататьОтчетБезГашения(*ПараметрыПодключения*.ИДУстройства);

16. Формування Z-звіту

Для формування Z-звіту використовується функція

БУЛЕВО НапечататьОтчетСГашением(ИДУстройства)

Параметри функції зазначені в Таблиці 17.1.

Таблиця 17.1 Параметри функції Z-звіту.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Результат = ОбъектДрайвера.
НапечататьОтчетСГашением(*ПараметрыПодключения*.ИДУстройства);

17. Закриття зміни

Для виконання операції службового внесення використовується функція

БУЛЕВО ЗакрытьСмену(ИДУстройства)

Параметри функції зазначені в Таблиці 18.1.

Таблиця 18.1 Параметри функції закриття зміни.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Приклад:

ИмяПараметра = "DeviceIDFnRRO";

ЗначениеПараметра = "4000001234";

Ответ = *ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);*

ИмяПараметра = "PathToCashalotDir";

ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";

Ответ = *ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);*

Результат = *ЗакрытьСмену(ОбъектДрайвера, Параметры, ПараметрыПодключения, ВходныеПараметры, ВыходныеПараметры);*

Порядок виклику функцій при виконанні касових операцій

При роботі з касою в системі BAS рекомендується послідовність виклику функцій ПРРО Cashalot, зазначена в Таблиці 18.2

Таблиця 18.2 Порядок виклику функцій при роботі з касою.

Касова операція	Послідовність функцій драйверу
Відкриття зміни	УстановитьПараметр ОткрытьСмену
Фіскалізація чеків	УстановитьПараметр ОткрытьЧек НапечататьФискальнуюСтроку або НапечататьФискСтроку ЗакрытьЧек або ОтменитьЧек
Повернення товару	УстановитьПараметр ОткрытьЧек НапечататьФискальнуюСтроку або НапечататьФискСтроку ЗакрытьЧек або ОтменитьЧек
Х-звіт	УстановитьПараметр НапечататьОтчетБезГашения
Z-звіт	УстановитьПараметр НапечататьОтчетСГашением
Закриття зміни	УстановитьПараметр ЗакрытьСмену
Внесення грошей	УстановитьПараметр НапечататьЧекВнесенияВыемки
Видача грошей	УстановитьПараметр НапечататьЧекВнесенияВыемки
Службові операції	ПолучитьОшибку або ПолучитьПараметры або ТестУстройства

Приклад реалізації підключення драйверу до Розрахунково-облікової платформи (.erf)

&НаКлиенте

Процедура КоманднаяПанельТаблицаДокументовМодосЗапуск (Команда)

Если

ПодключитьВнешнююКомпоненту ("C:\Users\User1\Cashalot\CashaLotApiDrv32.dll", "CashaLotApiDrv") Тогда

ОбъектДрайвера = Новый ("AddIn.CashaLotApiDrv.CashaLotApiDrv");

```

ИмяПараметра = "PathToCashalotDir";
Параметр = " C:\Users\User1\Cashalot\CashaLotApiDrv32.dll";

УстановленПутьCashalot = ОбъектДрайвера.УстановитьПараметр (ИмяПараметра,
Параметр) ;
Сообщить (" Установлен Путь Cashalot : " + УстановленПутьCashalot) ;

ИмяПараметра = "DeviceIDFnRRO";
Параметр = "4000007619";

ФН = ОбъектДрайвера.УстановитьПараметр (ИмяПараметра, Параметр) ;

Сообщить (" Установлен Номер Фискальной Кассы: " + ФН) ;
таблицапарам = "";
ответ = ОбъектДрайвера.ПолучитьПараметры (таблицапарам) ;
Сообщить (" Параметры: " + таблицапарам) ;

ответ = ОбъектДрайвера.Подключить ("4000007619") ;
Сообщить (" Подключить: " + ответ) ;

ответ = ОбъектДрайвера.ОткрытьСмену ("4000007619") ;
Сообщить (" ОткрытьСмену: " + ответ) ;

//службове внесения
ответ = ОбъектДрайвера.НапечататьЧекВнесенияВыемки ("4000007619", 50) ;
Сообщить (" Службове внесения: 50 грн " + ответ) ;

ФискальныйНомерКассы ="4000007619";
НомерСмены = 0;
НомерЧека = 0;
НомерЧекаВозврата = 0;
ФискальныйЧек = Истина;
ТипЧека = 0;

ответ = ОбъектДрайвера.ОткрытьЧек (ФискальныйНомерКассы, ФискальныйЧек, ТипЧека,
Истина, НомерЧека, НомерСмены) ;
Сообщить ("===== Відкрито чек ===== " + ответ ) ;
Сообщить ("ФискальныйНомерКассы " + ФискальныйНомерКассы ) ;
Сообщить ("НомерЧека " + НомерЧека ) ;
Сообщить ("НомерСмены " + НомерСмены ) ;

ИДУстройства = "4000007619";
Наименование = "НазваТовару1";
КодУКТЗЕД = "";
Артикул = "АртикулТовару1";
Штрихкод = "123456789";
ЕдиницаИзмерения = "шт";
НомерСекции = "";
Количество = "1";
Цена = "183";
Сумма = "181,17";
ПроцентСкидки = "";
СуммаСкидки = "1.83";
НДС = "20";

```

```

ПризнакСуммаВключаетНДС      = "true";
СуммаНДС                      = "30.2";
ПризнакАкцизногоНалога      = "false";
НомерСтроки                   = "1";
ДопРеквизиты                 = "";

ответ = ОбъектДрайвера.НапечататьФискальнуюСтроку
(ИДУстройства, Наименование, КодУКТЗЕД, Артикул, Штрихкод, ЕдиницаИзмерения, НомерСекции,
Количество, Цена, Сумма, ПроцентСкидки, СуммаСкидки, НДС, ПризнакСуммаВключаетНДС, СуммаНДС,
ПризнакАкцизногоНалога, НомерСтроки, ДопРеквизиты);
Сообщить ("===== фискальный рядок 1 ===== " + ответ );
ответ = ОбъектДрайвера.НапечататьФискальнуюСтроку
(ИДУстройства, Наименование, КодУКТЗЕД, Артикул, Штрихкод, ЕдиницаИзмерения, НомерСекции,
Количество, Цена, Сумма, ПроцентСкидки, СуммаСкидки, НДС, ПризнакСуммаВключаетНДС, СуммаНДС,
ПризнакАкцизногоНалога, НомерСтроки, ДопРеквизиты);
Сообщить ("===== фискальный рядок 2 ===== " + ответ );

НаличнаяОплата              = "300";
ОплатаКартой                  = "10";
ОплатаКредитом                = "20";
ОплатаСертификатом           = "37";
СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления = "362,34";
ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат = "";
НомерЧекаЧисловой            = "";
ФискальныйНомерСозданногоЧека = "";

ответ =
ОбъектДрайвера.PrintNoFiscString(ИДУстройства, "#####
##" + Символы.ПС+"#Тестовий коментар#"
+Символы.ПС+"#####");
Сообщить ("===== Нефискальні рядки ===== " + ответ );

ответ =
ОбъектДрайвера.ЗакретьЧек(ИДУстройства, НаличнаяОплата, ОплатаКартой, ОплатаКредитом,
ОплатаСертификатом, СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления, Фиск
альныйНомерЧекаПоКоторомуПроизводитсяВозврат, НомерЧекаЧисловой, ФискальныйНомерСозда
нногоЧека);
Сообщить ("===== Закрито чек ===== " + ответ );
Сообщить ("ФН чека " + ФискальныйНомерСозданногоЧека );
Сообщить ("локальний номер чека " + НомерЧекаЧисловой );

ответ = ОбъектДрайвера.НапечататьОтчетБезГашения(ИДУстройства);
Сообщить ("===== X-звіт ===== " + ответ );

//службова видача
ответ = ОбъектДрайвера.НапечататьЧекВнесенияВыемки ("4000007619", -20);
Сообщить (" Службова видача: 20 грн " + ответ);

ответ = ОбъектДрайвера.ЗакретьСмену("4000007619");
Сообщить ("ЗакретьСмену: " + ответ );

КонецЕсли;

КонецПроцедуры

```


Варіант 2а – API ПРРО Cashalot з COM інтерфейсом CashalotApi32.dll, CashalotApi64.dll

В останньому оновленні з'явилась можливість використання **декількох екземплярів ПРРО Cashalot**, що значно розширив перелік варіантів використовуватися для вже існуючих параметрів та функцій.

Для зручності користувачів та запобіганню дублюванню ключових назв в вже існуючих варіантів підключення та використання нових компонентів та бібліотек, додано поняття **Обов'язкові параметри***.

В склад яких можуть входити також додаткові залежності від призначення функції входять додаткові **Обов'язкові параметри***, що можуть бути не обов'язковими для інших функцій. Обов'язкові параметри* не залежать від Інтерфейс користувача та повинні бути використані окремо в випадку з Одним екземпляром ПРРО **SetParameter**.

Для використання розрахунково-обліковою платформою або системою декількох екземплярів ПРРО, використовується параметр **SetParameterEx**.

Використовують бібліотеки елементів інтерфейсу **CashalotApi**, що розміщені в кореневих каталогах шлях до яких необхідно вказати при встановленні **підключення до об'єктів COM, кожного екземпляру ПРРО Cashalot окремо**. Шлях до кореневих каталогах є **Обов'язковим параметри* PathToCashalotDir***. (див. Встановлення налаштувань бібліотеки).

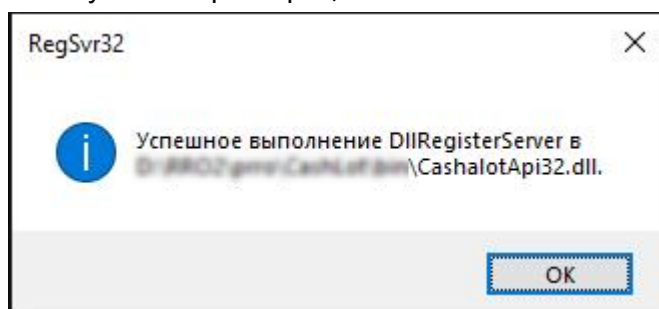
```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApi32.dll  
(для 32-бітної платформи)
```

```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApi64.dll  
(для 64-бітної платформи)
```

Наприклад:

```
regsvr32 C:\Users\User1\AppData\Local\Cashalot\CashalotApi32.dll
```

Після успішної реєстрації бібліотеки з'явиться повідомлення зразку:



В бібліотеках CashalotApi реалізована обробка наступних функцій:

1. **SetParameter**
2. **SetParameterEx**
3. **GetVersion**
4. **GetInterfaceRevision**
5. **ServiceInput**
6. **ServiceInputEx**
7. **ServiceOutput**
8. **ServiceOutputEx**
9. **OpenShift**
10. **FiscalizeCheck**
11. **FiscalizeReturnCheck**

- 12. **GetReceiptLocalNumberByFiscalNumber**
- 13. **GetReceiptFiscalNumberByLocalNumber**
- 14. **GetReceiptXML**
- 15. **GetFirstReceiptByPeriod**
- 16. **GetFirstReceiptByShift**
- 17. **GetNextReceipt**
- 18. **GetFirstZReportByPeriod**
- 19. **GetNextZReport**
- 20. **GetZReportByShift**
- 21. **ShowReceipt**
- 22. **GetLastError**
- 23. **PrintXReport**
- 24. **GetXReport**
- 25. **GetLastZReport**
- 26. **PrintZReport**
- 27. **CloseShift**

Функції, виділені жирним шрифтом, повертають відповідь у вигляді структури

CashalotApiRetVal, що може містити такі властивості:

Властивість	Тип	Значення
Description	STRING	Опис
JsonVal	STRING	JSON-структура
ReceiptFiscalNum	STRING	Фіскальний номер чеку
ReceiptLocalNum	STRING	Локальний номер чеку
Return	BOOL	Успішність операції
ShiftID	STRING	Ідентифікатор касової зміни
XML	STRING	XML-структура документу
Type	INTEGER	Тип документа (див. FiscalizeCheck())

Приклади використання COM-функцій наведено у вигляді

```
Ret = App.GetVersion()
```

1. Встановлення налаштувань API бібліотеки SetParameter

SetParameter щодо встановлення налаштувань підключення API бібліотек Функція призначена для встановлення значень параметрів налаштувань API бібліотеки. Рекомендовано встановити всі параметри на початку роботи.

Параметри

Ім'я	Тип	Опис
Name	STRING	Ім'я параметра
Value	STRING	Значення параметра

Значення Name

Name	Тип	Опис
------	-----	------

DeviceIDFnRRO	STRING	Фіскальний номер ПРРО * ²
PathToCashalotDir	STRING	Шлях до каталогу з встановленим ПРРО Cashalot * ³
NOINTERFACEMODE	STRING	Ознака використання інтерфейсу РМК для авторизації та службових операцій видачі та внесення готівки в касу. "False" – при роботі з АРІ викликається діалогове вікно. "True" – авторизація відбувається шляхом передачі параметрів авторизації PathToCertificate та PwdToCertificate, вікна службових операцій видачі та внесення готівки не викликаються (проте, результат операції службового внесення/видачі готівки відображається).
NOAUTOOPENSIFT	STRING	Перевірка стану зміни (відкрита або закрита). Якщо зміна закрита при виклику будь-якої СОМ-функції, зміна автоматично відкривається. "False" – автоматично відкривати зміну, якщо вона закрита. "True" – не відкривати автоматично зміну. В цьому випадку робота СОМ-функції буде перервана і результат виконання буде повернуто у вигляді помилки.
USETOKEN	STRING	Чи використовується токен для авторизації. "True" – використовується "False" – не використовується (використовується файловий ключ) Зауваження. Параметр USETOKEN має бути заданий після параметрів PathToCashalotDir, DeviceIDFnRRO та NOINTERFACEMODE. Для пришвидшення завантаження програми рекомендовано не використовувати токени.
PathToCertificate	STRING	Шлях до папки з файловим ключем користувача. Зауваження. В папці повинні міститися ключ і сертифікат тільки одного користувача. Якщо касою користуються декілька касирів, необхідно помістити їх сертифікати і ключі в окремі папки.
PwdToCertificate	STRING	Пароль користувача
DEFAULTPRINTERNAME	STRING	Визначає принтер для друку при натисканні кнопки "Друк" в діалогових вікнах: 1) "" – пустий рядок, виводиться діалог вибору принтера. 2) "DEFAULTWINDOWS" – використовувати принтер Windows за замовчуванням 3) "ім'я принтера чеків" - в параметр передається ім'я принтера чеків, який буде використовуватись для друку чеків в ПРРО Кашалот

NEW

²

³ Обов'язкові параметри

AUTOPRINTMODE	STRING	Параметр визначає автоматичну відправку на друк чеку або службового звіту без відображення інформаційного вікна "True" – відправляти на друк "False" – не відправляти на друк Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), то чек одразу друкується, в іншому випадку відображається вікно вибору принтера для друку.
NOPRINTMODE	STRING	Дозвіл на друк чеку або службового звіту в режимі AUTOPRINTMODE = "True" "True" – заборонити друк "False" – дозволити друк
OtherParametrs	STRING	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.

Відповідь

	BOOL	Успішність операції
--	------	---------------------

Приклад:

```

App = CreateObject("AddIn.CashaLotApi")

param = "PathToCashalotDir"
paramVal = "C:\Users\User\AppData\Roaming\Cashalot"
ret =App.SetParameter(param, paramVal)
param = "DeviceIDFnRRO"
paramVal = "4000007621"
ret = App.SetParameter(param, paramVal)
param = "NOINTERFACEMODE"
paramVal = "True"
ret = App.SetParameter(param, paramVal)
param = "NOAUTOOPENSIFT"
paramVal = "True"
ret = App.SetParameter(param, paramVal)
param = "USETOKEN"
paramVal = "False"
ret = App.SetParameter(param, paramVal)
param = "PathToCertificate"
paramVal = "C:\Users\User\Certificate\Cashier1\"
ret = App.SetParameter(param, paramVal)
param = "PwdToCertificate"
paramVal = "UseOn!yStr0ngPa$$words"
ret = App.SetParameter(param, paramVal)
param = "DEFAULTPRINTERNAME"
paramVal = "Microsoft Print to PDF"
ret = App.SetParameter(param, paramVal)
param = "AUTOPRINTMODE"

```

```

paramVal = "True"
ret = App.SetParameter(param, paramVal)
param = "NOPRINTMODE"
paramVal = "False"
ret = App.SetParameter(param, paramVal)

```

Приклад відповіді:

```
ret = True;
```

NEW

2. Встановлення налаштувань API бібліотеки SetParameterEx

Функція призначена для використання розрахунково-обліковою платформою або системою **декількох екземплярів ПРРО одночасно**. Перед початком роботи необхідно задати всі обов'язкові параметри. Якщо використовується робота одночасно з декількома ПРРО то встановлення параметрів здійснюється виключно через **SetParameterEx**. У такому випадку **SetParameter** не використовується.

При встановленні параметрів за допомогою **SetParameterEx**, **обов'язково потрібно зазначити шлях до кореневого каталогу кожного екземпляру ПРРО Cashalot та ФН номер каси окремо**.

Обов'язково необхідно створити для кожного екземпляру ПРРО окремо каталог та вказати шлях до кожного екземпляру індивідуально. За потреби вкласти в цей каталог сертифікати користувачів. Власне таким чином може бути реалізовано розподілення доступу між користувачами на основі вкладених індивідуально ключів для кожного каталогу окремо.

Приклад вказаний нижче;

Параметри

Ім'я	Тип	Опис
Name	STRING	Ім'я параметра
Value	STRING	Значення параметра

Значення Name

Name	Тип	Опис
DeviceIDFnRRO	STRING	Фіскальний номер ПРРО *
PathToCashalotDir	STRING	Шлях до каталогу з встановленим ПРРО Cashalot *
NOINTERFACEMODE	STRING	Ознака використання інтерфейсу РМК для авторизації та службових операцій видачі та внесення готівки в касу. "False" – при роботі з API викликається діалогове вікно. "True" – авторизація відбувається шляхом передачі параметрів авторизації PathToCertificate та PwdToCertificate, вікна службових операцій видачі та внесення готівки не викликаються (проте, результат операції службового внесення/видачі готівки відображається).
NOAUTOOPENSIFT	STRING	Перевірка стану зміни (відкрита або закрита). Якщо зміна закрита при виклику будь-якої СОМ-функції, зміна автоматично відкривається. "False" – автоматично відкривати зміну, якщо вона закрита.

		"True" – не відкривати автоматично зміну. В цьому випадку робота COM-функції буде перервана і результат виконання буде повернуто у вигляді помилки.
USETOKEN	STRING	Чи використовується токен для авторизації. "True" – використовується "False" – не використовується (використовується файловий ключ) Зауваження. Параметр USETOKEN має бути заданий після параметрів PathToCashalotDir, DeviceIDFnRRO та NOINTERFACEMODE. Для пришвидшення завантаження програми рекомендовано не використовувати токени.
PathToCertificate	STRING	Шлях до папки з файловим ключем користувача. Зауваження. В папці повинні міститися ключ і сертифікат тільки одного користувача. Якщо касою користуються декілька касирів, необхідно помістити їх сертифікати і ключі в окремі папки.
PwdToCertificate	STRING	Пароль користувача
DEFAULTPRINTERNAME	STRING	Визначає принтер для друку при натисканні кнопки "Друк" в діалогових вікнах: 1) "" – пустий рядок, виводиться діалог вибору принтера. 2) "DEFAULTWINDOWS" – використовувати принтер Windows за замовчуванням 3) "ім'я принтера чеків" - в параметр передається ім'я принтера чеків, який буде використовуватись для друку чеків в ПРРО Кашалот
AUTOPRINTMODE	STRING	Параметр визначає автоматичну відправку на друк чеку або службового звіту без відображення інформаційного вікна "True" – відправляти на друк "False" – не відправляти на друк Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), то чек одразу друкується, в іншому випадку відображається вікно вибору принтера для друку.
NOPRINTMODE	STRING	Дозвіл на друк чеку або службового звіту в режимі AUTOPRINTMODE = "True" "True" – заборонити друк "False" – дозволити друк
OtherParametrs	STRING	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.

Відповідь

	BOOL	Успішність операції
--	------	---------------------

Приклад:

```
Set App = CreateObject("AddIn.CashaLotApi")
```

```

App.SetParameterEx " FiscalNumberRRO1","PathToCashalotDir",
"C:\Users\User\AppData\Roaming\Cashalot1\"
App.SetParameterEx " XXXXXXXXXXXX ", "DeviceIDFnRRO", " XXXXXXXXXXXX "
set ret1 = App.OpenShift ("XXXXXXXXXXXX")
MsgBox "resultOpenShift1.Json" & ret1.JsonVal

App.SetParameterEx " FiscalNumberRRO2","PathToCashalotDir",
"C:\Users\User\AppData\Roaming\Cashalot2\"
App.SetParameterEx " YYYYYYYYYY ", "DeviceIDFnRRO", " YYYYYYYYYY "
set ret2 = App.OpenShift ("YYYYYYYYYYY ")
MsgBox "resultOpenShift2.Json" & ret2.JsonVal
Set ret3 = App.CloseShift ("XXXXXXXXXXXX ")
MsgBox ret3.JsonVal
set ret2 = App.CloseShift ("YYYYYYYYYYY ")
MsgBox ret4.JsonVal

```

Приклад відповіді:

```
ret = True;
```

3. GetVersion – Отримання номеру версії API бібліотеки

Параметри

Функція не має вхідних параметрів.

Відповідь

	STRING	Номер версії API бібліотеки
--	--------	-----------------------------

Приклад:

```

App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetVersion()

```

Приклад відповіді:

```
Ret = "1,1";
```

4. GetInterfaceRevision – Отримання номеру ревізії API бібліотеки

Параметри

Функція не має вхідних параметрів.

Відповідь

	STRING	Номер ревізії API бібліотеки
--	--------	------------------------------

Приклад:

```

App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetInterfaceRevision ()

```

Приклад відповіді:

```
Ret = "3003";
```

5. OpenShift – Відкриття зміни

При виклику цієї функції API бібліотеки ПРРО Cashalot відкриває вікно авторизації, в якому потрібно зазначити папку з файловим ключем касира або старшого касира, ввести пароль, після чого відбудеться відкриття касової зміни.

Параметри

Ім'я	Тип	Опис
------	-----	------

FiscalNumberRRO	STRING	Ідентифікатор пристрою
-----------------	--------	------------------------

Для відкриття зміни в "тихому" режимі без відображення вікна авторизації необхідно за допомогою функції **SetParameter** встановити параметр
NOINTERFACEMODE = True

При виклику функцій, якщо зміна не відкрита, відбувається автоматичне відкриття зміни з викликом функції **OpenShift**. Для запобігання автоматичного відкриття зміни за допомогою функції **SetParameter** встановіть параметр

NOAUTOOPENSIFT= True

В цьому випадку, поки не буде явно відкрито зміну викликом **OpenShift**, виклик інших функцій, які вимагають відкритої зміни, буде перериватись з поверненням помилки:

`"ErrorString": "Операція не можлива, необхідно відкрити зміну"`

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	STRING	JSON-структура, що містить значення успішності операції, помилки, ID зміни
Return	BOOL	Успішність операції
ShiftID	STRING	Ідентифікаційний номер поточної зміни

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
Ret = App.OpenShift("4000007621")
```

Приклад відповіді:

```
Ret.Description = ""
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"1a9eb031-b1ce-4ee7-8a11-329824d895a7"}}
Ret.Return = True
Ret.ShiftID = 1a9eb031-b1ce-4ee7-8a11-329824d895a7
```

або:

```
Ret.Description = "Авторизація відмінена користувачем"
Ret.JsonVal = {"Ret":false,"ErrorString":"Авторизація відмінена користувачем", "Values":{}}
Ret.Return = False
Ret.ShiftID = ""
```

6. FiscalizeCheck – Фіскалізація чека продажу товарів, послуг

Для створення та фіскалізації чеку продажу товарів та/або послуг необхідно сформувати JSON структуру **JSONGoodsLstData**, що міститиме дані номенклатури, та структуру JSON структуру **JSONPayData** з формами оплати

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	STRING	Ідентифікатор пристрою
strJSONData	STRING	JSON структура з описом проданих товарів або послуг
strJSONPayData	STRING	JSON структура з підсумками оплати по чеку

Структура **JSONGoodsLstData** містить масив ReceiptLst, кожен елемент якого містить параметри номенклатури, та об'єкти Comment і DocType:

Поле	Значення
VendorCode*	Артикул
Name	Найменування товару
DocNumber**	Номер договору (при використанні ломбардних послуг)
CdUKDZED	Код УКТЗЕД
CdDKPP	Код ДКПП
GoodsType	Тип номенклатури: 1 – товар 2 – послуга 3 – ломбардна послуга ** За замовчуванням товар, якщо параметр не задано.
Barcode	Штрих-код
UnitType*	Одиниця вимірювання
Quantity*	Кількість товару
Price*	Ціна одиниці товару (в форматі <Гривні, Копійки>)
Amount*	Кінцева сума по позиції чека (з врахуванням всіх націнок та знижок, у форматі <Гривні, Копійки>)
DiscountPrc	Процент знижки (ігнорується, якщо заповнена сума знижки)
DiscountSum	Сума знижки на позицію
VATRate	ПДВ в процентах (від 0 до 99,99)
IsPriceIncludeVAT	Ознака, що ціна враховує ПДВ
SumVAT	ПДВ в процентах (від 0 до 99,99)
PaymentOrderType	Спосіб оплати: 0 - звичайна 1 - передоплата 2 - післяплата 3 - інтернет-продаж
SumPreparePayed	Сума передоплати
ParentReceiptFiscalNumber	Фіскальний номер чеку по якому здійснюється доплата/повернення
SaleOrderNumber	Номер замовлення/договору
IsExcise	Ознака підакцизного товару

NEW

NEW
NEW

NEW

ExciseStampBarcode	Штрих-код марки акцизного податку. Поле ігнорується, якщо ознака підакцизного товару IsExcise = false.
OtherParametrs	Додаткові параметри у вигляді XML таблиці

*** обов'язкові параметри при передачі даних товару.**

** Ломбардні операції описані в розділі "[Ломбардні послуги](#)".

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

Параметр **OtherParameters** передається в форматі XML з кодуванням UTF-8. Атрибути XML-рядка вказані в таблиці нижче.

Зазначимо, що параметри XML-рядка OtherParameters мають пріоритет над параметрами, які явно передані в ReceiptLst.

Атрибути XML-рядка OtherParameters

XML атрибут	Опис атрибута
NOMENCLATURE *	Найменування товару
UKTZED	Код УКТЗЕД
VENDORCODE *	Артикул
BARCODE	Штрих-код
DIMENSION *	Одиниця вимірювання
ParentReceiptFiscalNumber	Фіскальний номер чеку по якому здійснюється доплата/повернення
SaleOrderNumber	Номер замовлення/договору
SumPreparePayed	Вказуємо суму передоплати, вказуємо загальну суму по чеку.
DISCOUNT	Процент знижки (ігнорується, якщо заповнена сума знижки)
DISCOUNTAMOUNT	Сума знижки
VAT *	Ставка ПДВ в процентах
PRICEINCLUDESVAT або PRICEINCLUDEVAT	Ознака, що ціна враховує ПДВ
AMOUNTVAT	Сума ПДВ, у форматі <Гривні, Копійки>
ISEXCISABLE	Ознака підакцизного товару, True/False

* обов'язкові параметри в рядку OtherParameters, якщо не вказані відповідні явні параметри. Опціональний об'єкт Comment використовується для додавання нефіскальної інформації:

Comment	Опціональний рядок нефіскальної інформації, коментар. Допускається використання одного коментарю на чек. За необхідності сформулювати коментар з декількох рядків, для переходу на новий рядок використовується комбінація "\r" або "\n".
---------	---

Об'єкт DocType є обов'язковим при проведенні ломбардних операцій, призначений для визначення типу документа (чека). Ломбардні операції описані в розділі ["Ломбардні послуги"](#).

DocType	Тип документа 0 – чек продажу 4 – ломбардний чек За замовчуванням DocType = 0, якщо об'єкт не задано.
---------	--

Структура **JSONPayData** містить підсумки оплати по чеку, і має наступні поля:

Поле	Значення
SumCash	Сума оплати чеку готівкою
SumPayByCard	Сума оплати чеку банківською картою
SumPayByCredit	Сума оплати чеку в кредит
SumPayByCertificate	Сума оплати чеку сертифікатом
SumPayCheck	Сума чеку до оплати, з урахуванням всіх надбавок і знижок
SumPreparePayed	Сума попередньої оплати
CustomerEmail	Опціонально. Електронна адреса покупця, на яку можна відправити фіскалізований чек, передається в поле електронної адреси при натисканні кнопки "Надіслати чек на Email" екрану відображення чеку. Рекомендовано використовувати при встановленому параметрі відображення вікон інтерфейсу: SetParameter("NOINTERFACEMODE","False")
Необов'язкові параметри інформації по транзакції терміналу при безготівкових розрахунках:	
TerminalID	Код терміналу
ApprovalCode	Код підтвердження
RRN	PPH транзакції
IssuerName	Платіжна система
PAN	Номер картки
TransactionDate	Дата транзакції (в форматі dd.MM.yyyy HH:mm:ss)
SignVerif	Електронний підпис
AcquireName	Термінал

InvoiceNumber	Номер чеку
ParentRRN	PPH транзакції чека по якому здійснено повернення

Увага! Різниця між SumPayCheck та сумою по всіх видах оплати вважається значенням заокруглення. Тому розробник має самостійно реалізувати алгоритм заокруглення значення виданої готівки або суми оплати по чеку.

Відповідь

В якості відповіді повертається структура CashalotApiRetVal, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документу (див. нижче)
XML	String	XML-структура фіскалізованого чеку

Тип документу "*Type*" визначається числом, яке формується наступним чином:

1<Клас документу><Тип документу><Додатковий тип документу>

Де

Число	Кількість знаків	Значення
<Клас документу>	1	0 - Чек 1 - Z-звіт 2 - X-звіт
<Тип документу>	3	000 - Чек реалізації товарів/послуг 001 - Чек переказу коштів 002 - Чек операції обміну валюти 003 - Чек видачі готівки 100 - Відкриття зміни 101 - Закриття зміни 102 - Початок офлайн сесії 103 - Завершення офлайн сесії
<Додатковий тип документу>	3	000 - Касовий чек (реалізація) 001 - Видатковий чек (повернення) 002 - Чек операції «службове внесення»/«отримання авансу» 003 - Чек операції «отримання підкріплення» 004 - Чек операції «служба видача»/«інкасація» 005 - Чек сторнування попереднього чека

Приклад:

```
JSONGoodsLstData =
{"ReceiptLst": [{"VendorCode": "АртикулТовару1", "Name":
"НазваТовару1", "CdUKDZED": "", "CdDKPP": "",
"GoodsType": "", "Barcode": "123456789", "UnitType": "шт",
"Quantity": "1", "Price": "183", "Amount": "181,17",
"DiscountPrc": "", "DiscountSum": "1.83",
"VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "30.2",
"IsExcise": false, "OtherParamtrs": null}, {"VendorCode":
"АртикулТовару2", "Name": "НазваТовару2", "CdUKDZED": "",
"CdDKPP": "", "GoodsType": "", "Barcode": "999999999",
"UnitType": "шт", "Quantity": "1", "Price": "183",
"Amount": "181,17", "DiscountPrc": "",
"DiscountSum": "1.83", "VATRate": "20",
"IsPriceIncludeVAT": true, "SumVAT": "30.2", "IsExcise": true,
"ExciseStampBarcode": "1111",
"OtherParamtrs": null}], "Comment": "Тестовий коментар"}

JSONPayData =
{"SumCash": "52", "SumPayByCard": "310,34", "SumPayByCredit": n
ull, "SumPayByCertificate": null, "SumPayCheck": "362,34", "Termina
lID": "S1K700J8", "ApprovalCode": "68014B", "RRN":
"048867173620", "IssuerName": "MASTERCARD", "PAN":
"XXXXXXXXXXXX6883", "TransactionDate": "" + Strings.Format(Now,
"dd.MM.yyyy HH:mm:ss") + "", "AcquireName": "Кашалот
банк", "InvoiceNumber":
"224", "CustomerEmail": "MyFirstRegular.customer@gmail.com"}
App = CreateObject("AddIn.CashaLotApi")
Ret = App.App.FiscalizeCheck("4000007621", JSONGoodsLstData, JSONPayData)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret": true, "ErrorString": "", "Values": {"ShiftID": "f7d3ba88-5379-4bba-a2ff-9ce3aec66
c78", "ReceiptFiscalNumber": "474775", "ReceiptLocalNumber": "1162", "OfflineMode": "Fals
e", "Type": "10000000", "Base64Str1251ReceiptXML": "PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGl
uZz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1
hNFTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWN1U2NoZW1hTG9jYXRpb249ImNoZWNRmDEueHNkI
j4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVFQ1RTEwPC9ET0NUVWFVBFpG0KICAgIDxET0NTVUJUVWFVBFpA8
L0RPQ1NVQ1RZUEU+DQogICAgPFVJRD43Q0FERkNBMS0wMDNDELQ1MjMtOUNGMi0xMUUzRjk0RjNFMEM8L1V
JRD4NCiAgICA8VE1OPjI0MjM0MjI3PC9USU4+DQogICAgPE9SR05NPTLOWiAi0uXx8i3js+Tw7uzl8vb17f
LwIjwvT1JHTk0+DQogICAgPFBPUSU5UTk0+0uXx8i3js+Tw7uzl8jwvUE9JTlR0TT4NCiAgICA8UE9JTlRBR
ERSPTPK0MCvzCAsIOwUIMro5eIsIMdk8OXx4DwvUE9JTlRBRERSPPg0KICAgIDxPukRFUkRBVEU+MTkwODIw
MjE8L09SREVSREFURT4NCiAgICA8T1JERVJUSU1FPjEwNDA0MTwvT1JERVJUSU1FPg0KICAgIDxPukRFUk5
VTT4xMTYyPC9PukRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzwvQ0FTSERFU0tOVU0+DQogICAgPENBU0
hSRUdJU1RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNVURVJOVU0+DQogICAgPENBU0hJRVi+y+Dh7vDg8
u7w7ejpIM8uIM8uPC9DQVNIUSVSPg0KICAgIDxDt01NRU5UPiMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMj
IyMjIyMjIw0gIyDS5fHy7uLo6SDq7uzl7fLg8CAKIYmJiYmJiYmJiYmJiYmJiYmJiYmJiYmJiYmJiYmJ
jPC9DT01NRU5UPg0KICAgIDxWRVI+MTwvVkVSPg0KICAgIDxPukRFU1RBWE5VTT40NzQ3NzU8L09SREVSVE
FYT1VNPg0KICAgIDwhLS08Q0FTSEFMT1RDT01NRU5UPg0KICAgUEFZTUVOVFNuQVRVUz4zPC9QV1NRU5UU
1RBVFVTPg0KPC9DQVNIQUxPVENPTU1FTlQ+LS0+DQogIDwvQ0hFQ0tIRUFEPg0KICAgQ0hFQ0tUT1RBTD4N
CiAgICA8U1VNPjM2Mi4zNDwvU1VNPg0KICAgIDxESVNDT1VOVFNVTt4ZLjY2PC9ESVNDT1VOVFNVTt4NCiA
gPC9DSEVDS1RPVEFMPg0KICAgQ0hFQ0tQQVk+DQogICAgPFJPVYBST1dOVU09IjEiPg0KICAgICAgPFBBWU
```

ZPUk1DRD4wPC9QQVlGT1JNQ0Q+DQogICAgICA8UEFZRk9STU5NPsP00rLCysA8L1BBWUZPUk1OTT4NCiAgICAgIDxTVU0+NTIuMDA8L1NVTT4NCiAgICA8L1JPVz4NCiAgICA8Uk9XIFJVPV05VTT0iMiI+DQogICAgICA8UEFZRk9STUNEPjE8L1BBWUZPUk1DRD4NCiAgICAgIDxQQVlGT1JNTk0+ysDQ0srAPC9QQVlGT1JNTk0+DQogICAgICA8U1VNPjMxMC4zNDwvU1VNPg0KICAgICAgPFBBWVNzUz4NCiAgICAgICAgPFJVPyBST1dOVU09IjEiPg0KICAgICAgICAgIDxOQU1FPk1BU1RFUkNBukQ8L05BTUU+DQogICAgICAgICAgPEFDUVVJUkVOT7P8Oji4PIgweDt6jwvQUNRVU1SRU5NPg0KICAgICAgICAgIDxBQ1FVSVJFVfJBT1NJRD4wNDg4NjcxNzM2MjA8L0FDUVVJUkVUukFOU01EPg0KICAgICAgICAgIDxQT1NUUkFOU0RBVEU+MTkwODIwMjExMDQwNDA8L1BPU1RSQU5TREFURT4NCiAgICAgICAgICA8UE9TVfJBT1NOVU0+MjI0PC9QT1NUUkFOU05VTT4NCiAgICAgICAgICA8RVBaREVUQUlMUz5YWfhYWfhYWfhYWfhYWFg2ODgzPC9FUFpERVRBSUxTPg0KICAgICAgICAgIDxBVVRIQ0Q+NjgwMTRCPC9BVVRIQ0Q+DQogICAgICAgICAgPFNVTT4zMTAuMzQ8L1NVTT4NCiAgICAgICAgPC9ST1c+DQogICAgICA8L1BBWVNzUz4NCiAgICA8L1JPVz4NCiAgPC9DSEVDS1BBWT4NCiAgPENIRUNLVEFYpG0KICAgIDxST1cgUk9XTlVNPStXj4NCiAgICAgIDxUWVBFpJA8L1RZUEU+DQogICAgICA8TkFNRT7PxmI8L05BTUU+DQogICAgICA8TEVUVEVSPsA8L0xVfVFRUj4NCiAgICAgIDxQUkM+MjAuMDA8L1BSQz4NCiAgICAgIDxUVVJOT1ZFUj4zNjYuMDA8L1RVUk5PVkVSPg0KICAgICAgPFNPVJDRVNVTT4zNjIuMzQ8L1NPVJDRVNVTT4NCiAgICAgIDxTVU0+NjAuMzk8L1NVTT4NCiAgICA8L1JPVz4NCiAgPC9DSEVDS1RBWD4NCiAgPENIRUNLQk9EWT4NCiAgICA8Uk9XIFJVPV05VTT0iMSI+DQogICAgICA8Q09ERT7A8PLo6vPr0u7i4PDzMTwvQ09ERT4NCiAgICAgIDxCQVJDT0RFPjEyMzQ1Njc4OTwvQkFSQ09ERT4NCiAgICAgIDxOQU1FPs3g5+Lg0u7i4PDzMTwvTkFNRT4NCiAgICAgIDxVTklUQ0Q+MjAwOTwvVU5JVENEPg0KICAgICAgPFVOSVROTT748jwvVU5JVE5NPg0KICAgICAgPEFNT1VOVD4xLjAwMDwvQU1PVU5UPg0KICAgICAgPFBSUNFPjE4My4wMDwvUFJJQ0U+DQogICAgICA8TEVUVEVSUz7APC9MRVRURVJTPg0KICAgICAgPENPU1Q+MTgzLjAwPC9DT1NUPg0KICAgICAgPERJU0NPVU5UVf1QRT4wPC9ESVNDT1VOVFRZUEU+DQogICAgICA8REL1TQ09VT1RTVU0+MS44MzwvREL1TQ09VT1RTVU0+DQogICAgPC9ST1c+DQogICAgPFJVPyBST1dOVU09IjIiPg0KICAgICAgPENPREU+wPDy6Orz69Lu4uDw8zI8L0NPREU+DQogICAgICA8QkFSQ09ERT45OTk5OTk5OTk5PC9CQVJDT0RFPg0KICAgICAgPE5BTUU+zeDn4uD57uLg8PMYPC9OQU1FPg0KICAgICAgPFVOSVDRD4yMDA5PC9VTklUQ0Q+DQogICAgICA8VU5JVE5NPvjyPC9VTklUTk0+DQogICAgICA8QU1PVU5UPjEuMDAwPC9BTU9VT1Q+DQogICAgICA8UFJJQ0U+MTgzLjAwPC9QUklDRT4NCiAgICAgIDxMRVRURVJTPsA8L0xVfVFRU1M+DQogICAgICA8Q09TVD4xODMuMDA8L0NPU1Q+DQogICAgICA8REL1TQ09VT1RUWVBFpJA8L0RJU0NPVU5UVf1QRT4NCiAgICAgIDxESVNDT1VOVFNVT4xLjgzPC9ESVNDT1VOVFNVT4NCiAgICA8L1JPVz4NCiAgPC9DSEVDS0JPRfk+DQo8L0NIRUNLPg==" } }

Ret.ReceiptFiscalNum = 474775

Ret.ReceiptLocalNum = 1162

Ret.Return = true

Ret.Type = 10000000

Ret.XML = " <?xml version="1.0" encoding="windows-1251" ?>

<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="check01.xsd">

<CHECKHEAD>

<DOCTYPE>0</DOCTYPE>

<DOCSUBTYPE>0</DOCSUBTYPE>

<UID>8C446439-6B01-4833-A326-AF8FA14CDCDF</UID>

<TIN>24234227</TIN>

<ORGNM>TOB "Тест-гідрометцентр"</ORGNM>

<POINTNM>Тест-гідромет</POINTNM>

<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>

<ORDERDATE>06092021</ORDERDATE>

<ORDERTIME>180649</ORDERTIME>

<ORDERNUM>1189</ORDERNUM>

<CASHDESKNUM>3</CASHDESKNUM>

<CASHREGISTERNUM>400007621</CASHREGISTERNUM>

<CASHIER>Лабораторний П. П.</CASHIER>

<COMMENT>#####

Тестовий коментар

#####</COMMENT>

<VER>1</VER>

```

<ORDERTAXNUM>494296</ORDERTAXNUM>
<!--<CASHALOTCOMMENT>
<PAYMENTSTATUS>3</PAYMENTSTATUS>
</CASHALOTCOMMENT>-->
</CHECKHEAD>
<CHECKTOTAL>
  <SUM>362.34</SUM>
  <DISCOUNTSUM>3.66</DISCOUNTSUM>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТИБКА</PAYFORMNM>
    <SUM>52.00</SUM>
  </ROW>
  <ROW ROWNUM="2">
    <PAYFORMCD>1</PAYFORMCD>
    <PAYFORMNM>КАРТКА</PAYFORMNM>
    <SUM>310.34</SUM>
    <PAYSYS>
      <ROW ROWNUM="1">
        <NAME>MASTERCARD</NAME>
        <ACQUIRENM>Приват Банк</ACQUIRENM>
        <ACQUIRETRANSID>048867173620</ACQUIRETRANSID>
        <POSTRANSDATE>06092021180646</POSTRANSDATE>
        <POSTRANSNUM>224</POSTRANSNUM>
        <EPZDETAILS>XXXXXXXXXXXX6883</EPZDETAILS>
        <AUTHCD>68014B</AUTHCD>
        <SUM>310.34</SUM>
      </ROW>
    </PAYSYS>
  </ROW>
</CHECKPAY>
<CHECKTAX>
  <ROW ROWNUM="1">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>O</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>366.00</TURNOVER>
    <SOURCESUM>353.71</SOURCESUM>
    <SUM>58.95</SUM>
  </ROW>
  <ROW ROWNUM="2">
    <TYPE>1</TYPE>
    <NAME>Акциз</NAME>
    <LETTER>Д</LETTER>
    <PRC>5.00</PRC>
    <TURNOVER>183.00</TURNOVER>
    <SOURCESUM>181.17</SOURCESUM>
    <SUM>8.63</SUM>
  </ROW>
</CHECKTAX>
<CHECKBODY>
  <ROW ROWNUM="1">

```

```

<CODE>АртикулТовару1</CODE>
<BARCODE>123456789</BARCODE>
<NAME>НазваТовару1</NAME>
<UNITCD>2009</UNITCD>
<UNITNM>шт</UNITNM>
<AMOUNT>1.000</AMOUNT>
<PRICE>183.00</PRICE>
<LETTERS>ОД</LETTERS>
<COST>183.00</COST>
<DISCOUNTTYPE>0</DISCOUNTTYPE>
<DISCOUNTSUM>1.83</DISCOUNTSUM>
<EXCISELABELS>
  <ROW ROWNUM="1">
    <EXCISELABEL>1111</EXCISELABEL>
  </ROW>
</EXCISELABELS>
</ROW>
<ROW ROWNUM="2">
  <CODE>АртикулТовару2</CODE>
  <BARCODE>999999999</BARCODE>
  <NAME>НазваТовару2</NAME>
  <UNITCD>2009</UNITCD>
  <UNITNM>шт</UNITNM>
  <AMOUNT>1.000</AMOUNT>
  <PRICE>183.00</PRICE>
  <LETTERS>О</LETTERS>
  <COST>183.00</COST>
  <DISCOUNTTYPE>0</DISCOUNTTYPE>
  <DISCOUNTSUM>1.83</DISCOUNTSUM>
</ROW>
</CHECKBODY>
</CHECK>"

```

7. FiscalizeReturnCheck – Фіскалізація чека повернення товарів, послуг

Для створення та фіскалізації чеку повернення товарів та/або послуг необхідно сформувати JSON структуру JSONGoodsLstData, що міститиме дані номенклатури, що повертається, структуру JSON структуру JSONPayData з формами повернення грошей та номер чеку, за яким здійснюється повернення.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою *
strJSONData	String	JSON структура з описом проданих товарів або послуг
strJSONPayData	String	JSON структура з підсумками оплати по чеку
strFiscalNumberReturnCheck	String	Фіскальний номер чеку, за яким здійснюється повернення

Структури **JSONGoodsLstData** та **JSONPayData** визначені так само, як і в специфікації функції формування чеку продажу [FiscalizeCheck](#).

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документу (див. нижче)
XML	String	XML-структура фіскалізованого чеку

Тип документу *"Type"* визначається числом, опис якого наведено в специфікації **FiscalizeCheck**.

Приклад:

```
JSONGoodsLstData =
  {"ReceiptLst": [{"VendorCode": "АртикулТовару1", "Name":
  "НазваТовару1", "CdUKDZED": "", "CdDKPP": "",
  "GoodsType": "", "Barcode": "123456789", "UnitType": "шт",
  "Quantity": "1", "Price": "183", "Amount": "181,17",
  "DiscountPrc": "", "DiscountSum": "1.83",
  "VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "30.2",
  "IsExcise": false, "OtherParams": null}, {"VendorCode":
  "АртикулТовару2", "Name": "НазваТовару2", "CdUKDZED": "",
  "CdDKPP": "", "GoodsType": "", "Barcode": "999999999",
  "UnitType": "шт", "Quantity": "1", "Price": "183",
  "Amount": "181,17", "DiscountPrc": "",
  "DiscountSum": "1.83", "VATRate": "20",
  "IsPriceIncludeVAT": true, "SumVAT": "30.2", "IsExcise": false,
  "OtherParams": null}], "Comment": "Тестовий коментар"}
JSONPayData =
  {"SumCash": "366", "SumPayByCard": null, "SumPayByCredit": null,
  "SumPayByCertificate": null, "SumPayCheck": "362,34"}
ReceiptFiscalNum = 391929
App = CreateObject("AddIn.CashaLotApi")
Ret = App.FiscalizeReturnCheck(ReceiptFiscalNum, JSONGoodsLstData,
JSONPayData, ReceiptFiscalNum)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret": true, "ErrorString": "", "Values": {"ShiftID": "e204a4d9-3d3a-45f3-850d-732fe3d1a
317", "ReceiptFiscalNumber": "444799", "ReceiptLocalNumber": "860", "OfflineMode": "False
", "Type": "10000001", "Base64Str1251ReceiptXML": "PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGlu
Zz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1h
NTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZWlhTG9jYXRpb249ImNoZWNrMDEueHNkIj
4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVF1QRTR4wPC9ET0NUWVBFPg0KICAgIDxETONTVUJUWVBFPjE8L
```



```

<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
<ORDERRETNUM>444798</ORDERRETNUM>
<CASHIER>Лабораторний П. П.</CASHIER>
<COMMENT>#####
# Тестовий коментар
#####</COMMENT>
<VER>1</VER>
<ORDERTAXNUM>444799</ORDERTAXNUM>
</CHECKHEAD>
<CHECKTOTAL>
<SUM>362.34</SUM>
<DISCOUNTSUM>3.66</DISCOUNTSUM>
</CHECKTOTAL>
<CHECKPAY>
<ROW ROWNUM=""1"">
  <PAYFORMCD>0</PAYFORMCD>
  <PAYFORMNM>ГОТІВКА</PAYFORMNM>
  <SUM>362.34</SUM>
</ROW>
</CHECKPAY>
<CHECKTAX>
<ROW ROWNUM=""1"">
  <TYPE>0</TYPE>
  <NAME>ПДВ</NAME>
  <LETTER>A</LETTER>
  <PRC>20.00</PRC>
  <TURNOVER>366.00</TURNOVER>
  <SOURCESUM>362.34</SOURCESUM>
  <SUM>60.39</SUM>
</ROW>
</CHECKTAX>
<CHECKBODY>
<ROW ROWNUM=""1"">
  <CODE>АртикулТовару1</CODE>
  <BARCODE>123456789</BARCODE>
  <NAME>НазваТовару1</NAME>
  <UNITCD>2009</UNITCD>
  <UNITNM>шт</UNITNM>
  <AMOUNT>1.000</AMOUNT>
  <PRICE>183.00</PRICE>
  <LETTERS>A</LETTERS>
  <COST>183.00</COST>
  <DISCOUNTTYPE>0</DISCOUNTTYPE>
  <DISCOUNTSUM>1.83</DISCOUNTSUM>
</ROW>
<ROW ROWNUM=""2"">
  <CODE>АртикулТовару2</CODE>
  <BARCODE>9999999999</BARCODE>
  <NAME>НазваТовару2</NAME>
  <UNITCD>2009</UNITCD>
  <UNITNM>шт</UNITNM>
  <AMOUNT>1.000</AMOUNT>
  <PRICE>183.00</PRICE>
  <LETTERS>A</LETTERS>
  <COST>183.00</COST>

```

```

<DISCOUNTTYPE>0</DISCOUNTTYPE>
<DISCOUNTSUM>1.83</DISCOUNTSUM>
</ROW>
</CHECKBODY>
</CHECK>"

```

8. GetReceiptLocalNumberByFiscalNumber – Отримання локального номеру чеку за його фіскальним номером

Функція дозволяє визначити локальний номеру чеку за відомим фіскальним номером чеку. Пошук чеку відбувається в локальній базі чеків, а якщо його не знайдено, то виконується пошук чеку на сервері ФСКО.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO1	String	Ідентифікатор пристрою
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Відповідь

	Number	Локальний номер чеку
--	--------	----------------------

Якщо номер чеку не знайдено, то повертається пустий рядок.

Якщо задано параметр **NOAUTOOPENSHIFT= True**, виклик функції при закритій зміні переривається і відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```

FiscalNumberRRO1 = "4000014020"
ReceiptFiscalNumber = "391930"
App = CreateObject("AddIn.CashaLotApi")
ret = ComObjApp.FiscalizeCheck(FiscalNumberRRO1 , JSONGoodsLstData,
JSONPayData );
ReceiptLocalNumber = ret.ReceiptLocalNum ;

```

Приклад відповіді:

Ret = 503

9. GetReceiptXML – Отримання XML структури фіскального чеку

Функція дозволяє отримати XML-структуру зареєстрованого чеку за його відомим фіскальним номером.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою *
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Якщо задано параметр **NOAUTOOPENSHIFT= True**, виклик функції при закритій зміні переривається і відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Відповідь

	String	XML-структура чеку
--	--------	--------------------

Приклад:

```
FiscalNumberRRO = "4000014020"
```

```
ReceiptFiscalNumber = "391930"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetReceiptXML(FiscalNumberRRO, ReceiptFiscalNumber)
```

Приклад відповіді:

```
Ret = <?xml version="1.0" encoding="windows-1251"?>
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>1</DOCSUBTYPE>
    <UID>3E7D7D79-A0E4-4EB9-A8CE-1EC78D04B038</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Киев, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>122452</ORDERTIME>
    <ORDERNUM>810</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <ORDERRETNUM>442921</ORDERRETNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442922</ORDERTAXNUM>
  </CHECKHEAD>
  <CHECKTOTAL>
    <SUM>362.34</SUM>
    <DISCOUNTSUM>3.66</DISCOUNTSUM>
  </CHECKTOTAL>
  <CHECKPAY>
    <ROW ROWNUM="1">
      <PAYFORMCD>0</PAYFORMCD>
      <PAYFORMNM>ГОТІВКА</PAYFORMNM>
      <SUM>362.34</SUM>
    </ROW>
  </CHECKPAY>
  <CHECKTAX>
    <ROW ROWNUM="1">
      <TYPE>0</TYPE>
      <NAME>ПДВ</NAME>
      <LETTER>А</LETTER>
      <PRC>20.00</PRC>
      <TURNOVER>366.00</TURNOVER>
      <SOURCESUM>362.34</SOURCESUM>
      <SUM>60.39</SUM>
    </ROW>
  </CHECKTAX>
  <CHECKBODY>
    <ROW ROWNUM="1">
      <CODE>АртикулТовару1</CODE>
      <BARCODE>123456789</BARCODE>
      <NAME>НазваТовару1</NAME>
      <UNITCD>2009</UNITCD>
      <UNITNM>шт</UNITNM>
```

```

<AMOUNT>1.000</AMOUNT>
<PRICE>183.00</PRICE>
<LETTERS>A</LETTERS>
<COST>183.00</COST>
<DISCOUNTTYPE>0</DISCOUNTTYPE>
<DISCOUNTSUM>1.83</DISCOUNTSUM>
</ROW>
<ROW ROWNUM="2">
<CODE>АртикулТовару2</CODE>
<BARCODE>9999999999</BARCODE>
<NAME>НазваТовару2</NAME>
<UNITCD>2009</UNITCD>
<UNITNM>шт</UNITNM>
<AMOUNT>1.000</AMOUNT>
<PRICE>183.00</PRICE>
<LETTERS>A</LETTERS>
<COST>183.00</COST>
<DISCOUNTTYPE>0</DISCOUNTTYPE>
<DISCOUNTSUM>1.83</DISCOUNTSUM>
</ROW>
</CHECKBODY>
</CHECK>

```

10. GetFirstReceiptByPeriod – Отримати перший чек з періоду

Функцію **GetFirstReceiptByPeriod()** рекомендовано використовувати в ланцюжку запитів для отримання чеків за певний період часу. Дана функція викликається першою для підготовки списку чеків, і для отримання першого чеку з вказаного періоду часу.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
dateBeg	String	Початок періоду в форматі ДДММГГГГ
dateEnd	String	Кінець періоду в форматі ДДММГГГГ

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення першого чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип першого чеку, помилки, локальний та фіскальний номери першого чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер першого чеку
ReceiptLocalNum	String	Локальний номер першого чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип першого чеку (див. нижче)
XML	String	XML-структура першого чеку

Тип документу "Type" визначається числом, опис якого наведено в специфікації FiscalizeCheck.

Приклад:

```
FiscalNumberRRO = "4000007621"  
StartDate = "19072021"  
EndDate = "22072021"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.GetFirstReceiptByPeriod(FiscalNumberRRO, StartDate,  
EndDate)
```

Приклад відповіді:

(перший чек – відкриття зміни)

```
Ret.JsonVal =  
{ "Ret":true, "ErrorString":""," "Values":{"ShiftID":"1d936740-7255-444e-a6e0-f717c0c50fae", "ReceiptFiscalNumber":"438767", "ReceiptLocalNumber":"780", "OfflineMode":"False", "Type":"10100000", "Base64Str1251ReceiptXML":"PD94bWwgdmVyc2l1b251IHRlcj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbnN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZW1hTG9jYXRpb249ImNoZWNrMDEueHNkIj4NCiAgPENIRUNLSEVB RD4NCiAgICA8RE9DVF1QRT4xMDA8L0RPQ1RZUEU+DQogICAgPFVJRDR4xRDkzNjc0M C03MjU1LTQ0NEUtQTZFM C1GNzE3QzBDNTBGQUU8L1VJRDR4NCiAgICA8T1JERVJEQVRFpjE5MDcyMDIxPC9P UkRFUkRBVEU+DQogICAgPE9SREVSVE1NRT4wOTU1MzE8L09SREVSVE1NRT4NCiAgICA8T1JERVJOVU0+Nzg wPC9PUkRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzwwQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1 RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNUVJ0VU0+DQogICAgPENBU0hJRVI+Y+Dh7vDg8u7w7ejpI M/w6Orr40Qgz/Do6uvvg507i6Pc8L0NBU0hJRVI+DQogICAgPFZFUj4xPC9WRVI+DQogICAgPE9SREVSVEFY T1VNPjQzODc2NzwwT1JERVJUQVhOVU0+DQogIDwvQ0hFQ0tIRUFEPg0KPC9DSEVDSz4=" } }  
Ret.ReceiptFiscalNum = 438767  
Ret.ReceiptLocalNum = 780  
Ret.Return = true  
Ret.Type = 10100000  
Ret.XML = " <?xml version="1.0" encoding="windows-1251"?>  
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:noNamespaceSchemaLocation="check01.xsd">  
  <CHECKHEAD>  
    <DOCTYPE>100</DOCTYPE>  
    <UID>1D936740-7255-444E-A6E0-F717C0C50FAE</UID>  
    <ORDERDATE>19072021</ORDERDATE>  
    <ORDERTIME>095531</ORDERTIME>  
    <ORDERNUM>780</ORDERNUM>  
    <CASHDESKNUM>3</CASHDESKNUM>  
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>  
    <CASHIER>Лабораторний Приклад Прикладович</CASHIER>  
    <VER>1</VER>  
    <ORDERTAXNUM>438767</ORDERTAXNUM>  
  </CHECKHEAD>  
</CHECK>  
"
```

11. GetFirstReceiptByShift – Отримати перший чек з зміни

Функцію `GetFirstReceiptByShift()` рекомендовано використовувати в ланцюжку запитів для отримання чеків певної касової зміни. Дана функція викликається першою для підготовки списку чеків, і для отримання першого чеку з вказаної касової зміни.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
dateBeg	String	Початок періоду в форматі ДДММГГГГ
dateEnd	String	Кінець періоду в форматі ДДММГГГГ

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення першого чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип першого чеку, помилки, локальний та фіскальний номери першого чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер першого чеку
ReceiptLocalNum	String	Локальний номер першого чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип першого чеку (див. нижче)
XML	String	XML-структура першого чеку

Тип документу *"Type"* визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
FiscalNumberRRO = "4000007621"  
ShiftID = "ef956a75-3760-4880-a5fe-07c126f81930"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.GetFirstReceiptByShift(deviceID, ShiftID)
```

Приклад відповіді:

(перший чек – відкриття зміни)

```
Ret.JsonVal =  
{ "Ret": true, "ErrorString": "", "Values": { "ShiftID": "ef956a75-3760-4880-a5fe-07c126f81930", "ReceiptFiscalNumber": "435498", "ReceiptLocalNumber": "678", "OfflineMode": "False", "Type": "10100000", "Base64Str1251ReceiptXML": "PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8Q0hfQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZW1hTG9jYXRpb249ImNoZWNrMDEueHNkIj4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVF1QRT4xMDA8L0RQPQ1RZUEU+DQogICA8PFVJRDR5FRjk1NkE3NS0zNzYwLTQ0ODAtQTVGRS0wN0MxMjZGODE5MzA8L1VJRDR4NCiAgICA8T1JERVJEQVRFPjE2MDcyMDIxPC9PUkRFUkRBVEU+DQogICA8PE9SREVSVE1NRT4xNTE3MzQ8L09SREVSVE1NRT4NCiAgICA8T1JERVJOUVU0+Njc4PC9PUkRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzZwQ0FTSERFU0tOVU0+DQogICA8PENBU0hSRUdJU1RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNUURVJOVU0+DQogICA8PENBU0hJRVI+y+Dh7vDg8u7w7ejpIM/w6Orr40Qgz/Do6uvvg507i6Pc8L0NBU0hJRVI+DQogICA8PFZFUj4xPC9WRVI+DQogICA8PE9SREVSVEFYT1VNPjQzNTQ5ODdwT1JERVJUQVhOVU0+DQogIDwvQ0hfQ0tIRUFEPg0KPC9DSEVDSz4=" }  
Ret.ReceiptFiscalNum = 435498  
Ret.ReceiptLocalNum = 678
```



```

Ret.Return = true
Ret.Type = 10100000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<CHECK xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='check01.xsd'>
  <CHECKHEAD>
    <DOCTYPE>100</DOCTYPE>
    <UID>EF956A75-3760-4880-A5FE-07C126F81930</UID>
    <ORDERDATE>16072021</ORDERDATE>
    <ORDERTIME>151734</ORDERTIME>
    <ORDERNUM>678</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний Приклад Прикладович</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>435498</ORDERTAXNUM>
  </CHECKHEAD>
</CHECK>"

```

12. GetNextReceipt– Отримати наступний чек з періоду

Функція **GetNextReceipt** використовується в ланцюжку запитів для отримання чеків за певний період часу. Дана функція викликається після виклику функції **GetFirstReceiptByPeriod** або **GetFirstReceiptByShift**, які готують список чеків. Функцію можна викликати декілька разів, поки не буде отримано весь список чеків. Ознакою кінця списку вважається неуспішність чергового виклику функції.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення наступного чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип першого чеку, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції. Коли в списку більше нема чеків повертається false .
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип поточного чеку (див. нижче)
XML	String	XML-структура поточного чеку

Тип документу "**Type**" визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
```

```

FiscalNumberRRO = "4000007621"
StartDate = "19072021"
EndDate = "22072021"
Set Ret = App.GetFirstReceiptByPeriod(FiscalNumberRRO, StartDate,
EndDate)
Do while Ret.Return
'Обробити отримані чеки
Set Ret = App.GetNextReceipt(FiscalNumberRRO)
Loop

```

Приклад відповіді:

(успішна операція – чек повернення)

```

Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"9de7f9d4-046e-4f86-b686-6ad203068
93a","ReceiptFiscalNumber":"442759","ReceiptLocalNumber":"808","OfflineMode":"False
","Type":"10000001","Base64Str1251ReceiptXML":"PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGlu
Zz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1h
NTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZWlhTG9jYXRpb249ImNoZW5rMDEueHNkIj
4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVF1QRT4wPC9ET0NUWVBFPg0KICAgIDxETONTVUJUWVBFPjE8L
0RPQ1NVQ1RZUEU+DQogICAgPFVJRd41REJGNjU5Ni1GRtU5LTQyRDUtQUM1RC00RDNENTE0MjgzMTE8L1VJ
RD4NCiAgICA8T1JERVJEQVRFPjIyMDcyMDIxPC9PUkrFUKRBVEU+DQogICAgPE9SREVSVE1NRT4xMDIwMzM
8L09SREVSVE1NRT4NCiAgICA8T1JERVJOUVU0+ODA4PC9PUkrFUK5VTT4NCiAgICA8Q0FTSERFU0tOVU0+Mz
wvQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNUURVJOV
U0+DQogICAgPENBU0hJRVI+y+Dh7vDg8u7w7eJpIM8uIM8uPC9DQVNIUkVHSVNUURVJOVU0+DQogICAg
Pg0KICAgIDxPUkrFUF1RBWE5VTT40NDI3NTk8L09SREVSVEFYt1VNPiANCIAGPC9DSEVDS0hFQUQ+DQogIDx
DSEVDS1RPVEFMPg0KICAgIDxTVU0+MzYyLjM0PC9TVU0+DQogIDwvQ0hFQ0tUT1RBTd4NCjwvQ0hFQ0s+"}
}

Ret.ReceiptFiscalNum = 442759
Ret.ReceiptLocalNum = 808
Ret.Return = true
Ret.Type = 10100001

Ret.XML = " <?xml version="1.0" encoding="windows-1251"?>
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>1</DOCSUBTYPE>
    <UID>5DBF6596-FE59-42D5-AC5D-4D3D51428311</UID>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>102033</ORDERTIME>
    <ORDERNUM>808</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442759</ORDERTAXNUM>
  </CHECKHEAD>
  <CHECKTOTAL>
    <SUM>362.34</SUM>
  </CHECKTOTAL>
</CHECK>"

```

(неуспішна операція – кінець ланцюжку отриманих чеків)

```
Ret.JsonVal = {"Ret":false,"ErrorString":"","Values":{}}
```

```
Ret.Return = false
```

13. GetCurrentStatus – Отримати інформацію по поточному стану каси

Функція **GetCurrentStatus()** використовується для отримання інформації по поточному стану каси. Перелік параметрів, доступних для перевірки, наведено в таблиці.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура CashalotApiRetVal, що містить значення наступного чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення "Ret" – успішність операції, "ErrorString" – рядок з описом помилки (в разі невдачі), "Values" – масив параметрів каси (див. таблицю нижче).
Return	Bool	Успішність операції.
ShiftID	String	Ідентифікаційний номер зміни. Якщо зміна відкрита, то відповідає ідентифікатору поточної зміни, якщо зміна закрита, то відповідає ідентифікатору останньої зміни.

Параметри та значення, які передаються в масиві Values структури JsonVal

ShiftState	Стан зміни: 1 – закрита, 2 – відкрита
ShiftStateStr	Стан зміни: "Closed" – закрита, "Opened" – відкрита.
ShiftLocalNumber	Локальний номер зміни
ShiftFiscalNumber	Фіскальний номер зміни
ShiftID	Ідентифікатор зміни
LastCheckLocalNumber	Локальний номер останнього чеку
LastCheckFiscalNumber	Фіскальний номер останнього чеку
LastCheckDateTime	Мітка часу останнього чеку
IsOfflineMode	Стан офлайн-режиму каси: 0 – каса в онлайн-режимі; 1 – каса в офлайн-режимі
NextLocalNumber	Наступний локальний номер чеку
CashBalance	Значення залишків готівки в касі
Base64Str1251CurrentStatusXML	XML-рядок вказаних вище параметрів в форматі Base64.

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
FiscalNumberRRO = "4000007621"
Set Ret = App.GetCurrentStatus(FiscalNumberRRO)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftState":"2","ShiftStateStr":"Opened","ShiftLocalNumber":"1306","ShiftFiscalNumber":"509939","ShiftID":"a6d67115-5a3e-4e70-9cbc-de663f8cb256","LastCheckLocalNumber":"1307","LastCheckFiscalNumber":"509940","LastCheckDateTime":"2021-09-23T13:02:32","IsOfflineMode":"0","NextLocalNumber":"1308","CashBalance":"52","Base64Str1251CurrentStatusXML":"PD94bWwgdMVyc2lvb2I9ICl1xLjAiIGVuY29kaW5nPSAiVVRGLTgiID8+PCBEb2N1bWVudE9ldHBldFBhcmFtZXRlcnMgPia8IFBhcmFtZXRlcnMgQ2h1Y2tOdW1iZXIgaPSAiMTMwNyIgaGQ2FzaEJhbGFuY2UgPSAiNTIiIEl1ZT2ZmbGl1ZGUgPSAiMCIgaTmV4dExvY2FsTnVtYmVyID0gIjEzMDgiIEExhc3RDaGVja0Zpc2NhbE51bWJlciA9ICIlMDk5NDAiIFNoaWZ0TnVtYmVyPSAiMTMwNyIgaGU2hpZnRJRCA9ICJhNmQ2NzExNS01YTNlLlTRlNzAtOWNiYy1kZTY2M2Y4Y2IyNlYyIFNoaWZ0RmlzY2FsTnVtYmVyID0gIjUwOTkzOSIgaGU2hpZnRTRdGF0ZT0gIjIiIERhdGVUaW1lPSAiMjA5MS0wOS0yMlQxMzowMj0zMiIgaLz4gPC8gRG9jZW1lbnRpdXRwdXRQYXJhbWV0ZXJzID4="}}
Ret.Return = true
Ret.ShiftID = "a6d67115-5a3e-4e70-9cbc-de663f8cb256"
```

14. GetFirstZReportByPeriod – Отримати перший Z-звіт за період

Функцію **GetFirstZReportByPeriod()** рекомендовано використовувати в ланцюжку запитів для отримання Z-звітів за певний період часу. Дана функція викликається першою для підготовки списку Z-звітів, і для отримання першого Z-звіту з вказаного періоду часу.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
dateBeg	String	Початок періоду в форматі ДДММГГГГ
dateEnd	String	Кінець періоду в форматі ДДММГГГГ

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення першого Z-звіту з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, тип документа Z-звіту (11000000), локальний та фіскальний номери Z-звіту, ID зміни та Base64 рядок XML структури Z-звіту
ReceiptFiscalNum	String	Фіскальний номер першого Z-звіту
ReceiptLocalNum	String	Локальний номер першого Z-звіту
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документа Z-звіту - 11000000
XML	String	XML-структура першого Z-звіту

Функція **GetNextZReport()** використовується в ланцюжку запитів для отримання Z-звітів за певний період часу. Дана функція викликається після виклику функції **GetFirstZReportByPeriod()**, яка готує список Z-звітів. Функцію можна викликати декілька разів, поки не буде отримано весь список Z-звітів. Ознакою кінця списку вважається неуспішність чергового виклику функції.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення наступного чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери Z-звіту, ID зміни та Base64 рядок XML структури Z-звіту
ReceiptFiscalNum	String	Фіскальний номер Z-звіту
ReceiptLocalNum	String	Локальний номер Z-звіту
Return	Bool	Успішність операції. Коли в списку більше нема Z-звіту повертається false .
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип Z-звіту - - 11000000
XML	String	XML-структура Z-звіту

Тип Z-звіту **"Type"** визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
FiscalNumberRRO = "4000007621"
StartDate = "19072021"
EndDate = "22072021"
Set Ret = App.GetFirstZReportByPeriod(FiscalNumberRRO, StartDate,
EndDate)
Do while Ret.Return
'Обробити отримані звіти
Set Ret = App.GetNextZReport(FiscalNumberRRO)
Loop
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"2c93c78e-f8db-4672-8748-0f21e9754
5da","ReceiptFiscalNumber":"442727","ReceiptLocalNumber":"804","OfflineMode":"False
","Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2l1b2J0iMS4wIiBlbmNvZGluZ
z0id2luZG93cy0xMjUxIj8+DQo8W1JFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1M
U2NoZW1hLW1uc3RhbmNlIiB4c2k6bm90YW1lc3BhY2VTY2h1bWFMb2NhdGlvbj0ienJlcDAXLnhzZCI+DQo
```

```
gIDxaUkVQSEVBRD4NCiAgICA8VU1EPjYyRjQwQUYyLUJDREMtNDIzNC05NjQzLUU4MzY1QUYxOTNGRTwvVU
1EPg0KICAgIDxPUkRFUkRBVEU+MjIwNzIwMjE8L09SREVSREFURT4NCiAgICA8T1JERVJUSU1FPjA5MzQwN
TwvT1JERVJUSU1FPg0KICAgIDxPUkRFUk5VTT44MDQ8L09SREVST1VNPg0KICAgIDxDQVNIREVTS05VTT4z
PC9DQVNIREVTS05VTT4NCiAgICA8Q0FTSFJFR01TVEVST1VNPjQwMDAwMDc2MjE8L0NBU0hSRUdJU1RFUk5
VTT4NCiAgICA8Q0FTSElFUj7L4OHu8ODy7vDt6Okgy4gzy48L0NBU0hJRVI+DQogICAgPFZFUj4xPC9WRV
I+DQogICAgPE9SREVSVEFYT1VNPjQ0MjcyNzwwT1JERVJUQVhOVU0+DQogIDwvWlJFUEhFQUUQ+DQogIDxaU
kVQk9EWT4NCiAgICA8U0VSVklDRU1OUFVUPjAuMDA8L1NFU1ZJQ0VJT1BVVD4NCiAgICA8U0VSVklDRU9V
VFBVVD4wLjAwPC9TRVJWSUNFT1VUUFVUPg0KICA8L1pSRVBCT0RZPg0KPC9aUkVQPg==", "ServiceInput
":null, "ServiceOutput":null, "OrderCount": "0", "OrderExpCount": "0", "TotalSUM":null, "T
otalSUMCash":null, "TotalSUMCard":null, "TotalTAXSUM":null, "ReturnSUM":null, "ReturnSU
MCash":null, "ReturnSUMCard":null, "ReturnTAXSUM":null}}
```

```
Ret.ReceiptFiscalNum = 442727
```

```
Ret.ReceiptLocalNum = 804
```

```
Ret.Return = true
```

```
Ret.Type = 11000000
```

```
Ret.XML = " <?xml version="1.0" encoding="windows-1251"?>
```

```
<ZREP xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="zrep01.xsd">
```

```
<ZREPHEAD>
```

```
<UID>62F40AF2-BCDC-4234-9643-E8365AF193FE</UID>
```

```
<ORDERDATE>22072021</ORDERDATE>
```

```
<ORDERTIME>093405</ORDERTIME>
```

```
<ORDERNUM>804</ORDERNUM>
```

```
<CASHDESKNUM>3</CASHDESKNUM>
```

```
<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
```

```
<CASHIER>Лабораторний П. П.</CASHIER>
```

```
<VER>1</VER>
```

```
<ORDERTAXNUM>442727</ORDERTAXNUM>
```

```
</ZREPHEAD>
```

```
<ZREPBODY>
```

```
<SERVICEINPUT>0.00</SERVICEINPUT>
```

```
<SERVICEOUTPUT>0.00</SERVICEOUTPUT>
```

```
</ZREPBODY>
```

```
</ZREP>"
```

16. GetZReportByShift– Отримати Z-звіт касової зміни

Функція **GetZReportByShift()** призначена для отримання Z-звіту вказаної касової зміни.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
ShiftID	String	Ідентифікатор касової зміни

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення Z-звіту вказаної касової зміни:

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, тип документу Z-звіту (11000000), локальний та фіскальний номери Z-звіту, ID зміни та Base64 рядок XML структури Z-звіту
---------	--------	--

ReceiptFiscalNum	String	Фіскальний номер чеку Z-звіту
ReceiptLocalNum	String	Локальний номер чеку Z-звіту
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документа Z-звіту - 11000000
XML	String	XML-структура Z-звіту

Тип документа Z-звіту "*Type*" визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
FiscalNumberRRO = "4000007621"
ShiftID = "2c93c78e-f8db-4672-8748-0f21e97545da"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetZReportByShift(FiscalNumberRRO, ShiftID)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"2c93c78e-f8db-4672-8748-0f21e97545da","ReceiptFiscalNumber":"442727","ReceiptLocalNumber":"804","OfflineMode":False,"Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2l1b3J0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZWlhLWluc3RhbmNlIiB4c2k6bm90YW1lc3BhY2V2YTY2hlbWFMb2NhdGlvbj0ienJlcDAXLnhzZCI+DQogIDxaUkVQSEVBRD4NCiAgICA8VU1EPjYyRjQwQUYyLUJDREMTNDIzNC05NjQzLUU4MzY1QUYxOTNGRTwvVU1EPg0KICAgIDxUSU4+MjQyMzQyMjc8L1RJTj4NCiAgICA8T1JHTk0+0s7CICLS5fHyLeOz5PDu7OXy9uXt8vAiPC9PUkdOTT4NCiAgICA8UE9JTlROTt7S5fHyLeOz5PDu7OXyPC9QT01OVE5NPg0KICAgIDxQT01OVEFERFI+08rQwK/NwCwg7C4gyuj14iwgOTw5fHgPC9QT01OVEFERFI+DQogICAgPE9SREVSREFURT4yMjA3MjA5MTwvT1JERVJEQVRFPg0KICAgIDxPUkRFU1RJTUU+MDkzNDA1PC9PUkRFU1RJTUU+DQogICAgPE9SREVSST1VNPjgwNDwvT1JERVJOVU0+DQogICAgPENBU0hERVNLT1VNPjM8L0NBu0hERVNLT1VNPg0KICAgIDxQVNIUkVHSVNUrVJOVU0+NDAMDAwNzYyMTwvQ0FTSEJFR01TVEVST1VNPg0KICAgIDxQVNIUSUVSPsvg4e7w4PLu8O3o6SDPLiDPLjwvQ0FTSElFUj4NCiAgICA8VkvSPjE8L1ZFUj4NCiAgICA8T1JERVJUQVhOVU0+NDQyNzI3PC9PUkRFU1RBWE5VTT4NCiAgPC9aUkVQSEVBRD4NCiAgPFpSRVBSRUFMSVo+DQogICAgPFNVTT4zNjIuMzQ8L1NVTT4NCiAgICA8T1JERVJTQ05UPjE8L09SREVSU0NOVD4NCiAgICA8UEFZRk9STVM+DQogICAgICA8Uk9XIFJPV05VTT0iMSI+DQogICAgICAgIDxQQVlGT1JNQ0Q+MDwvUEFZRk9STUNEPg0KICAgICAgICA8UEFZRk9STU5NPsPO0rLCysA8L1BBWUZPuk1OTT4NCiAgICAgICAgPFNVTT4yMTIuMzQ8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgICAgPFJpVjBST1dOVU09IjIiPjQ0KICAgICAgICA8UEFZRk9STUNEPjI8L1BBWUZPuk1DRD4NCiAgICAgICAgPFBBWUZPuk1OTT7RxdSDyNSyysDSPC9QVlGT1JNTk0+DQogICAgICAgIDxTVU0+NtAuMDA8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgICAgPFJpVjBST1dOVU09IjMiPjQ0KICAgICAgICA8UEFZRk9STUNEPjM8L1BBWUZPuk1DRD4NCiAgICAgICAgPFBBWUZPuk1OTT7K0MXEYNI8L1BBWUZPuk1OTT4NCiAgICAgICAgPFNVTT4xMDAuMDA8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgIDwvUEFZRk9STVM+DQogICAgPFRBWEVTPg0KICAgICAgPFJpVjBST1dOVU09IjEiPjQ0KICAgICAgICA8VFlQRt4wPC9UWVBFp0KICAgICAgICA8T1kFNRT7PxmI8L05BTUU+DQogICAgICAgIDxMRVRURVI+wDwvTEVUVEVSPg0KICAgICAgICA8UFJDPjIwLjAwPC9QUkM+DQogICAgICAgIDxUVVJOT1ZFUj4zNjYuMDA8L1RVUk5PVkVSPg0KICAgICAgICA8U09VUkNFU1VNPjM2Mi4zNDwvU09VUkNFU1VNPg0KICAgICAgICA8U1VNPjYwLjM5PC9TVU0+DQogICAgICA8L1JPVz4NCiAgICA8L1RBWEVTPg0KICAgL1pSRVBSRUFMSVo+DQogIDxaUkVQZk9EWT4NCiAgICA8U0VSVklDRU1OUFVUPjAuMDA8L1NFULzJQ0VJTBVVD4NCiAgICA8U0VSVklDRU1VVFVVD4wLjAwPC9TRVJWSUNFT1VUUFVUPg0KICAgL1pSRVBT0RZPg0KPC9aUkVQPg==","ServiceInput":"0","ServiceOutput":"0","OrderCount":"1","OrderExpCount":"0","TotalSUM":"362,34","TotalSUMCash":"212,34","TotalSUMCard":null,"TotalTAXSUM":"60,39","ReturnSUM":null,"ReturnSUMCash":null,"ReturnSUMCard":null,"ReturnTAXSUM":null,"TotalTAX_A_20%_SUM":"60,39","TotalTAX_A_20%_TURNOVER":"362,34","TotalTAX_A_20%_DISCOUNTSUM":"3,66","TotalTAX_A_20%_REMOVEDTAXES":null}}
```



```

Ret.ReceiptFiscalNum = 442727
Ret.ReceiptLocalNum = 804
Ret.Return = true
Ret.Type = 11000000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='zrep01.xsd'>
  <ZREPHEAD>
    <UID>62F40AF2-BCDC-4234-9643-E8365AF193FE</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>093405</ORDERTIME>
    <ORDERNUM>804</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442727</ORDERTAXNUM>
  </ZREPHEAD>
  <ZREPREALIZ>
    <SUM>362.34</SUM>
    <ORDERSCNT>1</ORDERSCNT>
    <PAYFORMS>
      <ROW ROWNUM="1">
        <PAYFORMCD>0</PAYFORMCD>
        <PAYFORMNM>ГОТІВКА</PAYFORMNM>
        <SUM>212.34</SUM>
      </ROW>
      <ROW ROWNUM="2">
        <PAYFORMCD>2</PAYFORMCD>
        <PAYFORMNM>СЕРТИФІКАТ</PAYFORMNM>
        <SUM>50.00</SUM>
      </ROW>
      <ROW ROWNUM="3">
        <PAYFORMCD>3</PAYFORMCD>
        <PAYFORMNM>КРЕДИТ</PAYFORMNM>
        <SUM>100.00</SUM>
      </ROW>
    </PAYFORMS>
    <TAXES>
      <ROW ROWNUM="1">
        <TYPE>0</TYPE>
        <NAME>ПДВ</NAME>
        <LETTER>А</LETTER>
        <PRC>20.00</PRC>
        <TURNOVER>366.00</TURNOVER>
        <SOURCESUM>362.34</SOURCESUM>
        <SUM>60.39</SUM>
      </ROW>
    </TAXES>
  </ZREPREALIZ>

```

```

<ZREPBODY>
  <SERVICEINPUT>0.00</SERVICEINPUT>
  <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

17. ShowReceipt – Вивести візуальне відображення чеку на екран

Функція використовується для виклику вікна з візуальним відображенням зареєстрованого чеку. Вікно з відображенням чеку також дозволяє використати додаткові функції друку або відправки на електронну пошту чеку, кнопки яких присутні у вікні чеку.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Відповідь

	Bool	Успішність операції
--	------	---------------------

Приклад:

```

FiscalNumberRRO = "4000007621"
ReceiptFiscalNumber = "391930"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ShowReceipt (FiscalNumberRRO, ReceiptFiscalNumber)

```

Приклад відповіді:

```
Ret = True
```



18. PrintTextDocument – Надрукувати довільний нефіскальний текст

Функція використовується для друку на чекову стрічку довільного тексту. Після виклику відображається вікно з зображенням тексту і можливістю роздрукувати.

У випадку, коли задано автоматичну відправку чеків на друк і принтер для друку за замовчуванням, друк заданого тексту відбувається без попереднього перегляду:

```

SetParameter("AUTOPRINTMODE", "true");
SetParameter("DEFAULTPRINTERNAME", "xprint80-name");

```

Функцію можливо викликати без відкриття касової зміни.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
DocumentPackage	String	Текст для друку

Відповідь

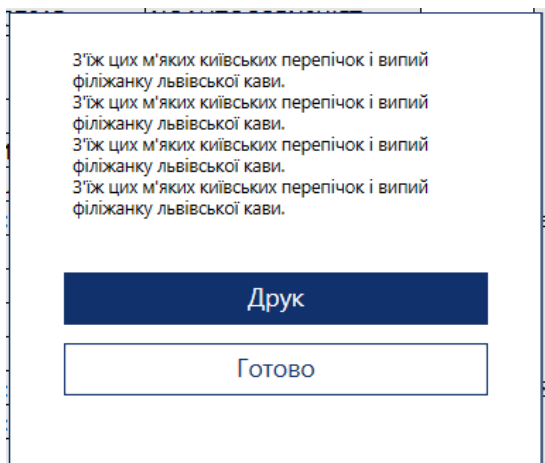
	Bool	Успішність операції
--	------	---------------------

Приклад:

```
FiscalNumberRRO = "4000007621"
DocumentPackage = "З'їж цих м'яких київських перепічок і випий філіжанку львівської кави.
З'їж цих м'яких київських перепічок і випий філіжанку львівської кави.
З'їж цих м'яких київських перепічок і випий філіжанку львівської кави.
З'їж цих м'яких київських перепічок і випий філіжанку львівської кави."
App = CreateObject("AddIn.CashaLotApi")
Ret = App.PrintTextDocument(FiscalNumberRRO, DocumentPackage)
```

Приклад відповіді:

Ret = True



19. ServiceInput – Службове внесення грошей в касу

Для виконання службового внесення в касу використовують функцію **ServiceInput**. Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції [SetParameter](#)), то при цьому також виводиться службове вікно для підтвердження операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою
Amount	Number	Додатне значення суми внесення у форматі <Грн, Коп>

Відповідь

	Bool	Успішність операції
--	------	---------------------

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

і повертається **false**.

Приклад:

```
FiscalNumRRO = "4000007621"  
Amount = "100"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.ServiceInput (FiscalNumRRO, Amount)
```

NEW

20. ServiceInputEx – Службове внесення грошей в касу

Для виконання програмного службового внесення в касу використовують функцію **ServiceInputEx**.

Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції **SetParameter**), то при цьому також виводиться службове вікно для підтвердження операції.

Якщо задано параметр NOINTERFACEMODE = "True", то на екран виводиться вікно з результатом проведеної операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою
Amount	Number	Додатне значення суми внесення у форматі <Грн, Коп>

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, локальний та фіскальний номери документу службового внесення
Return	Bool	Успішність операції
ShiftID	String	Ідентифікатор зміни

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumRRO = "4000007621"  
Amount = "100"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.ServiceInputEx (FiscalNumRRO, Amount)
```

Приклад відповіді:

```
Ret.JsonVal =  
{ "Ret":true, "ErrorString":""," "Values":{"ShiftID":"09992bfb-87cd-4f50-b685-1223a686091b"}}  
Ret.ShiftID = "09992bfb-87cd-4f50-b685-1223a686091b"  
Ret.Return = true
```

21. ServiceOutput – Службова видача грошей з каси

Для виконання службової видачі з каси використовують функцію **ServiceOutput**. Якщо задано параметр **NOINTERFACEMODE = "False"** (за допомогою функції **SetParameter**), то при цьому також виводиться службове вікно для підтвердження операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою
Amount	Number	Додатне значення суми видачі у форматі <Грн, Коп>

Відповідь

	Bool	Успішність операції
--	------	---------------------

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumRRO = "4000007621"  
Amount = "100"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.ServiceOutput(FiscalNumRRO, Amount)
```

Приклад відповіді:

```
Ret = True
```

22. ServiceOutputEx – Службова видача грошей з каси

Для виконання програмного службового внесення в касу використовують функцію

ServiceOutputEx.

Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції SetParameter()), то при цьому також виводиться службове вікно для підтвердження операції.

Якщо задано параметр NOINTERFACEMODE = "True", то на екран виводиться вікно з результатом проведеної операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою*
Amount	Number	Додатне значення суми видачі у форматі <Грн, Коп>

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, локальний та фіскальний номери документу службової видачі
ReceiptFiscalNum	String	Фіскальний номер документу
ReceiptLocalNum	String	Локальний номер документу
Return	Bool	Успішність операції

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumRRO = "4000007621"  
Amount = "100"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.ServiceOutputEx(FiscalNumRRO, Amount)
```

Приклад відповіді:

```
Ret.JsonVal =  
{ "Ret":true, "ErrorString":"","Values":{"ReceiptFiscalNumber":"430098", "ReceiptLocalNumber":"665"} }  
Ret.ReceiptFiscalNum = 430098  
Ret.ReceiptLocalNum = 665  
Ret.Return = true
```

23. GetLastError – Отримання опису останньої помилки

У разі неуспішної операції виклик цієї функції дозволяє отримати опис помилки.

Параметри

Функція не має вхідних параметрів.

Відповідь

	String	Опис помилки
--	--------	--------------

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.GetLastError()
```

Приклади відповіді:

```
Ret = "Помилка закриття зміни"  
Ret = "Помилка відправки чеку до ФСКО"
```

24. PrintXReport - Формування X-звіту

Функція дозволяє викликати на екран X-звіт для перегляду та відправки на друк.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою

Відповідь

	Bool	Успішність операції
--	------	---------------------

Приклад:

```
FiscalNumRRO = "4000007621"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.PrintXReport(FiscalNumRRO)
```

Приклад відповіді:

```
Ret = True
```

25. GetXReport - Формування X-звіту

Виклик функції **GetXReport()** дозволяє сформувати і програмно отримати X-звіт у вигляді JSON-структури.

Параметри

Ім'я	Тип	Опис
------	-----	------

FiscalNumberRRO	String	Ідентифікатор пристрою *
ShowPrintReport	Boolean	Ознака необхідності вивести звіт на друк. "true" - на екран виводиться візуальне відображення X-звіту для перегляду та можливість вручну роздрукувати звіт (аналогічно функції PrintXReport()).

Відповідь

В якості відповіді повертається структура CashlotApiRetVal, що містить значення

JsonVal	String	JSON-структура, яка містить інформацію X-звіту
Return	Bool	Успішність операції
ShiftId	String	Ідентифікатор зміни, в якій сформовано X-звіт
Type	Integer	Тип документу чеку - 12000000
XML	String	XML-структура X-звіту

JsonVal містить наступні об'єкти:

"ServiceInput"	Загальна сума службових внесень по зміні
"ServiceOutput"	Загальна сума службових видач по зміні
"OrderCount"	Кількість чеків продажу
"OrderExpCount"	Кількість чеків повернення
"SumCash"	Поточний залишок готівки в касі
"TotalSUM"	Загальна сума продажу
"TotalSUMCash"	Загальна сума продажу "Готівка"
"TotalSUMCard"	Загальна сума продажу "Картка"
"TotalTAXSUM"	Загальна сума ПДВ по продажу
"ReturnSUM"	Загальна сума повернення
"ReturnSUMCash"	Загальна сума продажу "Готівка"
"ReturnSUMCard"	Загальна сума продажу "Картка"
"ReturnTAXSUM"	Загальна сума ПДВ по продажу
"TotalTAX_<1>_<2> %_<3>"	Деталізація податків по фіскальним чекам
"ReturnTAX_<1>_<2> >%_<3>"	Деталізація податків по чекам повернення
"Base64Str1251ReportXML"	Дані Z-звіту у вигляді XML-структури в кодуванні Windows1251 конвертованої в Base64-рядок


```
1VNPg0KICAgICAgPC9ST1c+DQogICAgPC9UQVhFUz4NCiAgPC9aUkVQUkVUVVVJOPg0KICA8WlJFUEJPRFk+
DQogICAgPFNFULzJQ0VJT1BVVD4wLjAwPC9TRVJWSUNFSU5QVVQ+DQogICAgPFNFULzJQ0VPVVRQVVQ+MC4
wMDwvU0VSVklDRU9VFBVVD4NCiAgPC9aUkVQUkVUVVVJOPg0KICA8WlJFUEJPRFk+=", "ServiceInput":null, "Ser
viceOutput":null, "OrderCount": "2", "OrderExpCount": "2", "TotalSUM": "724,68", "TotalSUM
Cash": "724,68", "TotalSUMCard": null, "TotalTAXSUM": "120,78", "ReturnSUM": "724,68", "Ret
urnSUMCash": "724,68", "ReturnSUMCard": null, "ReturnTAXSUM": "120,78", "TotalTAX_A_20%_S
UM": "120,78", "TotalTAX_A_20%_TURNOVER": "724,68", "TotalTAX_A_20%_DISCOUNTSUM": "7,32"
, "TotalTAX_A_20%_REMOVEDTAXES": null, "ReturnTAX_A_20%_SUM": "120,78", "ReturnTAX_A_20%
_TURNOVER": "724,68", "ReturnTAX_A_20%_DISCOUNTSUM": "7,32", "ReturnTAX_A_20%_REMOVEDTA
XES": null}}
```

```
Ret.Return = true
```

```
Ret.ShiftID = 9de7f9d4-046e-4f86-b686-6ad20306893a
```

```
Ret.Type = 12000000
```

```
Ret.XML = "<?xml version='\"1.0\"' encoding='\"windows-1251\"' ?>
```

```
<ZREP xmlns:xsi='\"http://www.w3.org/2001/XMLSchema-instance\"'
```

```
xsi:noNamespaceSchemaLocation='\"zrep01.xsd\"'
```

```
<ZREPHEAD>
```

```
<UID>2C8C30ED-7221-4F9A-BE54-8B52F3E8E279</UID>
```

```
<TIN>24234227</TIN>
```

```
<ORGNM>ТОБ "Тест-гідрометцентр"</ORGNM>
```

```
<POINTNM>Тест-гідромет</POINTNM>
```

```
<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
```

```
<ORDERDATE>22072021</ORDERDATE>
```

```
<ORDERTIME>122516</ORDERTIME>
```

```
<ORDERNUM>0</ORDERNUM>
```

```
<CASHDESKNUM>3</CASHDESKNUM>
```

```
<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
```

```
<CASHIER>Лабораторний П. П.</CASHIER>
```

```
<VER>1</VER>
```

```
</ZREPHEAD>
```

```
<ZREPREALIZ>
```

```
<SUM>724.68</SUM>
```

```
<ORDERSCNT>2</ORDERSCNT>
```

```
<PAYFORMS>
```

```
<ROW ROWNUM='\"1\"'>
```

```
<PAYFORMCD>0</PAYFORMCD>
```

```
<PAYFORMNM>ГОТІВКА</PAYFORMNM>
```

```
<SUM>724.68</SUM>
```

```
</ROW>
```

```
</PAYFORMS>
```

```
<TAXES>
```

```
<ROW ROWNUM='\"1\"'>
```

```
<TYPE>0</TYPE>
```

```
<NAME>ПДВ</NAME>
```

```
<LETTER>A</LETTER>
```

```
<PRC>20.00</PRC>
```

```
<TURNOVER>732.00</TURNOVER>
```

```
<SOURCESUM>724.68</SOURCESUM>
```

```
<SUM>120.78</SUM>
```

```
</ROW>
```

```
</TAXES>
```

```
</ZREPREALIZ>
```

```
<ZREPRETURN>
```

```
<SUM>724.68</SUM>
```

```

<ORDERSCNT>2</ORDERSCNT>
<PAYFORMS>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТІВКА</PAYFORMNM>
    <SUM>724.68</SUM>
  </ROW>
</PAYFORMS>
<TAXES>
  <ROW ROWNUM="1">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>A</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>732.00</TURNOVER>
    <SOURCESUM>724.68</SOURCESUM>
    <SUM>120.78</SUM>
  </ROW>
</TAXES>
</ZREPRETURN>
<ZREPBODY>
  <SERVICEINPUT>0.00</SERVICEINPUT>
  <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

26. GetLastZReport – Отримати останній Z-звіт

Виклик функції **GetLastZReport ()** дозволяє отримати інформацію по останньому зареєстрованому Z-звіту у вигляді структури `CashalotApiRetVal`.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою *
ShowPrintReport	Boolean	Ознака необхідності вивести звіт на друк. "true" - на екран виводиться візуальне відображення Z-звіту для перегляду та можливість вручну роздрукувати звіт (аналогічно функції <code>PrintXReport()</code>).

Відповідь

В якості відповіді повертається структура `CashalotApiRetVal`, що містить значення

JsonVal	String	JSON-структура, яка містить інформацію останнього зареєстрованого Z-звіту
Return	Bool	Успішність операції
ShiftId	String	Ідентифікатор зміни, в якій сформовано Z-звіт
Type	Integer	Тип документу чеку - 11000000
XML	String	XML-структура Z-звіту
ReceiptFiscalNum	String	Фіскальний номер документу
ReceiptLocalNum	String	Локальний номер документу

JsonVal містить наступні об'єкти:

"ServiceInput"	Загальна сума службових внесень по зміні
"ServiceOutput"	Загальна сума службових видач по зміні
"OrderCount"	Кількість чеків продажу
"OrderExpCount"	Кількість чеків повернення
"TotalSUM"	Загальна сума продажу
"TotalSUMCash"	Загальна сума продажу "Готівка"
"TotalSUMCard"	Загальна сума продажу "Картка"
"TotalTAXSUM"	Загальна сума ПДВ по продажу
"ReturnSUM"	Загальна сума повернення
"ReturnSUMCash"	Загальна сума продажу "Готівка"
"ReturnSUMCard"	Загальна сума продажу "Картка"
"ReturnTAXSUM"	Загальна сума ПДВ по продажу
"TotalTAX_<1>_<2>%_<3>"	Деталізація податків по фіскальним чекам
"ReturnTAX_<1>_<2>%_<3>"	Деталізація податків по чекам повернення
"Base64Str1251ReportXML"	Дані Z-звіту у вигляді XML-структури в кодуванні Windows1251 конвертованої в Base64-рядок

Назви об'єктів податків чеків оплати TotalTAX та повернення ReturnTAX побудовані з використанням складових:

Складова	Зміст складової	Можливі значення	Приклад
<1>	Літера податку зазначена в картках товарів	A, B, B, ...	TotalTAX_A_20%_SUM ReturnTAX_B_5%_TURNOVER TotalTAX_Є_10%_REMOVEDTAXES
<2>	Відсоток податку	0, 5, 20, ...	
<3>	Сума	SUM	
	Обіг	TURNOVER	
	Знижка	DISCOUNTSUM	
	Виключені податки	REMOVEDTAXES	

Приклад:

```
FiscalNumRRO = "4000007621"
ShowPrintReport = false
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetLastZReport(FiscalNumRRO, ShowPrintReport)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"2c93c78e-f8db-4672-8748-0f21e97545da","ReceiptFiscalNumber":"442727","ReceiptLocalNumber":"804","OfflineMode":"False","Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZWlhLWluc3RhbmNlIiB4c2k6bm90YW1lc3BhY2V2YTY2h1bWFMb2NhdGlvbj0ienJlcDAXLnhzZCI+DQogIDxaUkVQSEVBRD4NCiAgICA8VU1EPjYyRjQwQUYyLUJDRERtNDIzNC05NjQzLUU4MzY1QUYxOTNGRTwvVU1EPg0KICAgIDxUSU4+MjQyMzQyMjc8L1RJTj4NCiAgICA8T1JHTk0+0s7CICLS5fHyLeOz5PDU7OXy9uXt8vAiPC9PUkdOTT4NCiAgICA8UE9JTlROTt7S5fHyLeOz5PDU7OXyPC9QT01OVE5NPg0KICAgIDxQT01OVEFERFI+08rQwK/NwCwg7C4gyuj14iwgWOTw5fHgPC9QT01OVEFERFI+DQogICAgPE9SREVSREFURt4yMjA3MjA5MTwvT1JERVJEQVRFPg0KICAgIDxPukRFU1RJTUU+MDkzNDA1PC9PUkRFU1RJTUU+DQogICAgPE9SREVSST1VNPjgwNDwvT1JERVJOVU0+DQogICAgPENBU0hERVNLT1VNPjM8L0NBU0hERVNLT1VNPg0KICAgIDxQVNIUkVHSVNUURVJOVU0+NDAMDAwNzYyMTwvQ0FTSFJFR01TVEVST1VNPg0KICAgIDxQVNIUSUVSPsvg4e7w4PLu8O3o6SDPLiDPLjwvQ0FTSElFUj4NCiAgICA8VkvSPjE8L1ZFUj4NCiAgICA8T1JERVJUQVhOVU0+NDQyNzI3PC9PUkRFU1RBWE5VTT4NCiAgPC9aUkVQSEVBRD4NCiAgPFpSRVBSRUfMSVo+DQogICAgPFNVTT4zNjIuMzQ8L1NVTT4NCiAgICA8T1JERVJTQ05UPjE8L09SREVSU0NOVD4NCiAgICA8UEFZRk9STVM+DQogICAgICA8Uk9XIFJPV05VTT0iMSI+DQogICAgICAgIDxQQVlGT1JNQ0Q+MDwvUEFZRk9STUNEPg0KICAgICAgICA8UEFZRk9STU5NPsPO0rLCysA8L1BBWUZPUk1OTT4NCiAgICAgICAgPFNVTT4yMTIuMzQ8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgICAgPFJPVyBST1dOVU09IjIiPg0KICAgICAgICA8UEFZRk9STUNEPjI8L1BBWUZPUk1DRD4NCiAgICAgICAgPFBBWUZPUk1OTT7RxdDSyNSyysDSPC9QQVlGT1JNTk0+DQogICAgICAgIDxTVU0+NTAuMDA8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgICAgPFJPVyBST1dOVU09IjMiPg0KICAgICAgICA8UEFZRk9STUNEPjM8L1BBWUZPUk1DRD4NCiAgICAgICAgPFBBWUZPUk1OTT7K0MXEYNI8L1BBWUZPUk1OTT4NCiAgICAgICAgPFNVTT4xMDAuMDA8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgIDwvUEFZRk9STVM+DQogICAgPFRBWEVTPg0KICAgICAgPFJPVyBST1dOVU09IjEiPg0KICAgICAgICA8VF1QRT4wPC9UWVBFPg0KICAgICAgICA8TkFNRT7PxmI8L05BTUU+DQogICAgICAgIDxMRVRURVI+WdWvTEVUVEVSPg0KICAgICAgICA8UFJDPjIwLjAwPC9QUkM+DQogICAgICAgIDxUVVJOT1ZFUj4zNjYyMDA8L1RVUk5PVkVSPg0KICAgICAgICA8U09VUkNFU1VNPjM2Mi4zNDwvU09VUkNFU1VNPg0KICAgICAgICA8U1VNPjYwLjM5PC9TVU0+DQogICAgICA8L1JPVz4NCiAgICA8L1RBWEVTPg0KICAgL1pSRVBSRUfMSVo+DQogIDxaUkVQZk9EWT4NCiAgICA8U0VSVk1DRU1OUFVUPjAuMDA8L1NFU1ZJQ0VJTBVVD4NCiAgICA8U0VSVk1DRU1VVFVVD4wLjAwPC9TRVJWSUNFT1VUUFVUPg0KICAgL1pSRVBCT0RZPg0KPC9aUkVQPg==", "ServiceInput": "0", "ServiceOutput": "0", "OrderCount": "1", "OrderExpCount": "0", "TotalSUM": "362,34", "TotalSUMCash": "212,34", "TotalSUMCard": null, "TotalTAXSUM": "60,39", "ReturnSUM": null, "ReturnSUMCash": null, "ReturnSUMCard": null, "ReturnTAXSUM": null, "TotalTAX_A_20%_SUM": "60,39", "TotalTAX_A_20%_TURNOVER": "362,34", "TotalTAX_A_20%_DISCOUNTSUM": "3,66", "TotalTAX_A_20%_REMOVEDTAXES": null}}
Ret.Return = true
Ret.ShiftID = 2c93c78e-f8db-4672-8748-0f21e97545da
Ret.ReceiptFiscalNum = 442727
Ret.ReceiptLocalNum = 804
Ret.XML = "<?xml version='\"1.0\"' encoding='\"windows-1251\"'?>
<ZREP xmlns:xsi='\"http://www.w3.org/2001/XMLSchema-instance\"'
xsi:noNamespaceSchemaLocation='\"zrep01.xsd\"'>
  <ZREPHEAD>
    <UID>62F40AF2-BCDC-4234-9643-E8365AF193FE</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ \"\"Тест-гідрометцентр\"\"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>093405</ORDERTIME>
    <ORDERNUM>804</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
```

```

<ORDERTAXNUM>442727</ORDERTAXNUM>
</ZREPHEAD>
<ZREPREALIZ>
  <SUM>362.34</SUM>
  <ORDERSCNT>1</ORDERSCNT>
  <PAYFORMS>
    <ROW ROWNUM=""1"">
      <PAYFORMCD>0</PAYFORMCD>
      <PAYFORMNM>ГОТІВКА</PAYFORMNM>
      <SUM>212.34</SUM>
    </ROW>
    <ROW ROWNUM=""2"">
      <PAYFORMCD>2</PAYFORMCD>
      <PAYFORMNM>СЕРТИФІКАТ</PAYFORMNM>
      <SUM>50.00</SUM>
    </ROW>
    <ROW ROWNUM=""3"">
      <PAYFORMCD>3</PAYFORMCD>
      <PAYFORMNM>КРЕДИТ</PAYFORMNM>
      <SUM>100.00</SUM>
    </ROW>
  </PAYFORMS>
  <TAXES>
    <ROW ROWNUM=""1"">
      <TYPE>0</TYPE>
      <NAME>ПДВ</NAME>
      <LETTER>A</LETTER>
      <PRC>20.00</PRC>
      <TURNOVER>366.00</TURNOVER>
      <SOURCESUM>362.34</SOURCESUM>
      <SUM>60.39</SUM>
    </ROW>
  </TAXES>
</ZREPREALIZ>
<ZREPBODY>
  <SERVICEINPUT>0.00</SERVICEINPUT>
  <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

27. PrintZReport – Формування Z-звіту

Виклик функції **PrintZReport()** призводить до виконання послідовності операцій аналогічних до виклику функції **CloseShift()**: службова видача з каси наявної готівки, формування Z-звіту та закриття зміни. Проте, виконання даної функції повертає ознаку успішності операції, в той час як виконання **CloseShift()** повертає структуру CashlotApiRetVal.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumRRO</u>	String	Ідентифікатор пристрою

Відповідь

	Bool	Успішність операції
--	------	---------------------

Приклад:

```
FiscalNumRRO = "4000007621"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.PrintZReport(FiscalNumRRO)
```

28. CloseShift - Закриття зміни

Функція використовується для закриття поточної зміни.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура CashalotApiRetVal, що містить значення Z-звіту закритої касової зміни:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери документу Z-звіту, ID зміни та Base64 рядок XML структури звіту
ReceiptFiscalNum	String	Фіскальний номер Z-звіту
ReceiptLocalNum	String	Локальний номер Z-звіту
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер закритої зміни
Type	Integer	Тип документу Z-звіту -
XML	String	XML-структура Z-звіту

JsonVal містить наступні об'єкти:

"ServiceInput"	Загальна сума службових внесень по зміні
"ServiceOutput"	Загальна сума службових видач по зміні
"OrderCount"	Кількість чеків продажу
"OrderExpCount"	Кількість чеків повернення
"TotalSUM"	Загальна сума продажу
"TotalSUMCash"	Загальна сума продажу "Готівка"
"TotalSUMCard"	Загальна сума продажу "Картка"
"TotalTAXSUM"	Загальна сума ПДВ по продажу
"ReturnSUM"	Загальна сума повернення
"ReturnSUMCash"	Загальна сума продажу "Готівка"
"ReturnSUMCard"	Загальна сума продажу "Картка"
"ReturnTAXSUM"	Загальна сума ПДВ по продажу
"TotalTAX_<1>_<2>%_<3>"	Деталізація податків по фіскальним чекам


```
gICAgICA8U1VNPjcyNC420DwvU1VNPg0KICAgICAgPC9St1c+DQogICAgPC9QQVlGT1JNUz4NCiAgICA8VE
FYRVM+DQogICAgICA8Uk9XIFJPV05VTT0iMSI+DQogICAgICAgIDxUWVBFpJA8L1RZUEU+DQogICAgICAgI
DxOQU1FPs/EwjwvTkFNRT4NCiAgICAgICAgPExFVFRFuj7APC9MRVRURVI+DQogICAgICAgIDxQUkM+MjAu
MDA8L1BSQz4NCiAgICAgICAgPFRVUk5PVkVSPjczMi4wMDwvVfVSTk9WRVI+DQogICAgICAgIDxTT1VSQ0V
TVU0+NzI0LjY4PC9TT1VSQ0VTVU0+DQogICAgICAgIDxTVU0+MTIwLjY4PC9TVU0+DQogICAgICA8L1JPVz
4NCiAgICA8L1RBWEVTPg0KICA8L1pSRVBSRVRVUk4+DQogIDxaUkVQQk9EWT4NCiAgICA8U0VSVklDRU1OU
FVUPjUwLjAwPC9TRVJSUNFSU5QVVQ+DQogICAgPFNFU1ZJQ0VPVVRQVVQ+MzAuMDA8L1NFU1ZJQ0VPVVRQ
VVQ+DQogIDwvWlJFUEJPRFk+DQo8L1pSRVA+", "ServiceInput": "50", "ServiceOutput": "30", "Ord
erCount": "2", "OrderExpCount": "2", "TotalSUM": "724,68", "TotalSUMCash": "724,68", "Total
SUMCard": null, "TotalTAXSUM": "120,78", "ReturnSUM": "724,68", "ReturnSUMCash": "724,68",
"ReturnSUMCard": null, "ReturnTAXSUM": "120,78", "TotalTAX_A_20%_SUM": "120,78", "TotalTA
X_A_20%_TURNOVER": "724,68", "TotalTAX_A_20%_DISCOUNTSUM": "7,32", "TotalTAX_A_20%_REMO
VEDTAXES": null, "ReturnTAX_A_20%_SUM": "120,78", "ReturnTAX_A_20%_TURNOVER": "724,68",
"ReturnTAX_A_20%_DISCOUNTSUM": "7,32", "ReturnTAX_A_20%_REMOVEDTAXES": null}}
```

```
Ret.Return = True
```

```
Ret.ShiftID = 9de7f9d4-046e-4f86-b686-6ad20306893a
```

```
Ret.ReceiptFiscalNum = 443043
```

```
Ret.ReceiptLocalNum = 813
```

```
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
```

```
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
```

```
xsi:noNamespaceSchemaLocation='zrep01.xsd'>
```

```
<ZREPHEAD>
```

```
<UID>90FDEE47-F299-4DAD-B482-89C737F5CF3D</UID>
```

```
<TIN>24234227</TIN>
```

```
<ORGNM>TOB ""Тест-гідрометцентр"</ORGNM>
```

```
<POINTNM>Тест-гідромет</POINTNM>
```

```
<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
```

```
<ORDERDATE>22072021</ORDERDATE>
```

```
<ORDERTIME>143143</ORDERTIME>
```

```
<ORDERNUM>813</ORDERNUM>
```

```
<CASHDESKNUM>3</CASHDESKNUM>
```

```
<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
```

```
<CASHIER>Лабораторний П. П.</CASHIER>
```

```
<VER>1</VER>
```

```
<ORDERTAXNUM>443043</ORDERTAXNUM>
```

```
</ZREPHEAD>
```

```
<ZREPREALIZ>
```

```
<SUM>724.68</SUM>
```

```
<ORDERSCNT>2</ORDERSCNT>
```

```
<PAYFORMS>
```

```
<ROW ROWNUM='1'>
```

```
<PAYFORMCD>0</PAYFORMCD>
```

```
<PAYFORMNM>ГОТІВКА</PAYFORMNM>
```

```
<SUM>724.68</SUM>
```

```
</ROW>
```

```
</PAYFORMS>
```

```
<TAXES>
```

```
<ROW ROWNUM='1'>
```

```
<TYPE>0</TYPE>
```

```
<NAME>ПДВ</NAME>
```

```
<LETTER>A</LETTER>
```

```
<PRC>20.00</PRC>
```

```
<TURNOVER>732.00</TURNOVER>
```

```
<SOURCESUM>724.68</SOURCESUM>
```

```
<SUM>120.78</SUM>
```

```

    </ROW>
  </TAXES>
</ZREPREALIZ>
<ZREPRETURN>
  <SUM>724.68</SUM>
  <ORDERSCNT>2</ORDERSCNT>
  <PAYFORMS>
    <ROW ROWNUM=""1"">
      <PAYFORMCD>0</PAYFORMCD>
      <PAYFORMNM>ГОТІВКА</PAYFORMNM>
      <SUM>724.68</SUM>
    </ROW>
  </PAYFORMS>
<TAXES>
  <ROW ROWNUM=""1"">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>A</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>732.00</TURNOVER>
    <SOURCESUM>724.68</SOURCESUM>
    <SUM>120.78</SUM>
  </ROW>
</TAXES>
</ZREPRETURN>
<ZREPBODY>
  <SERVICEINPUT>50.00</SERVICEINPUT>
  <SERVICEOUTPUT>30.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

29. Приклад роботи з кількома ПРРО одночасно в одній обліковій системі використовується новий метод SetPropertyEx

Приклад відкриття зміни:

Процедура ПодключениеСОМОбъект ()

Если ComObjApp = Неопределено Тогда

```
ИмяПлагина = "C:\Users\User\AppData\Roaming\Cashalot\CashaLotApi32.dll";
```

```
ПодключитьВнешнююКомпоненту (ИмяПлагина, "CashaLotApi");
```

Попытка

```
Компонент = Новый ("AddIn.CashaLotApi.CashaLotApi");
```

Исключение

```
ПодключитьВнешнююКомпоненту ("AddIn.CashaLotApi");
```

```
Компонент = Новый ("AddIn.CashaLotApi");
```

КонецПопытки;

```
ComObjApp = Новый СОМОбъект ("AddIn.CashaLotApi");
```

```
FiscalNumberRRO1 = "XXXXXXXXXX";
```

```

ComObjApp.SetParameterEx (FiscalNumberRR01, "PathToCashalotDir",
"C:\Users\User\AppData\Roaming\Cashalot\Cashalot1");
ComObjApp.SetParameterEx (FiscalNumberRR01, "DeviceIDFnRR0",
FiscalNumberRR01 );
ComObjApp.SetParameterEx (FiscalNumberRR01, "NOINTERFACEMODE", "False");
ComObjApp.SetParameterEx (FiscalNumberRR01, "PathToCertificate",
"C:\Cashalot\Cashalot1\Ключи\Касир");
ComObjApp.SetParameterEx (FiscalNumberRR01, "PwdToCertificate", "111");
ComObjApp.SetParameterEx (FiscalNumberRR01, "USETOKEN", "False");
ComObjApp.SetParameterEx (FiscalNumberRR01, "NOAUTOOPENSIFT", "False");
ComObjApp.SetParameterEx (FiscalNumberRR01, "DEFAULTPRINTERNAME", "");
ComObjApp.SetParameterEx (FiscalNumberRR01, "AUTOPRINTMODE", "True");

FiscalNumberRR02 = "YYYYYYYYYYY";

ComObjApp.SetParameterEx ( FiscalNumberRR02, "PathToCashalotDir" ),
"C:\Users\User\AppData\Roaming\Cashalot\Cashalot2");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "DeviceIDFnRR0",
FiscalNumberRR02 );
ComObjApp.SetParameterEx ( FiscalNumberRR02, "NOINTERFACEMODE", "False");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "PathToCertificate",
"C:\Cashalot\Cashalot2\Ключи\Старший касир");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "PwdToCertificate", "111");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "USETOKEN", "False");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "NOAUTOOPENSIFT", "False");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "DEFAULTPRINTERNAME", "");
ComObjApp.SetParameterEx ( FiscalNumberRR02, "AUTOPRINTMODE", "True");

```

КонецЕсли;

КонецПроцедуры

```

\\\ Дали виконуємо відкриття змін для кас FiscalNumberRR01 = "XXXXXXXXXX" та
FiscalNumberRR02 = "YYYYYYYYYYY"

```

Процедура OpenShift (Команда)

ПодключениеСОМОбъект ();

```

gh = ComObjApp.OpenShift (FiscalNumberRR01);
Сообщить ("Зміна відкрита ");

```

КонецПроцедуры

&НаКлиенте

Процедура OpenShift_2 (Команда)

ПодключениеСОМОбъект ();

```

gh = ComObjApp.OpenShift (FiscalNumberRR02);

```

```
Сообщить("Зміна відкрита ");
```

```
КонецПроцедуры
```

NEW

30. Можна задавати параметри, які ПРРО використовувати для кожного методу окремо.

Приклад для відкриття зміни

```
Процедура OpenShift (Команда)
```

```
ИмяПлагина = "C:\ProgramData\Cashalot\Cashalot2\CashaLotApi64.dll";
```

```
ПодключитьВнешнююКомпоненту (ИмяПлагина, "CashaLotApi");
```

```
FiscalNumberRRO = "XXXXXXXXXX";
```

```
Попытка
```

```
Компонент = Новый("AddIn.CashaLotApi.CashaLotApi");
```

```
Исключение
```

```
ПодключитьВнешнююКомпоненту("AddIn.CashaLotApi");
```

```
Компонент = Новый("AddIn.CashaLotApi");
```

```
КонецПопытки;
```

```
ComObjApp = Новый СОМОбъект("AddIn.CashaLotApi");
```

```
ComObjApp.SetParameterEx (FiscalNumberRRO, "PathToCashalotDir",  
"C:\ProgramData\Cashalot\Cashalot_2\");
```

```
ComObjApp.SetParameterEx FiscalNumberRRO, ("DeviceIDFnRRO",  
FiscalNumberRRO);
```

```
ComObjApp.SetParameterEx (FiscalNumberRRO, "AUTOPRINTMODE", "True");
```

```
gh = ComObjApp.OpenShift(FiscalNumberRRO);
```

```
Сообщить("Зміна відкрита ");
```

```
КонецПроцедуры
```

- Реєстрація СОМ об'єкта

- Завдання налаштувань для кількох ПРРО за допомогою нового методу SetParameterEx

- робота одночасно з двома примірниками програм SetParameterEx:

OpenShift	(Открытие смены)
FiscalizeReceipt	(Регистрация чека)
FiscalizeReturnCheck	(Возврат чека)
ServiceInput	(Служебное внесение)
ServiceOutput	(Служебная выдача)
PrintXReport	(X отчет)
PrintZReport	(Z отчет)
CloseShift	(Закрытие смены)
GetReceiptLocalNumber	(Получение локального номера чека)
GetReceiptFiscalNumber	(Получение фискального номера чека)

GetReceiptXML	(Получение XML чека)
GetCurrentStatus	(Состояние кассы)

31. Ломбардні операції

Для фіскалізації ломбардних послуг введено системні номенклатури, які необхідно використовувати при виклику методу фіскалізації чеку `Fiskalizecheck()`. Повернення товару при ломбардних операціях не підтримується.

Також необхідно вказати тип документу `DocType = 4` в JSON-структурі параметрів чеку.

За необхідності можна вказати номер договору `DocNumber`, за яким надається ломбардна послуга.

Системні номенклатури для ломбардних операцій.

Артикул	Назва	Тип послуги	Напрямок руху*
"ЛО-00001"	"Повернення страхування"	"3"	-
"ЛО-00002"	"Надання кредиту"	"3"	-
"ЛО-00003"	"Добір кредиту"	"3"	-
"ЛО-00004"	"Надання додаткового кредиту"	"3"	-
"ЛО-00005"	"Зберігання"	"2"	+
"ЛО-00006"	"Страховий платіж"	"2"	+
"ЛО-00007"	"Часткове повернення кредиту (Частковий викуп)"	"2"	+
"ЛО-00008"	"Повне повернення кредиту (Повний викуп)"	"2"	+
"ЛО-00009"	"Повернення кредиту"	"2"	+
"ЛО-00010"	"Користування кредитом"	"2"	+
"ЛО-00011"	"Сплата відсотків по кредиту"	"2"	+
"ЛО-00012"	"Сплата пені"	"2"	+

* "-" видача грошей з каси; "+" повернення грошей в касу.

Напрямок руху необхідно враховувати при обчисленні кінцевого підсумку в чеку: від позицій типу 2 віднімаються позиції типу 3.

Приклад

```
Set App = CreateObject("AddIn.CashaLotApi")
DeviceID = "4000007621"
Itmdata =
{"ReceiptList": [{"VendorCode": "ЛО-00009", "Name": "Повернення кредиту", "DocNumber": "01/a12 від 15.08.21", "GoodsType": "2", "CdUKDZED": null, "CdDKPP": null, "Barcode": null, "UnitType": "шт", "Quantity": "1", "Price": "100", "Amount": "100", "DiscountPrc": null, "DiscountSum": null, "IsPriceIncludeVAT": false, "VATRate": null, "SumVAT": null, "IsExcise": false, "OtherParams": null}, {"VendorCode": "ЛО-00002", "Name": "Надання кредиту", "DocNumber": "31/612 від
```

```
15.08.21", "GoodsType": "3", "CdUKDZED": null, "CdDKPP": null, "Barcode": null, "UnitType": "шт", "Quantity": "1", "Price": "150", "Amount": "150", "DiscountPrc": null, "DiscountSum": null, "IsPriceIncludeVAT": false, "VATRate": null, "SumVAT": null, "IsExcise": false, "OtherParameters": null}, {"Comment": "Тестова ломбардна операція", "DocType": "4"}]

```

```
Paydata =
{"SumCash": "-50", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "-50"}
Itmdata = [{"ReceiptLst": [{"VendorCode": "ЛО-00009", "Name": "Повернення кредиту", "DocNumber": "01/a12 від 15.08.21", "GoodsType": "2", "CdUKDZED": null, "CdDKPP": null, "Barcode": null, "UnitType": "шт", "Quantity": "1", "Price": "100", "Amount": "100", "DiscountPrc": null, "DiscountSum": null, "IsPriceIncludeVAT": false, "VATRate": null, "SumVAT": null, "IsExcise": false, "OtherParameters": null}, {"VendorCode": "ЛО-00002", "Name": "Надання кредиту", "DocNumber": "31/612 від 15.08.21", "GoodsType": "3", "CdUKDZED": null, "CdDKPP": null, "Barcode": null, "UnitType": "шт", "Quantity": "1", "Price": "50", "Amount": "50", "DiscountPrc": null, "DiscountSum": null, "IsPriceIncludeVAT": false, "VATRate": null, "SumVAT": null, "IsExcise": false, "OtherParameters": null}], "Comment": "Тестова ломбардна операція", "DocType": "4"}]
Paydata =
{"SumCash": "50", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "50"}

```

```
Set ret = App.FiscalizeCheck(DeviceID, Itmdata, Paydata)
```

```
Wscript.Echo ret.XM
```

Приклад відповіді

<p>ТОВ "Тест-гідрометцентр" Тест-гідромет УКРАЇНА, м. Київ, Адреса ІД 24234227 ФН ПРРО 4000007621 КАСИР Лабораторний П. П.</p> <p>Чек №527988</p>	
<p>ОДЕРЖАНО: Повернення кредиту 100.00 Договір № 01/a12 від 15.08.21 СУМА 100.00 ГРН</p>	
<p>ВИДАНО: Надання кредиту 150.00 Договір № 31/612 від 15.08.21 СУМА 150.00 ГРН</p>	
<p>Видано клієнту 50.00 грн</p>	
<p>ГОТІВКА 50.00 ГРН Тестова ломбардна операція</p>	
527988	10/7/2021 3:23:40 PM
	
ФІСКАЛЬНИЙ ЧЕК ВИДАТКОВИЙ ЧЕК	ОНЛАЙН Cashalot
<p>Друк</p> <p>Надіслати на Email</p> <p>Готово</p>	

<p>ТОВ "Тест-гідрометцентр" Тест-гідромет УКРАЇНА, м. Київ, Адреса ІД 24234227 ФН ПРРО 4000007621 КАСИР Лабораторний П. П.</p> <p>Чек №528004</p>	
<p>ОДЕРЖАНО: Повернення кредиту 100.00 Договір № 01/a12 від 15.08.21 СУМА 100.00 ГРН</p>	
<p>ВИДАНО: Надання кредиту 50.00 Договір № 31/612 від 15.08.21 СУМА 50.00 ГРН</p>	
<p>Одержано від клієнта 50.00 грн</p>	
<p>ГОТІВКА 50.00 ГРН повернення більше видачі</p>	
528004	10/7/2021 3:28:55 PM
	
ФІСКАЛЬНИЙ ЧЕК ВИДАТКОВИЙ ЧЕК	ОНЛАЙН Cashalot
<p>Друк</p> <p>Надіслати на Email</p> <p>Готово</p>	

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"17b433a0-1021-4dfc-b954-1bc3bfbf2950","ReceiptFiscalNumber":"527988","ReceiptLocalNumber":"1375","OfflineMode":"False","Type":"10004000","Base64Str1251ReceiptXML":"PD94bWwgdMvYc2l1b2J0iMS4wIiBlbmNvZG1uZz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZW1hTG9jYXRpb249ImNoZWNRmDEueHNkIj4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVFlQRT40PC9ET0NUVWBFp0KICAgIDxET0NTVUJUUVWBFpja8L0RPQ1NVQ1RZUEU+DQogICAgPFVJRjRD5CNzFCMjRGQ101OTEzLTQxQTktOENERS04ODYxOTRCNkEzOTU8L1VJRjRD4NCiAgICA8VElOPjI0MjM0MjI3PC9USU4+DQogICAgPE9SR05NPTLOWiAi0uXx8i3js+Tw7uzl8vbl7fLwIjwvT1JHTk0+DQogICAgPFBPSU5UTk0+0uXx8i3js+Tw7uzl8jwvUE9JT1ROTt4NCiAgICA8UE9JT1RBRERSPtPK0McvzCAsIOwIMro5eIsIMdk8OXx4DwvUE9JT1RBRERSp0KICAgIDxPUkRFUkRBVEU+MDcxMDIwMjE8L09SREVSREFURt4NCiAgICA8T1JERVJUSU1FPjE1MjM0MDwvT1JERVJUSU1FPg0KICAgIDxPUkRFUk5VTT4xMzc1PC9PUkRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzwwQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1RFUk5VTT40MDAwMDA3NjIxcPC9DQVNIUkVHSVNURVJOVU0+DQogICAgPENBU0hJRVI+y+Dh7vDg8u7w7ejpIM8uIM8uPC9DQVNIUUVSPg0KICAgIDxDT01NRU5UPtLl8fLu4uAg6+7s4eDw503gIO7v5fDg9rP/PC9DT01NRU5UPg0KICAgIDxWRVI+MTwvVkVSPg0KICAgIDxPUkRFUkRBWE5VTT41Mjc5ODg8L09SREVSVEFYTlVNPg0KICAgIDwhLS08Q0FTSEFMT1RDT01NRU5UPg0KICAg8UEFZTUVOVFNUQVRVUz40PC9QQV1NRU5UU1RBVFTVp0KPC9DQVNIQUxPVENPTU1FTlQ+LS0+DQogIDwvQ0hFQ0tIRUFEPg0KICAg8Q0hFQ0tUT1RBTd4NCiAgICA8U1VNPi01MC4wMDwvU1VNPg0KICAgIDxQV05TVU1JU1NVRUQ+MTUwLjAwPC9QV05TVU1JU1NVRUQ+DQogICAgPFbXT1NVTvJFQ0VJvKVEPjEwMC4wMDwvUfD0U1VNUkVDRU1WRUQ+DQogIDwvQ0hFQ0tUT1RBTd4NCiAgPENIRUNLUEFZPg0KICAgIDxST1cgUk9XTlVNPStIj4NCiAgICA8IDxQQV1GT1JNQ0Q+MDwvUEFZRk9STUNEPg0KICAgICA8PFBBWUZPUk10TT7DztKywSrAPC9QQV1GT1JNTk0+DQogICAgICA8U1VNPi01MC4wMDwvU1VNPg0KICAgIDwvUk9XPg0KICAg8L0NIRUNLUEFZPg0KICAg8Q0hFQ0tCT0RZPg0KICAgIDxST1cgUk9XTlVNPStIj4NCiAgICA8IDxDT0RFPsvOLTAWMDA5PC9DT0RFPg0KICAgICA8PE5BTUU+z+7i5fdt5e3t/yDq8OXk6PlzPC9QU1FPg0KICAgICA8PFVOSVRDRD4yMDA5PC9VTklUQ0Q+DQogICAgICA8VU5JVE5NPvjyPC9VTklUTk0+DQogICAgICA8QU1PVU5UPjEuMDAwPC9BTU9VTlQ+DQogICAgICA8UFJJQU0+MTAwLjAwPC9QUk1DRT4NCiAgICA8IDxDT1NUPjEwMC4wMDwvQ09TVd4NCiAgICA8IDxQQV1FU1BBQ1ROVU0+MDEvYTEyIokz5CAxNS4wOC4yMTwvUEFZRvJQQUNUT1VNPg0KICAgICA8PFbXTkrJUj50cnVlPC9QV05ESVI+DQogICAgPC9ST1c+DQogICAgPFJPVyBST1dOVU09IjIiPj0KICAgICA8PENPREU+y84tMDAwMDI8L0NPREU+DQogICAgICA8TktFNRT7N4OTg7e3/IOrw5eTo8vM8L05BTUU+DQogICAgICA8VU5JVENEPjIwMDk8L1VOSVRDRD4NCiAgICA8IDxVTklUTk0++PI8L1VOSVROTT4NCiAgICA8IDxBTU9VTlQ+MS4wMDA8L0FNT1VOVD4NCiAgICA8IDxQUk1DRT4xNTAuMDA8L1BSSUNFPg0KICAgICA8PENPU1Q+MTUwLjAwPC9DT1NUPg0KICAgICA8PFBBWUVSUEFDVE5VTT4zMS/hMTIj4rPkIDE1LjA4LjIxcPC9QQV1FU1BBQ1ROVU0+DQogICAgICA8UFdORE1SPmZhbHNlPC9QV05ESVI+DQogICAgPC9ST1c+DQogIDwvQ0hFQ0tCT0RZPg0KPC9DSEVDSz4="}}
```

Ret.ReceiptFiscalNum = 527988

Ret.ReceiptLocalNum = 1375

Ret.Return = true

Ret.Type = 10004000

ret.ShiftID = 17b433a0-1021-4dfc-b954-1bc3bfbf2950

Ret.XML = "<?xml version=\"1.0\" encoding=\"windows-1251\"?>

<CHECK xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"

xsi:noNamespaceSchemaLocation=\"check01.xsd\">

<CHECKHEAD>

<DOCTYPE>4</DOCTYPE>

<DOCSUBTYPE>0</DOCSUBTYPE>

<UID>B71B24FB-5913-41A9-8CDE-886194B6A395</UID>

<TIN>24234227</TIN>

<ORGNM>TOB "Тест-гідрометцентр"</ORGNM>

<POINTNM>Тест-гідромет</POINTNM>

<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>

<ORDERDATE>07102021</ORDERDATE>

<ORDERTIME>152340</ORDERTIME>

<ORDERNUM>1375</ORDERNUM>

<CASHDESKNUM>3</CASHDESKNUM>

```

<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
<CASHIER>Лабораторний П. П.</CASHIER>
<COMMENT>Тестова ломбардна операція</COMMENT>
<VER>1</VER>
<ORDERTAXNUM>527988</ORDERTAXNUM>
<!--<CASHALOTCOMMENT>
<PAYMENTSTATUS>4</PAYMENTSTATUS>
</CASHALOTCOMMENT>-->
</CHECKHEAD>
<CHECKTOTAL>
  <SUM>-50.00</SUM>
  <PWNSUMISSUED>150.00</PWNSUMISSUED>
  <PWNSUMRECEIVED>100.00</PWNSUMRECEIVED>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТІВКА</PAYFORMNM>
    <SUM>-50.00</SUM>
  </ROW>
</CHECKPAY>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>ЛО-00009</CODE>
    <NAME>Повернення кредиту</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>100.00</PRICE>
    <COST>100.00</COST>
    <PAYERPACTNUM>01/a12 від 15.08.21</PAYERPACTNUM>
    <PWNDIR>>true</PWNDIR>
  </ROW>
  <ROW ROWNUM="2">
    <CODE>ЛО-00002</CODE>
    <NAME>Надання кредиту</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>150.00</PRICE>
    <COST>150.00</COST>
    <PAYERPACTNUM>31/612 від 15.08.21</PAYERPACTNUM>
    <PWNDIR>>false</PWNDIR>
  </ROW>
</CHECKBODY>
</CHECK>"

```

32. Рекомендований порядок роботи з CashlotApi

Для коректної роботи розрахунково-облікової системи рекомендується описана послідовність виклику функцій при виконанні касових операцій.

1. Встановлення параметрів

Рекомендовано за допомогою виклику функції SetParameter або SetParameterEx (декілька ПРРО) встановити параметри в рекомендованій послідовності:

FiscalNumRRO
PathToCashalotDir
NOINTERFACEMODE
NOAUTOOPENSHIFT
USETOKEN
PathToCertificate
PwdToCertificate
OtherParametrs
перед початком роботи.

2. Відкриття зміни

Для відкриття касової зміни необхідно скористатися функцією

OpenShift

3. Продаж товарів, послуг

Для формування чека та його фіскалізації створюються JSON-структури з масивом товарів та підсумками оплати по чеку, які передаються в якості параметрів в функцію

FiscalizeCheck

У чеку необхідно сформувати щонайменше 1 рядок товарів, **та встановити розширений порядок оплати для поточного чека**

PaymentOrderType

Спосіб оплати (0-звичайна | 1 - передоплата | 2 - післяплата | 3 - інтернет продаж.

4. Повернення товарів, рекомпенсація послуг

Для повернення товару або рекомпенсації послуги необхідно скористатися функцією

FiscalizeReturnCheck

в якій потрібно вказати фіскальний номер чеку, за яким було продано товар або послугу.

5. Пошук фіскального та локального номеру чеку

Для визначення фіскального номеру чеку за його локальним номером використовується функція

GetReceiptFiscaNumberByLocalNumber

Для визначення локального номеру чеку за його фіскальним номером використовується функція

GetReceiptLocalNumberByFiscalNumber

6. Отримання XML-структури чеку

Функція

GetReceiptXML

повертає XML-структуру чеку за його вказаним фіскальним номером.

7. Відображення зареєстрованого чеку

Функція

ShowReceipt

відкриває вікно з візуальним відображенням чеку, яке аналогічне вікно з відображенням чеку після закриття чеку. У вікні є кнопки "Друк" та "Надіслати на email", які дозволяють роздрукувати або відправити на електронну пошту чек.

8. Надрукувати періодичний звіт по датам

Функція

PrintPReportDate

Функція надрукувати періодичний звіт по датам використовується для можливості друку періодичного звіту використовуючи проміжки часу, що фіксується значеннями (ДатаПочаткуПеріоду-у форматі "dd.ММ.уууу" та ДатаЗакінченняПеріоду- у форматі "dd.ММ.уууу").

9. Налаштування електронної пошти

Електронну пошту можна налаштувати після фіскалізації чеку або виклику функції відображення чеку (**ShowReceipt**). Для цього, у вікні візуального відображення зареєстрованого чеку необхідно натиснути кнопку "Надіслати на email", після чого відкривається вікно налаштування електронної пошти ПРРО Кашалот.

Для налаштування необхідно зазначити електронну адресу, з якої будуть відправлятися повідомлення, а також налаштування з'єднання з поштовою скринькою: вказати SMTP-сервер, порт, ім'я для входу в поштову скриню, пароль поштової скрині.

Після успішного налаштування електронної пошти повторне натискання на кнопку "Надіслати на email" викликає діалогове вікно, в якому потрібно вказати електронну адресу клієнта, на яку відправляється чек, за необхідності додати вказану адресу в базу даних постійних клієнтів, та відправити чек.

10. Службове внесення грошей в касу

Для виконання службового внесення в касу грошей використовується функція

ServiceInput

або

ServiceInputEx

11. Службова видача грошей з каси

Для виконання службової видачі з каси використовується функція

ServiceOutput

або

ServiceOutputEx

12. Формування X-звіту

X-звіт формується в результаті виконання функції

PrintXReport

або

GetXReport

13. Отримання чеків

Для отримання чеків з історії операцій реалізовано функції, що передбачають завантаження чеків за певний період часу або за певну касову зміну.

Для формування списку чеків за заданий період часу скористайтеся функцією

GetFirstReceiptByPeriod

Для формування списку чеків за задану касову зміну скористайтеся функцією

GetFirstReceiptByShift

Обидві функції формують список чеків і повертають перший чек з сформованого списку.

Для отримання наступних чеків скористайтеся функцією

GetNextReceipt

необхідну кількість разів, або поки не буде вичерпано список чеків

14. Отримання документів Z-звітів закритих касових змін

Для отримання Z-звітів з історії операцій реалізовано функції, що передбачають завантаження документів за певний період часу або для певної касової зміни.

Для формування списку документів за заданий період часу скористайтеся функцією

GetFirstZReportByPeriod

Функція формує список чеків і повертає перший документ з сформованого списку.

Для отримання наступних документів скористайтеся функцією

GetNextZReport

необхідну кількість разів, або поки не буде вичерпано список документів.

Для отримання Z-звіту заданої касової зміни скористайтеся функцією

GetZReportByShift

Якщо звіт для вказаної касової зміни відсутній (наприклад, поточна касова зміна), функція поверне ознаку неуспішності операції.

15. Отримання останнього Z-звіту

Для отримання і перегляду останнього зареєстрованого Z-звіту використовується функція

GetLastZReport

16. Закриття зміни, Z-звіт

Формування Z-звіту відбувається при закритті зміни, тому, виконання операції закриття зміни та формування Z-звіту рівнозначні за результатом. Для цього виконується одна з функцій

PrintZReport

або

CloseShift

Виклик вказаних функцій призводить до виконання послідовності операцій: службова видача з каси наявної готівки, формування Z-звіту та закриття зміни. Проте, виконання функції **PrintZReport()** повертає ознаку успішності операції, в той час як виконання **CloseShift()** повертає структуру **CashalotApiRetVal**.

Приклад міститься у вкладенні вище.

33. Приклад реалізації підключення COM бібліотеки на C++

1. Створення файлів заголовків *.h

Якщо відсутні файли заголовків, або потрібно їх оновити:

1. генеруємо через імпорт, вказавши шлях до COM бібліотеки, наприклад:
`#import "C:\\Users\\User1\\AppData\\Local\\CashaLot\\CashaLotApi64.dll"`
2. Компілюємо solution з рядком import (після компіляції цей рядок прибираємо).
3. Далі в папці debug\ вашого solution з'являються файли CashalotApi64.tlh та CashalotApi64.tli
4. Переносимо їх в каталог solution, перейменовуємо в CashalotApi.tlh, CashalotApi.tli (за бажанням, можна зібрати два x32 та x64, та підключати директивами)
5. Всередині CashalotApi.tlh прибираємо повний шлях до CashalotApi64.tli і також міняємо ім'я на CashalotApi.tli
6. Заголовочні файли готові до використання, підключаємо їх в solution:

```
#include "CashalotApi.tlh"  
using namespace CashalotApi;
```

2. Зауваження

1. При старті програми додаємо ініціалізацію роботи з COM об'єктами: `CoInitialize(NULL)`;
2. При закритті програми припиняємо роботу з COM об'єктами: `CoUninitialize()`;
3. Перед початком роботи з COM-об'єктом ініціалізуємо його викликом `InitCashalotApiComInterface()`, перед виходом з програми звільняємо ресурси викликом функції `TermCashalotApiComInterface()`;
4. Функції повертають структуру **ICashalotApiRetVal** або **BOOL**.

3. Лістинг

```
#include "stdafx.h"  
#include "SampleUseCashalotCOMAPI_CPP.h"  
#include "SampleUseCashalotCOMAPI_CPPDlg.h"  
#include "afxdialogex.h"  
#include <string> // std::string  
#include <sstream> // std::ostringstream  
#include <algorithm>  
#include <list>  
  
#ifdef _DEBUG
```

```

#define new DEBUG_NEW
#endif

#include "CashaLotApi.tlh"
using namespace CashaLotApi;

extern LPCTSTR pAppName;

CSampleUseCashaLotCOMAPICPPDlg::CSampleUseCashaLotCOMAPICPPDlg(CWnd* pParent /*=nullptr*/)
: CDialogEx(IDD_SAMPLEUSECASHALOTCOMAPI_CPP_DIALOG, pParent)
, m_FiscalNumberRRO(_T(""))
, m_PathToCashaLot(_T(""))
, m_IsNoInterfaceMode(FALSE)
, m_PathCert(_T(""))
, m_pwdCert(_T(""))
, m_PrinterName(_T(""))
, m_useToken(FALSE)
, m_UseDefCashalotPrinter(FALSE)
, m_PrintUseDefWindowsPrinter(FALSE)
, m_PrintUseNamedPrinter(FALSE)
, m_DateBeg(COleDateTime::GetCurrentTime())
, m_DateEnd(COleDateTime::GetCurrentTime())
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CSampleUseCashaLotCOMAPICPPDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
    DDX_Control(pDX, IDC_EDPATH, m_edPathToCashaLot);
    DDX_Control(pDX, IDC_EDFISKNUMRCP, m_edFiscNumRcp);
    DDX_Text(pDX, IDC_EDPATH2, m_FiscalNumberRRO);
    DDX_Text(pDX, IDC_EDPATH, m_PathToCashaLot);
    DDX_Control(pDX, IDC_EDPATH2, m_edFiscalNumberRRO);
    DDX_Control(pDX, IDC_BTGETPATH, m_btnPath);
    DDX_Check(pDX, IDC_CHECKNOINTERFACEMODE, m_IsNoInterfaceMode);
    DDX_Text(pDX, IDC_EDPATHTOCERT, m_PathCert);
    DDX_Text(pDX, IDC_EDPWD, m_pwdCert);
    DDX_Control(pDX, IDC_RADIO1, m_rdbtPrintUseNamedPrinter);
    DDX_Control(pDX, IDC_RADIO2, m_rdbtPrintUseDefPrinter);

    DDX_Text(pDX, IDC_EDPATH5, m_PrinterName);
    DDX_Control(pDX, IDC_EDPATHTOCERT, m_edPathCert);
    DDX_Control(pDX, IDC_EDPWD, m_edPwdCert);
    DDX_Control(pDX, IDC_CHECKNOINTERFACEMODE, m_chNoInterfaceMode);
    DDX_Check(pDX, IDC_CHECKUSETOKEN, m_useToken);
    DDX_Control(pDX, IDC_CHECKUSETOKEN, m_chUseToken);
    DDX_Check(pDX, IDC_CHECKUSECASHALOTPRINTERSETTINGS, m_UseDefCashalotPrinter);
    DDX_Control(pDX, IDC_CHECKUSECASHALOTPRINTERSETTINGS, m_chUseDefCashalotPrinter);
    DDX_Control(pDX, IDC_EDPATH5, m_edPrinterName);
    DDX_Radio(pDX, IDC_RADIO2, m_PrintUseDefWindowsPrinter);
    DDX_Radio(pDX, IDC_RADIO1, m_PrintUseNamedPrinter);
    DDX_DateTimeCtrl(pDX, IDC_DATETIMEPICKER1, m_DateBeg);
    DDX_DateTimeCtrl(pDX, IDC_DATETIMEPICKER2, m_DateEnd);
}

BEGIN_MESSAGE_MAP(CSampleUseCashaLotCOMAPICPPDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BTOPENSHIFT, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtopenshift)
    ON_BN_CLICKED(IDC_BTcloseShift, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtcloseshift)
    ON_BN_CLICKED(IDC_BTfiscalizeReceipt,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtfiscalizereceipt)

```

```

    ON_BN_CLICKED(IDC_BTReturnReceipt,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtreturnreceipt)
    ON_BN_CLICKED(IDC_BTServiceInput, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceinput)
    ON_BN_CLICKED(IDC_BTServiceOutput,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceoutput)
    ON_BN_CLICKED(IDC_BTServiceOutput2, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedPrintXRep)
    ON_WM_DESTROY ()
    ON_BN_CLICKED(IDC_BTGETPATH, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpath)
    ON_BN_CLICKED(IDC_BTGETPATHTOCERT,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpathtocert)
    ON_BN_CLICKED(IDC_BTPrintXRepEx, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintxrepex)
    ON_BN_CLICKED(IDC_CHECKUSECASHALOTPRINTERSETTINGS,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedCheckusecshalotprintersettings)
    ON_BN_CLICKED(IDC_RADIO2, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio2)
    ON_BN_CLICKED(IDC_RADIO1, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio1)
    ON_BN_CLICKED(IDC_BTGETPRINTERNAME,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetprintername)
    ON_BN_CLICKED(IDC_BTPrintLastZRep,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintlastzrep)
    ON_BN_CLICKED(IDC_BTGETRECEIPTANDREPFROMPERIOD,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedGetRcpByPeriod)
END_MESSAGE_MAP ()

// CSampleUseCashaLotCOMAPICPPDlg message handlers

BOOL CSampleUseCashaLotCOMAPICPPDlg::OnInitDialog ()
{
    CDialogEx::OnInitDialog ();

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here
    CWinApp* pApp = AfxGetApp ();
    m_PathToCashaLot = pApp->GetProfileString(pAppName, _T("PathToCashaLot"));
    m_FiscalNumberRRO = pApp->GetProfileString(pAppName, _T("FiscalNumberRRO"));
    m_PathCert = pApp->GetProfileString(pAppName, _T("PathToCert"));
    m_pwdCert = pApp->GetProfileString(pAppName, _T("PwdToCert"));
    m_IsNoInterfaceMode = pApp->GetProfileString(pAppName, _T("NoInterfaceMode")) == L"1";
    m_useToken = pApp->GetProfileString(pAppName, _T("UseToken")) == L"1";

    m_PrintUseDefWindowsPrinter = (pApp->GetProfileString(pAppName,
_T("PrintUseDefWindowsPrinter")) == L"0")? 0 : -1;
    m_PrintUseNamedPrinter = (pApp->GetProfileString(pAppName, _T("PrintUseNamedPrinter")) ==
L"0")? 0 : -1;;
    m_PrinterName = pApp->GetProfileString(pAppName, _T("PrinterName"));

    m_edPrinterName.SetWindowText(m_PrinterName);

    CString defPrndPrm = pApp->GetProfileString(pAppName, _T("UseDefCshalotPrinter"));
    if (defPrndPrm == L"") defPrndPrm = L"1";
    m_UseDefCshalotPrinter = defPrndPrm == L"1";

    m_chUseDefCshalotPrinter.SetCheck(m_UseDefCshalotPrinter);

    m_chNoInterfaceMode.SetCheck(m_IsNoInterfaceMode);
    m_chUseToken.SetCheck(m_useToken);

    m_edFiscalNumberRRO.SetWindowText(m_FiscalNumberRRO);
    m_edPathToCashaLot.SetWindowText(m_PathToCashaLot);
    m_edPathCert.SetWindowText(m_PathCert);
    m_edPwdCert.SetWindowText(m_pwdCert);
    UpdateData (0);
}

```

```

OnBnClickedCheckusecashalotprintersettings();
OnBnClickedRadio2();

return TRUE; // return TRUE unless you set the focus to a control
}
void CSampleUseCashaLotCOMAPICPPDlg::OnDestroy()
{
    UpdateData();
    CDialogEx::OnDestroy();

    CWinApp* pApp = AfxGetApp();
    pApp->WriteProfileString(pAppName, _T("PathToCashaLot"), m_PathToCashaLot);
    pApp->WriteProfileString(pAppName, _T("FiscalNumberRRO"), m_FiscalNumberRRO);
    pApp->WriteProfileString(pAppName, _T("PathToCert"), m_PathCert);
    pApp->WriteProfileString(pAppName, _T("PwdToCert"), m_pwdCert);
    pApp->WriteProfileString(pAppName, _T("NoInterfaceMode"), ((m_IsNoInterfaceMode == TRUE)? L"1"
: L"0"));
    pApp->WriteProfileString(pAppName, _T("UseToken"), ((m_useToken == TRUE)? L"1" : L"0"));
    pApp->WriteProfileString(pAppName, _T("UseDefCashaLotPrinter"), ((m_UseDefCashaLotPrinter ==
TRUE)? L"1" : L"0"));

    pApp->WriteProfileString(pAppName, _T("PrintUseDefWindowsPrinter"),
(m_PrintUseDefWindowsPrinter == 0 ? L"0" : L"-1"));
    pApp->WriteProfileString(pAppName, _T("PrintUseNamedPrinter"), (m_PrintUseNamedPrinter == 0 ?
L"0" : L"-1"));
    pApp->WriteProfileString(pAppName, _T("PrinterName"), m_PrinterName );
}

void CSampleUseCashaLotCOMAPICPPDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    CDialogEx::OnSysCommand(nID, lParam);
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CSampleUseCashaLotCOMAPICPPDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

// The system calls this function to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CSampleUseCashaLotCOMAPICPPDlg::OnQueryDragIcon()

```

```

{
    return static_cast<HCURSOR>(m_hIcon);
}

static int CALLBACK BrowseCallbackProc(HWND hwnd, UINT uMsg, LPARAM lParam, LPARAM lpData)
{
    if (uMsg == BFFM_INITIALIZED)
    {
        std::string tmp = (const char *)lpData;

        SendMessage(hwnd, BFFM_SETSELECTION, TRUE, lpData);
    }

    return 0;
}

CString BrowseFolder()
{
    TCHAR path[MAX_PATH] = { 0 };

    const char * path_param = "";

    BROWSEINFO bi = { 0 };
    bi.lpszTitle = _T("Выбор каталога в программе Cashalot");
    bi.ulFlags = BIF_RETURNONLYFSDIRS | BIF_NEWDIALOGSTYLE;
    bi.lpfnc = BrowseCallbackProc;
    bi.lParam = (LPARAM)path_param;

    LPITEMIDLIST pidl = SHBrowseForFolder(&bi);

    if (pidl != 0)
    {
        //get the name of the folder and put it in path
        SHGetPathFromIDList(pidl, path);

        //free memory used
        IMalloc * imalloc = 0;
        if (SUCCEEDED(SHGetMalloc(&imalloc)))
        {
            imalloc->Free(pidl);
            imalloc->Release();
        }

        return CString(path);
    }

    return CString("");
}

void CSampleUseCashalotCOMAPICPPDlg::OnBnClickedCheckusecashalotprintersettings()
{
    UpdateData();

    m_rdbtPrintUseDefPrinter.EnableWindow(!m_UseDefCashalotPrinter);
    m_rdbtPrintUseNamedPrinter.EnableWindow(!m_UseDefCashalotPrinter);
    m_edPrinterName.EnableWindow(!m_UseDefCashalotPrinter);
}

void CSampleUseCashalotCOMAPICPPDlg::OnBnClickedRadio2()
{
    UpdateData();
    m_rdbtPrintUseDefPrinter.SetCheck(m_PrintUseDefWindowsPrinter == 0);
    m_rdbtPrintUseNamedPrinter.SetCheck(m_PrintUseDefWindowsPrinter == -1);
}

```

```

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio1 ()
{
    UpdateData ();
    m_rdbtPrintUseDefPrinter.SetCheck(m_PrintUseDefWindowsPrinter == -1);
    m_rdbtPrintUseNamedPrinter.SetCheck(m_PrintUseDefWindowsPrinter == 0);
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetprintername ()
{
    CPrintDialog dlg(true, PD_NOPAGENUMS | PD_HIDEPRINTTOFILE | PD_NOCURRENTPAGE |
PD_NONETWORKBUTTON);
    if (dlg.DoModal () == IDOK)
    {
        m_PrinterName = dlg.GetDeviceName ();
        UpdateData (0);
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpath ()
{
    CString ret = BrowseFolder ();
    if (ret.GetLength ())
    {
        m_PathToCashaLot = ret;
        m_edPathToCashaLot.SetWindowText (m_PathToCashaLot);
    }
}

/////////////////////////////////ПРИКЛАД ВИКОРИСТАННЯ/////////////////////////////////
struct CashaLotApiRsp
{
    void Clear ()
    {
        Ret = false;
        ErrorString = "";
        Values.clear ();
    }

    CashaLotApiRsp ()
    {
        Clear ();
    }
    bool Ret ;
    std::string ErrorString ;
    std::list< std::pair<std::string, std::string> > Values;
};

CashaLotApiRsp rspEx;
CashaLotApiRsp DeserializeResp(std::string response)
{
    rspEx.Clear ();
    return rspEx;
}

//Структура інформації по позиції товару в чеці(артикул, назва сумма, кількість, податки і тд.)
struct ReceiptItemInfo
{
    ReceiptItemInfo ()
    {
        VendorCode = "";
        Name = "";
        CdUKDZED = "";
        CdDKPP = "";
    }
}

```

```

Barcode = "";
GoodsType = 1; //1- товар, 2- услуга
UnitType = "шт";
Quantity = 0;
Price = 0;
Amount = 0;
DiscountPrc = 0;
DiscountSum = 0;
VATRate = 0;
IsPriceIncludeVAT = true;
SumVAT = 0;
IsExcise = false;
OtherParameters = "";
}

std::string VendorCode;
std::string Name;
std::string CdUKDZED;
std::string CdDKPP;
std::string Barcode;
int GoodsType; //1- товар, 2- услуга
std::string UnitType;
double Quantity;
double Price;
double Amount;
double DiscountPrc;
double DiscountSum;
double VATRate;
bool IsPriceIncludeVAT;
double SumVAT;
bool IsExcise;
std::string OtherParameters;

//Повертає JSON рядок з інформацією по позиції товару
std::string ToJSON()
{
    std::ostream ret;

    ret << "{" <<
        "\"VendorCode\": \"" << (VendorCode.c_str()) <<
        "\", \"Name\": \"" << (Name.c_str()) <<
        "\", \"CdUKDZED\": \"" << (CdUKDZED.c_str()) <<
        "\", \"CdDKPP\": \"" << (CdDKPP.c_str()) <<
        "\", \"Barcode\": \"" << (Barcode.c_str()) <<
        "\", \"GoodsType\": \"" << GoodsType <<
        "\", \"UnitType\": \"" << (UnitType.c_str()) <<
        "\", \"Quantity\": \"" << Quantity <<
        "\", \"Price\": \"" << Price <<
        "\", \"Amount\": \"" << Amount <<
        "\", \"DiscountPrc\": \"" << DiscountPrc <<
        "\", \"DiscountSum\": \"" << DiscountSum <<
        "\", \"VATRate\": \"" << VATRate <<
        "\", \"IsPriceIncludeVAT\": \"" << ((!IsPriceIncludeVAT) ? "false" : "true") <<
        "\", \"SumVAT\": \"" << SumVAT <<
        "\", \"IsExcise\": \"" << ((!IsExcise) ? "false": "true") <<
        "\", \"OtherParameters\": \"" << (OtherParameters.c_str()) << "\""
        << "}"
        ;

    return ret.str();
}
};

struct ReceiptInfoLst
{
private:

```



```

std::list<ReceiptItemInfo*> lstReceiptItemsInfo;
std::string Comment;
public:
ReceiptInfoLst()
{
    Clear();
};
~ReceiptInfoLst()
{
    Clear();
};

std::string ToJSON()
{
    std::ostringstream ret;
    std::ostringstream comment;
    long cnt=0;

    if(Comment.length() > 0)
        comment << ", \"Comment\": \"" << Comment.c_str() << "\"";

    ret << "{\"ReceiptLst\":[";

    for(std::list<ReceiptItemInfo*>::iterator x= lstReceiptItemsInfo.begin(); x!=
lstReceiptItemsInfo.end(); ++x)
    {
        if(cnt++!=0)
            ret << ",";
        ret << ((*x)->ToJSON());
    }
    ret << "]"<< (comment.str().c_str()) <<"}";

    return ret.str();
}

_bstr_t ToBSTRJSON()
{
    return _bstr_t(ToJSON().c_str());;
}

void Clear()
{
    Comment.clear();
    for (std::list<ReceiptItemInfo*>::iterator x = lstReceiptItemsInfo.begin(); x !=
lstReceiptItemsInfo.end(); ++x)
        delete (*x);
    lstReceiptItemsInfo.clear();
}

void InsertItem( ReceiptItemInfo* check )
{
    lstReceiptItemsInfo.push_back(check);
}

void SetComment(std::string strComment)
{
    Comment = strComment;
}
};

//Структура оплати по чеку
struct PayInfoRsp
{
    //Сума готівки наданої покупцем(може бути більшою за сумму до сплати, з неї вираховується
решта/здача)
    double SumCash;
    //Сума оплати картою

```

```

double SumPayByCard;
//Сума оплати за рахунок кредитних коштів
double SumPayByCredit;
//Сума оплати сертифікатом
double SumPayByCertificate;
// Сума фактичної оплати всіма видами оплати, з урахуванням заокруглення
double SumPayCheck;
// Email платника
std::string CustomerEmail;

PayInfoRsp()
{
    //Сума готівки наданої покупцем(може бути більшою за сумму до сплати, з неї вираховується
решта/здача)
    SumCash =
    SumPayByCard =
    SumPayByCredit =
    SumPayByCertificate =
    SumPayCheck = 0;
    CustomerEmail = "";
}

//Отримати JSON рядок з інформацією про оплату по чеку(готівка, карта, знижки та заокруглення)
std::string ToJSON()
{
    std::ostringstream ret;
    long cnt = 0;
    ret << "{\"SumCash\":\"" << SumCash <<
        "\",\"SumPayByCard\":\"" << SumPayByCard <<
        "\",\"SumPayByCredit\":\"" << SumPayByCredit <<
        "\",\"SumPayByCertificate\":\"" << SumPayByCertificate <<
        "\",\"SumPayCheck\":\"" << SumPayCheck <<
        "\",\"CustomerEmail\":\"" << CustomerEmail << "\"}";

    return ret.str();
}

_bstr_t ToBSTRJSON()
{
    return _bstr_t(ToJSON().c_str());
}
};

std::ostringstream strRet;
ReceiptInfoLst gsLstCheckItems;
ICashalotApiRetValPtr retVal = NULL;
ICashaLotApiAddinPtr gsCashaLotApi = NULL;
BSTR fiscalNumRRO = NULL;
BSTR pathToCashaLot = NULL;
BSTR pathToCert = NULL;
BSTR pwdToCert = NULL;
bool IsNoInterfaceMode = false;
bool useToken = false;
int setupPrinterMode = 0;
BSTR printerName = NULL;

void SetPrinterParametr()
{
    if (gsCashaLotApi != NULL)
    {
        gsCashaLotApi->SetParameter( L"DEFAULTPRINTERNAME", (setupPrinterMode == 0)? BSTR(L"") :
((setupPrinterMode == 1)? BSTR(L"DEFAULTWINDOWS") : printerName) );
    }
}

//Ініціалізація COM об'єкта CashaLot

```

```

bool InitCashaLotApiComInterface ()
{
    if ( gsCashaLotApi == NULL )
    {
        // Вказуємо ІД COM об'єкта AddIn.CashaLotApi, який має бути попередньо зареєстрований в
        реєстрі(regsvr32 D:\prro\CashLot\CashalotApi64.dll)
        char* szProgID = "AddIn.CashaLotApi";
        WCHAR  szWideProgID[128];
        CLSID  clsid;
        long lLen = MultiByteToWideChar(CP_ACP,
            0,
            szProgID,
            strlen(szProgID),
            szWideProgID,
            sizeof(szWideProgID));

        szWideProgID[lLen] = '\0';
        HRESULT hr = ::CLSIDFromProgID(szWideProgID, &clsid);
        if (FAILED(hr))
        {
            MessageBox(0, L"Unable to get CLSID from ProgID. AddIn.CashaLotApi", 0, 0);
            return false;
        }

        IClassFactory* pCF;
        // Отримати фабрику класів
        hr = CoGetClassObject(clsid,
            CLSCTX_INPROC,
            NULL,
            IID_IClassFactory,
            (void**)&pCF);
        if (FAILED(hr))
        {
            MessageBox(0, L"Failed to GetClassObject server instance.", 0, 0);
            return false;
        }

        // за допомогою фабрики класів створити екземпляр
        // компонента та отримати інтерфейс IUnknown.
        IUnknown* pUnk;
        hr = pCF->CreateInstance(NULL, IID_IUnknown, (void**)&pUnk);

        // Release the class factory
        pCF->Release();

        if (FAILED(hr))
        {
            MessageBox(0, L"Failed to create server instance.", 0, 0);
            return false;
        }

        hr = pUnk->QueryInterface(__uuidof(ICashaLotApiAddin), (LPVOID*)&gsCashaLotApi);
        pUnk->Release();

        if (FAILED(hr))
        {
            MessageBox(0, L"Failed to create ICashaLotApiAddin", 0, 0);
            return false;
        }

        //-----Задаємо обов'язкові параметри(шлях до каталогу з програмою Кашалот, та фіскальний
        номер РРО з якою будемо працювати) -----

        gsCashaLotApi->SetParameter(L"PathToCashalotDir", pathToCashaLot);
        gsCashaLotApi->SetParameter(L"DeviceIDFnRRO", fiscalNumRRO);
        gsCashaLotApi->SetParameter(L"NOINTERFACEMODE", IsNoInterfaceMode ? L"true" : L"false");
    }
}

```

```

        gsCashaLotApi->SetParameter(L"NOAUTOOPENSHIFT", L"true");
        //Ознака використанні захищеного носія(має бути після PathToCshalotDir та DeviceIDFnRRO
та NOINTERFACEMODE, якщо необхідно пришвидшити завантаження, та немає необхідності використовувати
токени. Інакше токени будуть ініціалізуватися)
        gsCashaLotApi->SetParameter(L"USETOKEN", useToken ? L"true" : L"false");

        SetPrinterParametr();

        gsCashaLotApi->SetParameter(L"PathToCertificate", pathToCert);
        gsCashaLotApi->SetParameter(L"PwdToCertificate", pwdToCert);

    }

    return true;
}

bool CSampleUseCashaLotCOMAPICPPDlg::InitCashaLotApiComInterfaceEx()
{
    UpdateData();

    if (useToken != m_useToken && gsCashaLotApi != NULL)
        gsCashaLotApi->SetParameter(L"USETOKEN", m_useToken ? L"true" : L"false");

    useToken = m_useToken;

    if ( pathToCashaLot == NULL )
    {
        pathToCashaLot = ::SysAllocString(m_PathToCashaLot);
    }

    if (fiscalNumRRO == NULL )
    {
        fiscalNumRRO = ::SysAllocString(m_FiscalNumberRRO);
    }

    if (pathToCert != NULL && pathToCert != m_PathCert)
    {
        ::SysFreeString(pathToCert);
        pathToCert = ::SysAllocString(m_PathCert);
        if (gsCashaLotApi != NULL)
            gsCashaLotApi->SetParameter(L"PathToCertificate", pathToCert);
    }
    else
    if (pathToCert == NULL )
    {
        pathToCert = ::SysAllocString(m_PathCert);
    }

    if (pwdToCert != NULL && pwdToCert != m_pwdCert)
    {
        ::SysFreeString(pwdToCert);
        pwdToCert = ::SysAllocString(m_pwdCert);
        if (gsCashaLotApi != NULL)
            gsCashaLotApi->SetParameter(L"PwdToCertificate", pwdToCert);
    }
    else
    if (pwdToCert == NULL)
    {
        pwdToCert = ::SysAllocString(m_pwdCert);
    }

    int oldSetupPrinterMode = setupPrinterMode;
    setupPrinterMode = (m_UseDefCshalotPrinter)? 0 : m_PrintUseDefWindowsPrinter!=-1? 1 : 2 ;

    if (printerName != NULL && (printerName != m_PrinterName || oldSetupPrinterMode !=
setupPrinterMode) )
    {

```

```

        ::SysFreeString(printerName);
        printerName = ::SysAllocString(m_PrinterName);
        SetPrinterParametr();
    }
    else
    if(printerName == NULL)
        printerName = ::SysAllocString(m_PrinterName);

    if(IsNoInterfaceMode != m_IsNoInterfaceMode && gsCashaLotApi != NULL )
        gsCashaLotApi->SetParameter(L"NOINTERFACEMODE", m_IsNoInterfaceMode ? L"true" : L"false");

    IsNoInterfaceMode = m_IsNoInterfaceMode;

    return InitCashaLotApiComInterface();
}

//Очистка всіх ресурсів по роботі з COM об'єктом CashaLot
bool TermCashaLotApiComInterface()
{
    try
    {
        if (gsCashaLotApi != NULL)
        {
            gsCashaLotApi->Release();
            gsCashaLotApi = NULL;

            if (pathToCashaLot != NULL)
            {
                ::SysFreeString(pathToCashaLot);
                pathToCashaLot = NULL;
            }
            if (fiscalNumRRO != NULL)
            {
                ::SysFreeString(fiscalNumRRO);
                fiscalNumRRO = NULL;
            }
            if (pathToCert != NULL)
            {
                ::SysFreeString(pathToCert);
                pathToCert = NULL;
            }
            if (pwdToCert != NULL)
            {
                ::SysFreeString(pwdToCert);
                pwdToCert = NULL;
            }
            return true;
        }
    }
    catch (const char* msg)
    {
        MessageBox(0, BSTR(msg), L"Exception", MB_OK);
    }
    catch (...)
    {
        MessageBox(0, L"", L"Exception", MB_OK);
    }

    return false;
}

//Приклад відкриття зміни
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtopenshift()
{

```

```

if (!InitCashaLotApiComInterfaceEx ())
return;

gsCashaLotApi->SetParameter(L"NOINTERFACEMODE", IsNoInterfaceMode ? L"true" : L"false");

if (pathToCert != NULL)
{
    ::SysFreeString(pathToCert);
    pathToCert = NULL;
}
if (pwdToCert != NULL)
{
    ::SysFreeString(pwdToCert);
    pwdToCert = NULL;
}

pathToCert = ::SysAllocString(m_PathCert);
pwdToCert = ::SysAllocString(m_pwdCert);

gsCashaLotApi->SetParameter(L"PathToCertificate", pathToCert);
gsCashaLotApi->SetParameter(L"PwdToCertificate", pwdToCert);

_variant_t ret = gsCashaLotApi->OpenShift(fiscalNumRRO);

if (V_VT(&ret) == VT_DISPATCH)
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        m_edPathToCashaLot.EnableWindow(false);
        m_btnPath.EnableWindow(false);
        m_edFiscalNumberRRO.EnableWindow(false);

        strRet.str("");
        strRet << "Номер відкритої зміни: " << retVal->ShiftID;
        MessageBox(CString(strRet.str().c_str()));
    }
}
}

//Приклад закриття зміни
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtccloseshift()
{
    if (!InitCashaLotApiComInterfaceEx ())
        return;

    _variant_t ret = gsCashaLotApi->CloseShift(fiscalNumRRO);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet.str("");
            strRet << "Номер закритої зміни: " << retVal->ShiftID<<"\n"
                << "Результат у вигляді JSON\n" << retVal->JsonVal;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

//Приклад фіскалізації чека на дві позиції
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtfiscalizereceipt()
{
    if (!InitCashaLotApiComInterfaceEx ())
        return;
}

```

```

gsLstCheckItems.Clear();

ReceiptItemInfo* inf = new ReceiptItemInfo();
inf->VendorCode = "АртикулТовару1";
inf->Name = "НазваТовару1";
inf->Barcode = "123456789";
inf->UnitType = "шт";
inf->Quantity = 1;
inf->GoodsType = 1;/*товар*/
inf->Price = 183;
inf->Amount = 181.17;
inf->DiscountSum = 1.83;
inf->VATRate = 20;
inf->SumVAT = 30.2;
inf->IsExcise = false;
gsLstCheckItems.InsertItem(inf);

inf = new ReceiptItemInfo();
inf->VendorCode = "АртикулПослуги1";
inf->Name = "НазваПослуги1";
inf->Barcode = "923456780";
inf->UnitType = "шт";
inf->Quantity = 1;
inf->GoodsType = 2;/*послуга*/
inf->Price = 183;
inf->Amount = 181.17;
inf->DiscountSum = 1.83;
inf->VATRate = 20;
inf->SumVAT = 30.2;
inf->IsExcise = false;
gsLstCheckItems.InsertItem( inf );
gsLstCheckItems.SetComment("Тестовий рядок1\r\nТестовий рядок2\r\nТестовий рядок3");

_bstr_t jsonChkData = gsLstCheckItems.ToBSTRJSON();

PayInfoRsp payInfo;
payInfo.SumCash = 366;
payInfo.SumPayCheck = 362.34;
payInfo.SumPayByCredit = 100;
payInfo.SumPayByCertificate = 50;
_bstr_t jsonPayData = payInfo.ToBSTRJSON();

_variant_t ret = gsCashaLotApi->FiscalizeCheck(fiscalNumRRO, jsonChkData, jsonPayData);

if (V_VT(&ret) == VT_DISPATCH)
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        m_edFiscNumRcp.SetWindowTextW(retVal->ReceiptFiscalNum);

        strRet.str("");
        strRet << "Фіскальний номер створеного чека: " << retVal->ReceiptFiscalNum;
        strRet << "\nXML створеного чека: \n" << retVal->XML;
        MessageBox(CString(strRet.str().c_str()));
    }
}

gsLstCheckItems.Clear();
}

//Приклад фіскалізації чека повернення на дві позиції, з вказанням чека по якому здійснюється
повернення
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtreturnreceipt()
{

```

```

CString strfiscNumRcpToRet;
m_edFiscNumRcp.GetWindowText(strfiscNumRcpToRet);

if (strfiscNumRcpToRet.GetLength() == 0)
{
    MessageBox(L"Не заданий фiскальний номер чека по якому здiйснюється повернення!");
    return;
}

if (!InitCashaLotApiComInterfaceEx())
    return;

gsLstCheckItems.Clear();
ReceiptItemInfo* inf = new ReceiptItemInfo();
inf->VendorCode = "АртикулТовару1";
inf->Name = "НазваТовару1";
inf->Barcode = "123456789";
inf->UnitType = "шт";
inf->Quantity = 1;
inf->GoodsType = 1; /*товар*/
inf->Price = 183;
inf->Amount = 181.17;
inf->DiscountSum = 1.83;
inf->VATRate = 20;
inf->SumVAT = 30.2;
inf->IsExcise = false;
gsLstCheckItems.InsertItem(inf);

inf = new ReceiptItemInfo();
inf->VendorCode = "АртикулПослуги1";
inf->Name = "НазваПослуги1";
inf->Barcode = "923456780";
inf->UnitType = "шт";
inf->Quantity = 1;
inf->GoodsType = 2; /*послуга*/
inf->Price = 183;
inf->Amount = 181.17;
inf->DiscountSum = 1.83;
inf->VATRate = 20;
inf->SumVAT = 30.2;
inf->IsExcise = false;
gsLstCheckItems.InsertItem(inf);
_bstr_t jsonChkData = gsLstCheckItems.ToBSTRJSON();

PayInfoRsp payInfo;
payInfo.SumCash = 366;
payInfo.SumPayCheck = 362.34;
_bstr_t jsonPayData = payInfo.ToBSTRJSON();

_bstr_t retFiscNum = ::SysAllocString(strfiscNumRcpToRet);

_variant_t ret = gsCashaLotApi->FiscalizeReturnCheck(fiscalNumRRO, jsonChkData, jsonPayData,
retFiscNum );

if ( V_VT(&ret) == VT_DISPATCH )
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        strRet.str("");
        strRet << "Фiскальний номер створеного видаткового чека: " <<
retVal->ReceiptFiscalNum;
        //strRet << "\nXML створеного видаткового чека: \n" << retVal->XML;
        strRet << "\nТип чека: \n" << retVal->Type;
        MessageBox(CString(strRet.str().c_str()));
    }
}

```



```

}

::SysFreeString(retFiscNum);
}

//Приклад чека службового внесення
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceinput ()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;
    BSTR strSm = ::SysAllocString(L"100");

    VARIANT vSm;
    ::VariantInit(&vSm);
    V_VT(&vSm) = VT_BSTR;
    V_BSTR(&vSm) = strSm;

    _variant_t ret = gsCashaLotApi->ServiceInput(fiscalNumRRO, vSm);

    if (V_VT(&ret) == VT_BOOL && V_BOOL(&ret) != FALSE )
        MessageBox(L"Службове внесення виконано успішно");
    else
        MessageBox(L"Не вдалося виконати службове внесення");

    ::SysFreeString(strSm);
}

//Приклад чека службової видачі
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceoutput ()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;
    BSTR strSm = ::SysAllocString(L"100");

    VARIANT vSm;
    ::VariantInit(&vSm);
    V_VT(&vSm) = VT_BSTR;
    V_BSTR(&vSm) = strSm;
    _variant_t ret = gsCashaLotApi->ServiceOutput(fiscalNumRRO, vSm);

    if (V_VT(&ret) == VT_BOOL && V_BOOL(&ret) != FALSE)
        MessageBox(L"Службова видача виконана успішно");
    else
        MessageBox(L"Не вдалося виконати службову видачу");

    ::SysFreeString(strSm);
}

//Приклад чека друку X-Звіту(без гасіння)
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedPrintXRep ()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->PrintXReport(fiscalNumRRO);

    if (V_VT(&ret) == VT_BOOL && V_BOOL(&ret) != FALSE)
        MessageBox(L"X-звіт сформований успішно");
    else
        MessageBox(L"Не вдалося сформувати X-звіт");
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpathtocert ()
{
    CString ret = BrowseFolder();
}

```

```

    if (ret.GetLength())
    {
        m_PathCert = ret;
        m_edPathCert.SetWindowText(m_PathCert);
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintxrepex()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->GetXReport(fiscalNumRRO, VARIANT_FALSE);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet.str("");
            strRet << "Результат у вигляді JSON:\n" << retVal->JsonVal;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintlastzrep()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->GetLastZReport(fiscalNumRRO, VARIANT_TRUE);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet.str("");
            strRet << "Результат у вигляді JSON:\n" << retVal->JsonVal;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedGetRcpByPeriod()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    strRet.str("");
    BSTR dtBeg = ::SysAllocString(m_DateBeg.Format(CString("%d.%m.%Y")));
    BSTR dtEnd = ::SysAllocString(m_DateEnd.Format(CString("%d.%m.%Y")));
    //зчитуємо перший чек за період
    _variant_t ret = gsCashaLotApi->GetFirstReceiptByPeriod(fiscalNumRRO, dtBeg, dtEnd);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet << "Фіскальні номери чеків за період:\n";
            strRet << retVal->ReceiptFiscalNum;

            bool retN = false;

```

```

do
{
    retN = false;
    //зачитуємо наступний чек за період
    ret = gsCashaLotApi->GetNextReceipt(fiscalNumRRO);
    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            retN = true;
            strRet << ",";
            strRet << retVal->ReceiptFiscalNum;
        }
    }
    //читаємо поки будуть наступні
} while (retN);
}

ret = gsCashaLotApi->GetFirstZReportByPeriod(fiscalNumRRO, dtBeg, dtEnd);

if (V_VT(&ret) == VT_DISPATCH)
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        strRet << "\nфіскальні номери Z-звітів за період:\n";
        strRet << retVal->ReceiptFiscalNum;

        bool retN = false;
        do
        {
            retN = false;
            //зачитуємо наступний чек за період
            ret = gsCashaLotApi->GetNextZReport(fiscalNumRRO);
            if (V_VT(&ret) == VT_DISPATCH)
            {
                retVal = (ICashalotApiRetVal*)ret.pdispVal;
                if (retVal != NULL && retVal->Return != FALSE)
                {
                    retN = true;
                    strRet << ",";
                    strRet << retVal->ReceiptFiscalNum;
                }
            }
            //читаємо поки будуть наступні
        } while (retN);
    }
}

MessageBox(CString(strRet.str().c_str()));

::SysFreeString(dtBeg);
::SysFreeString(dtEnd);
}

```

Варіант 26 – API ПРРО Cashalot з .NET інтерфейсом CashalotApiBridge.dll

Підключення ПРРО до розрахунково-облікової системи відбувається за допомогою розробленої API бібліотеки ПРРО Cashalot **CashalotApiBridge.dll**, шлях до якої необхідно вказати для коректного налаштування. Користувачу необхідно розробити функції, що будуть транслювати виклик методів із бібліотеки Cashalot.

Зверніть увагу! Виклик бібліотеки CashalotApiBridge.dll необхідно здійснювати виключно з кореневого каталогу встановленого екземпляру Cashalot.

В якості прикладу в документі наведено **C#** реалізацію у вигляді методу **CallCashalotMethod** (“Метод”, Параметру).

Оскільки будь-яка дія в ПРРО (відкриття/закриття зміни, відправка чеків тощо) здійснюється шляхом підписання блоку даних КЕП, то під час роботи необхідно вказати шлях до сертифікатів, вибрати користувача і здійснити авторизацію.

1. Специфікація методів API бібліотеки ПРРО Cashalot при підключенні до облікової системи

За допомогою наявних методів API бібліотека реалізує виконання наступних операцій ПРРО Cashalot:

- Відкриття зміни
- Службове внесення грошових коштів в касу
- Службове вилучення грошових коштів з каси
- Проведення розрахункових операцій продажу товарів/послуг
- Проведення операції повернення товару
- Визначення номеру фіскального чека за локальним номером чека в касі
- Створення X-Звіту
- Визначення касового локального номеру фіскального чека
- Закриття зміни та створення Z-Звіту

Виконання методів повертає об'єкт **CashalotApiRsp**, що містить інформацію про результат роботи:

- Ознака успішності виконання функції (**bool**);
- Текст помилки (**string**);
- Масив параметрів (**Dictionary<string, string>**) (опціонально).

Приклад:

```
public class CashalotApiRsp
{
    //Ознака успішності виконання функції (true - успішно/false - не успішно)
    public bool Ret { get; set; } = false;
    //Рядок з текстом помилки
    public string ErrorString { get; set; } = String.Empty;
    //Масив параметрів, які можуть повернути функції API
    public Dictionary<string, string> Values { get; set; }
}
```

Для користувачів також повертаються службові повідомлення або графічні відображення чеків, які можна відправити на друк.

2. Порядок роботи обліково-розрахункової системи з API бібліотекою ПРРО Cashalot

Перед початком роботи каси виконується операція відкриття зміни.

При відкритій касовій зміні доступні операції проведення розрахункових операцій, повернення товарів, службові внесення та видача грошових коштів, а також, створення Х-Звітів. Всі перелічені операції додатково дозволяють відправку документів на друк.

Для завершення касової зміни необхідно провести операцію закриття зміни, під час якої відбувається службове вилучення грошових коштів з каси і формується Z-Звіт.

Деякі методи ПРРО Cashalot, вказані у розділі [Загальна інформація](#), виконуються неявно під час обробки запитів від розрахунково-облікової системи, і можуть повертати відповідні службові повідомлення (перехід у офлайн-режим тощо).

3. Специфікація методів бібліотеки CashalotApiBridge.dll

1. Відкриття зміни

Для відкриття зміни необхідно викликати метод **"OpenShift"** з параметром:

```
{  
    Фіскальний номер РРО  
}
```

Специфікація методу:

```
/// <summary>  
/// Відкрити зміну  
/// </summary>  
/// <param name="rroFiscalNumber"> фіскальний номер РРО</param>  
/// <returns>  
/// Функція повертає структуру у вигляді JSON  
/// Приклад:  
/// {"Ret":true,"ErrorString":"","  
/// "Values":{"ShiftID":"e8353c00-f9eb-4b4c-aebc-ad0e55fb9109"}}  
/// </returns>  
public string OpenShift( string rroFiscalNumber );
```

Приклад:

```
Object[] parameters = {"1111111111111111"/*фіскальний номер РРО*/};  
var ret = CallCashalotMethod( "OpenShift", parameters );
```

Cashalot виконує запит щодо стану обраного фіскального номера каси, після чого створює XML повідомлення із призначеним локальним номером і типом «Відкриття зміни», засвідчує його кваліфікованим електронним підписом продавця і надсилає на ФСКО.

2. Відкриття зміни

Для відкриття зміни необхідно викликати метод **"OpenShift"** з параметром:

```
{  
    Фіскальний номер РРО  
}
```

Специфікація методу:

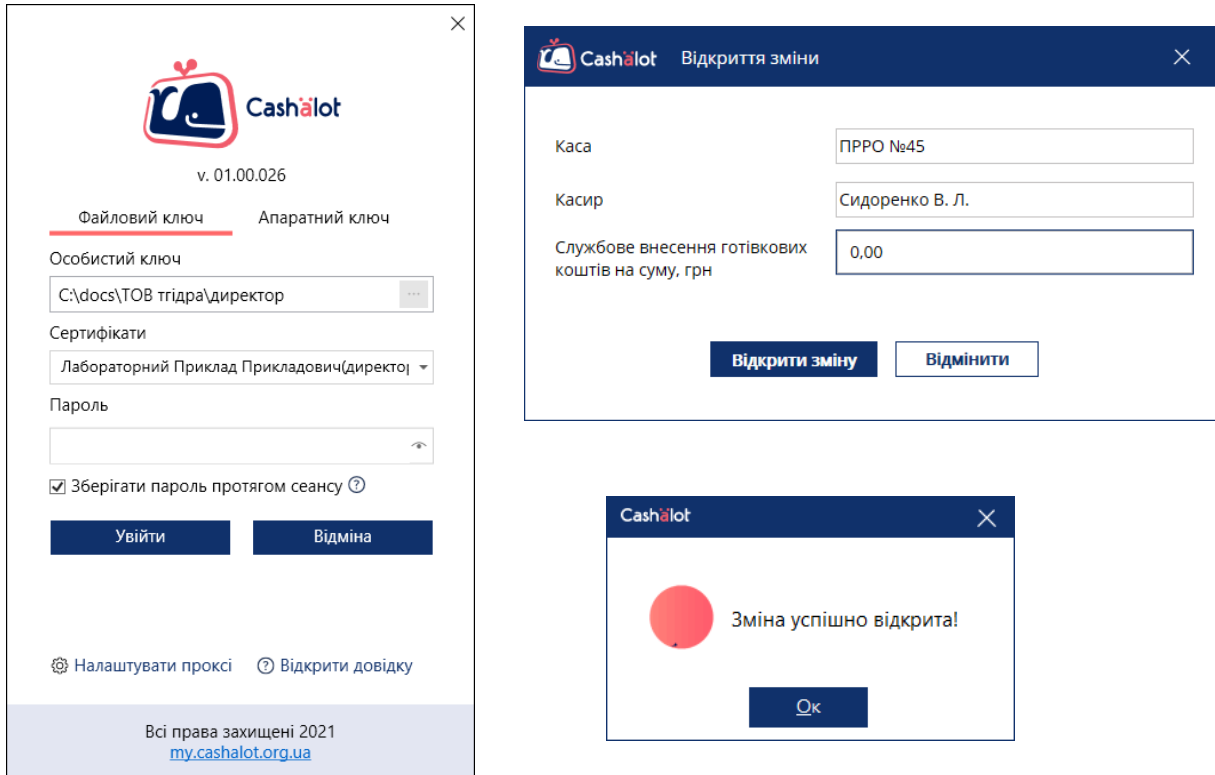
```
/// <summary>  
/// Відкрити зміну  
/// </summary>  
/// <param name="rroFiscalNumber"> фіскальний номер РРО</param>  
/// <returns>  
/// Функція повертає структуру у вигляді JSON  
/// Приклад:  
/// {"Ret":true,"ErrorString":"","  
/// "Values":{"ShiftID":"e8353c00-f9eb-4b4c-aebc-ad0e55fb9109"}}  
/// </returns>
```

```
public string OpenShift( string rroFiscalNumber );
```

Приклад:

```
Object[] parameters = {"11111111111111111111"}/*фіскальний номер РРО*/);  
var ret = CallCashalotMethod( "OpenShift", parameters );
```

Cashalot виконує запит щодо стану обраного фіскального номера каси, після чого створює XML повідомлення із призначеним локальним номером і типом «Відкриття зміни», засвідчує його кваліфікованим електронним підписом продавця і надсилає на ФСКО.



3. Службове внесення грошових коштів в касу

Для проведення операції службового внесення грошових коштів необхідно викликати метод **“ServiceInput”** з такими параметрами:

```
{  
    Фіскальний номер РРО ,  
    Сума службового внесення  
}
```

Специфікація методу:

```
/// <summary>  
/// Службове внесення готівки в касу  
/// </summary>  
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>  
/// <param name="sumVal"> Сума готівки </param>  
/// <returns>  
/// Функція повертає структуру у вигляді JSON  
/// Приклад:  
/// {"Ret":true,"ErrorString":"","Values":{}}  
/// </returns>  
public string ServiceInput( string rroFiscalNumber, string sumVal  
);
```

Приклад:

```
Object[] parameters = {"11111111111111111111"}, /* фіскальний номер РРО*/
```

```

        "5000.00" /* сума службового внесення*/
    };
    var ret = CallCashalotMethod( "ServiceInput", parameters );

```

Cashalot створює службове діалогове вікно для перевірки отриманих параметрів запиту і після підтвердження, відправляє відповідне XML повідомлення на Сервер ФСКО.

4. Службове вилучення (видача) грошових коштів з каси

Для проведення операції службового вилучення грошових коштів необхідно викликати метод **"ServiceOutput"** з такими параметрами:

```

{
    Фіскальний номер РРО ,
    Сума службової видачі
}

```

Специфікація методу:

```

/// <summary>
/// Службове виймання готівки з каси
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="sumVal"> Сума готівки </param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string ServiceOutput( string rroFiscalNumber, string sumVal);

```

Приклад:

```

Object[] parameters = {"1111111111111", /* фіскальний номер РРО */
    "5000.00" /* сума службового вилучення*/
};
var ret = CallCashalotMethod( "ServiceOutput", parameters );

```

Cashalot створює службове діалогове вікно для перевірки отриманих параметрів запиту і після підтвердження, відправляє відповідне XML повідомлення на Сервер ФСКО.

5. Проведення розрахункових операцій продажу товарів/послуг

Для виконання розрахункової операції продажу товарів або послуг необхідно викликати метод фіскалізації чеку **"FiscalizeCheck"**, в якості параметрів якого передаються 2 JSON рядки: **strJSONData** для списку покупок та **payDataJSON** для повної вартості покупок. Вказані JSON рядки мають містити наступну інформацію:

strJSONData

Структура містить масив ReceiptLst, кожен елемент якого містить параметри номенклатури, та об'єкти Comment і DocType.

Для кожного товару в чеку вказуються параметри у вигляді: параметр = значення:

Параметр	Значення	Приклад
VendorCode	Артикул товару	"24563"
Name	Найменування товару	"Бунчук гетьманський, сувенірний"
DocNumber	Номер договору (при використанні ломбардних послуг)	"01/a12 від 15.02.21"
CdUKDZED	Код УКДЗЕД	"1806903910"
CdDKPP	Код ДКПП	"47.00.69"
GoodsType	Тип товару: 1 – товар; 2 – послуга, 3 – послуга ломбардна *.	"1"

	За замовчуванням: "1"	
Barcode	Штрих-код	"123456789"
UnitType	Одиниця виміру. За замовчуванням: "шт"	"шт"
Quantity	Кількість	"1"
Price	Ціна (за одиницю товару)	"2000.00"
Amount	Вартість (сумарна, за всі одиниці товару, з врахуванням знижок)	"2000.00"
DiscountPrc	Сума знижки у відсотках (%)	"5.00"
DiscountSum	Сума знижки у грошових одиницях	"100.00"
IsPriceIncludeVAT	Чи включено ПДВ у ціну товару (true / false)	true
VATRate	Значення ПДВ у %	"20.00"
SumVAT	Значення ПДВ у грошових одиницях	"316.67"
IsExcise	Ознака підакцизного товару (true / false)	false
NEW PaymentOrderType	Спосіб оплати: 0-звичайна 1 - передплата 2 - післяплата 3 - інтернет продаж	"2"
NEW SumPreparePaid	Вказуємо суму передплати, вказуємо загальну суму по чеку.	"1500"
ParentReceiptFiscalNumber	фіскальний номер чеку по якому здійснюється доплата/повернення	"604779"
NEW SaleOrderNumber	Номер замовлення/договору, що відображається у структурній інформації про чек LstCheckInfoRsp	"555444333"

OtherParameters	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.	<?xml version="1.0" encoding="UTF-8"?><Table><Record VAT="20" IsExcisable="False" Nomenclature="СТЕР Пазл 104 Винни Пух" UKTZED="" VendorCode="82106" Barcode="" Dimension="шт" Discount="1" DiscountAmount="0.56" PriceIncludeVAT="1" AmountVAT="9.24"/></Table>
-----------------	---	---

* Використання ломбардних послуг описано в розділі "[Ломбардні послуги](#)".

Параметри повинні бути представлені у типі `string` (крім булевих параметрів `bool` `IsPriceIncludeVAT`, `IsExcise`), для уникнення помилок перетворення типів.

Опціональний об'єкт `Comment` використовується для додавання нефіскальної інформації:

Comment	Опціональний рядок нефіскальної інформації, коментар. Допускається використання одного коментарю на чек. За необхідності сформулювати коментар з декількох рядків, для переходу на новий рядок використовується комбінація <code>"\r"</code> або <code>"\n"</code> .
---------	--

Об'єкт `DocType` є обов'язковим при проведенні ломбардних операцій, призначений для визначення типу документа (чека). Ломбардні операції описані в розділі "[Ломбардні послуги](#)".

DocType	Тип документа 0 – чек продажу 4 – ломбардний чек За замовчуванням <code>DocType = 0</code> , якщо об'єкт не задано.
---------	--

payDataJSON

Для цього JSON рядка формується структура: параметр = значення:

Параметр	Значення	Приклад
SumCash	Сума готівки, надана покупцем	"1960.00"
SumPayByCard	Сума безготівкової оплати	"0.00"
SumPayByCredit	Сума оплати в кредит	"0.00"
SumPayByCertificate	Сума оплати сертифікатом, картою знижок та ін.	"0.00"
SumPayCheck	Сума по всім видам оплати, з урахуванням заокруглення	"1950.20"
SumPreparePayed	Сума попередньої оплати	"1960.00"

NEW

для якої значення також мають бути представлені у типі (`string`).

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

Заокруглення суми по чеку та решта реалізовані наступним чином.

Сума до заокруглення обчислюється як сума параметрів рядків товарів в чеку, представлені в strJSONData:

$$SumInit = Amount_1 + Amount_2 + \dots$$

Сума після заокруглення обчислюється за певним алгоритмом користувача або відповідно до Постанови НБУ⁴ і передається в payDataJSON в якості параметру SumPayCheck.

Решта по чеку розраховується як:

$$SumRest = SumCash + SumPayByCard + SumPayByCredit + SumPayByCertificate - SumPayCheck$$

Значення заокруглення, яке буде зареєстроване на сервері ФСКО, є різницею:

$$SumDelta = SumPayCheck - SumInit$$

Додатні значення заокруглення відповідають заокругленню “вгору”, від’ємні - заокругленню “вниз”. Наприклад:

SumInit	SumPayCheck	SumDelta
10,67	10,70	0,03
10,67	10,60	- 0,07

У випадку, якщо параметр **SumPayCheck** не визначений (null) або дорівнює нулю, то він замінюється сумою по всім формам оплати. В такому випадку заокруглення буде дорівнювати нулю.

DocType	Тип документа: 0 – чек продажу 4 – ломбардний чек За замовчуванням DocType = 0, якщо параметр не задано.	"0"
---------	---	-----

Специфікація методу:

```
/// <summary>
/// Фіскалізувати чек
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>
/// <param name="strJSONData">
/// Параметри з інформацією про строки чеку (товар з характеристиками,
/// кількість сумма і тд.), у вигляді JSON строки
/// Приклад:
/// {"ReceiptLst":[{"VendorCode":"АртикулТовару1","Name":"НазваТовару1","CdUK
DZED":"","CdDKPP":"","GoodsType":"","Barcode":"123456789","UnitType":"шт","Q
uantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":
"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":fa
lse,"OtherParameters":null},{VendorCode":"АртикулТовару2","Name":"НазваТовар
у2","CdUKDZED":"","CdDKPP":"","GoodsType":"","Barcode":"9999999999","UnitTyp
e":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","Dis
countSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","Is
Excise":false,"OtherParameters":null}}
/// </param>
/// <param name="payDataJSON">
/// Інформація про оплату у вигляді JSON строки
/// Приклад:
/// {"SumCash":"366","SumPayByCard":null,"SumPayByCredit":null,
"SumPayByCertificate":null}
```

⁴ <https://zakon.rada.gov.ua/laws/main/v0025500-18> п.4.

```

/// </param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{"ReceiptFiscalNumber":"203350",
"ReceiptLocalNumber":"1141"}}
/// </returns>
public string FiscalizeCheck( string rroFiscalNumber, string strJSONData,
string payDataJSON );

```

Приклад:

```

LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();
CheckInfo chInfo = new CheckInfo();
VendorCode = "АртикулТовару1";
Name = "НазваТовару1";
CdUKDZED = String.Empty;
Barcode = "123456789";
UnitType = "шт";
Quantity = "1";
Price = "183";
Amount = "181,17";
DiscountPrc= String.Empty;
DiscountSum = "1.83";
IsPriceIncludeVAT = true;
VATRate = "20";
SumVAT = "30.2";
lstGoodsInCheck.ReceiptLst.Add( chInfo );//додати рядок чека з товаром №1

chInfo.VendorCode = "АртикулТовару2";
chInfo.Name = "НазваТовару2";
chInfo.Barcode = "9999999999";
lstGoodsInCheck.ReceiptLst.Add( chInfo );//додати рядок чека з товаром №2

PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "366"; //Сума готівки до сплати після заокруглення
//Конвертація структур до формату JSON рядка
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );
//Параметри функції
Object[] parameters =
{
    edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
    strJSONData, // структура інформації з рядками чека
    payDataJSON // структура інформації про оплату по чеку
};
var ret = CallCashalotMethod( "FiscalizeCheck", parameters );
if ( !ret.Ret )
{
    MessageBox.Show( String.IsNullOrEmpty( ret.ErrorString ) ? "Не вдалося
        фіскалізувати чек" : ret.ErrorString );
}
else if ( ret.Values != null )
{
    string newCheckLocalNum = String.Empty;
    string newCheckFiscalNum = String.Empty;
    //отримання параметру в результаті виконання функції
    var vals = ret.Values.ToArray();
    //перебираємо параметри
    for ( int i = 0; i < vals.Length; i++ )
    {
        // отримуємо значення по ключу
        if ( vals[i].Key.ToLower() == "receiptlocalnumber" )
        {
            newCheckLocalNum = vals[i].Value;
        }
        else
        if ( vals[i].Key.ToLower() == "receiptfiscalnumber" )
        {

```

```

        newCheckFiscalNum = vals[i].Value;
    }
}
MessageBox.Show( $"Чек було успішно фіскалізовано.\n" +
    $"Чеку присвоєно локальний номер: { newCheckLocalNum
},\n" +
    $"та фіскальний номер: { newCheckFiscalNum } "
);
}

```

Приклад JSON рядків:

```

strJSONData=
{"ReceiptLst":
[
{
"VendorCode": "АртикулТовару1",
"Name": "НазваТовару1",
"CdUKDZED": "",
"CdDKPP": "",
"GoodsType": "",
"Barcode": "123456789",
"UnitType": "шт",
"Quantity": "1",
"Price": "20000.03",
"Amount": "19000.03",
"DiscountPrc": null,
"DiscountSum": "1000",
"VATRate": "20",
"IsPriceIncludeVAT": true,
"SumVAT": "3800.01",
"IsExcise": false,
"OtherParameters": null
},
{
"VendorCode": "АртикулТовару2",
"Name": "НазваТовару2",
"CdUKDZED": "",
"CdDKPP": "",
"GoodsType": "",
"Barcode": "123456789",
"UnitType": "шт",
"Quantity": "1",
"Price": "20000.03",
"Amount": "19000.03",
"DiscountPrc": null,
"DiscountSum": "1000",
"VATRate": "20",
"IsPriceIncludeVAT": true,
"SumVAT": "3800.01",
"IsExcise": false,
"OtherParameters": null
}
]
}

payDataJSON=
{
"SumCash": "20000",
"SumPayByCard": "19000",
"SumPayByCredit": null,
"SumPayByCertificate": null,
"SumPayCheck": "38000.10"
}

```

}

Приклад XML-рядка необов'язкових параметрів OtherParams:

```
<?xml version="1.0" encoding="UTF-8"?>
<Table>
  <Record
    VAT ="20"
    IsExcisable="False"
    Nomenclature="СТЕР Пазл 104 Віні Пух"
    UKTZED=""
    VendorCode="82106"
    Barcode=""
    Dimension="шт"
    Discount="1"
    DiscountAmount="0.56"
    PriceIncludeVAT="1"
    AmountVAT="9.24"
  />
</Table>
```

Після виконання запиту Cashalot створює діалогове вікно з графічним відображенням результатів реєстрації продажу у вигляді чеку.


ТОВ "Тест-гідрометцентр"
Тест-гідромет
УКРАЇНА, м. Київ, Адреса
ІД 24234227
ФН ПРРО 4000007619
КАСИР Лабораторний П. П.

Чек №494010

НазваТовару1		
1 X 20000,03		20000,03 0
Знижка	-1000,00	
НазваТовару2		
1 X 20000,03		20000,03
Знижка	-1000,00	

Сума, грн		38000,06
ПДВ 0=20%		3166,67

Заокруглення		0,04
До сплати		38000,10
ГОТІВКА, грн		20000,00
Решта, грн		999,90
КАРТКА		19000,00
Ingenico		
Термінал		51K700J8
ОПЛАТА		
Сума		19000,00 грн.
MASTERCARD		
ЕПЗ:	XXXXXXXXXXXX6883	
Код авторизації:	68014B	
RRN:	048867173620	
ДАТА:9/6/2021	ЧАС:1:07:28 PM	
ЧЕК:	224	
Касир:		
Держатель ЕПЗ: Підпис власника карти не потрібен		
494010		9/6/2021 1:07:31 PM



ФІСКАЛЬНИЙ ЧЕКОНЛАЙН Cashalot

Друк

Надіслати на Email

Готово

6. Визначення фіскального номеру чеку

З метою визначення номеру фіскального чека, зареєстрованого раніше в ФСКО, за локальним номером, зареєстрованим в касі при продажу виконується метод

“**GetReceiptFiscalNumberByLocalNumber**”. В якості параметру передається параметр:

```
{
    Фіскальний номер РРО ,
    Локальний номер чеку
}
```

Специфікація методу:

```

/// <summary>
/// Отримати фискальний номер чека по локальному номеру
/// </summary>
/// <param name="rroFiscalNumber">Фискальний номер РРО</param>
/// <param name="checkLocalNumber">Локальний номер чека</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","
"Values":{"ReceiptFiscalNumber":"203366"}}
/// </returns>
public string GetReceiptFiscalNumberByLocalNumber( string
rroFiscalNumber, string checkLocalNumber );

```

Приклад:

```

Object[] parameters = { edFiscalNumberRRO.Text.Trim(), checkLocalNumber};
var ret = CallCashalotMethod( "GetReceiptFiscalNumberByLocalNumber",
parameters );

```

Визначений фискальний номер чека використовується, зокрема, в якості додаткового параметра при фіскалізації чеку повернення товару.

7. Проведення операції повернення товару

Метод повернення товару схожий на метод фіскалізації чека. Додатково потрібно вказати в якості параметру номер фискального чека товару, який було продано.

Для виконання повернення товару чи товарів з фискального чека з відомим (визначеним) фискальним номером використовується метод **"FiscalizeReturnCheck"**, для якого формується чек повернення аналогічно операції [проведення розрахункової операції](#) із зазначенням тих товарів, які реєструються для повернення: створюються **strJSONData** для списку товарів повернення, **payDataJSON** для повної вартості покупок.

Додатково зазначається параметр:

Параметр	Значення	Приклад	Примітка
strFiscalNumReturnCheck	Фискальний номер чека по якому здійснюється повернення	"275301" "19255.3.7284"	онлайн-чек офлайн-чек

Специфікація методу:


```

/// <summary>
/// Фіскалізувати видатковий чек (повернення)
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="strJSONData">
///Параметри з інформацією про строки чеку(товар з характеристиками,
кількість сумма і тд.), у вигляді JSON строки
///Приклад:
///{"ReceiptLst":[{"VendorCode":"АртикулТовару1","Name":"НазваТовару1","CdUK
DZED":"","CdDKPP":"","GoodsType":"","Barcode":"123456789","UnitType":"шт","Q
uantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":
"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":fa
lse,"OtherParameters":null},{VendorCode":"АртикулТовару2","Name":"НазваТовар
у2","CdUKDZED":"","CdDKPP":"","GoodsType":"","Barcode":"9999999999","UnitTyp
e":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","Dis
countSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","Is
Excise":false,"OtherParameters":null}]
/// </param>
/// <param name="payDataJSON">
/// Інформація про оплату у вигляді JSON строки
/// Приклад:
///
{"SumCash":"366","SumPayByCard":null,"SumPayByCredit":null,"SumPayByCertific
ate":null,"SumPayCheck":"362,34"}
/// </param>
/// <param name="strFiscalNumReturnCheck"> Фіскальний номер чеку по якому
здійснюється повернення</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{"ReceiptFiscalNumber":"203350",
"ReceiptLocalNumber":"1141"}}
/// </returns>
public string FiscalizeReturnCheck( string rroFiscalNumber, string
strJSONData, string payDataJSON, string strFiscalNumReturnCheck)

```

Приклад:

```

// Відправка чека повернення
LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();
CheckInfo chInfo = new CheckInfo();
chInfo.VendorCode = "АртикулТовару1";
chInfo.Name = "НазваТовару1";
chInfo.CdUKDZED = String.Empty;
chInfo.Barcode = "123456789";
chInfo.UnitType = "шт";
chInfo.Quantity = "1";
chInfo.Price = "183";
chInfo.Amount = "181,17";
chInfo.DiscountPrc = String.Empty;
chInfo.DiscountSum = "1.83";
chInfo.IsPriceIncludeVAT = true;
chInfo.VATRate = "20";
chInfo.SumVAT = "30.2";
lstGoodsInCheck.ReceiptLst.Add( chInfo );
chInfo.VendorCode = "АртикулТовару2";
chInfo.Name = "НазваТовару2";
chInfo.Barcode = "9999999999";
lstGoodsInCheck.ReceiptLst.Add( chInfo );
PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "366";
payInfo.SumPayCheck = "362,34";
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );
Object[] parametersChRt =
{
    edFiscalNumberRRO.Text.Trim(),

```

```

        strJSONData,
        payDataJSON,
        edFiscalCheckNum.Text /*фіскальний номер чека, по якому здійснюється
повернення*/
    };
    var retChRt = CallCashalotMethod( "FiscalizeReturnCheck", parametersChRt );
    if ( !retChRt.Ret )
    {
        MessageBox.Show( String.IsNullOrEmpty( retChRt.ErrorString ) ? "Не
вдалося здійснити повернення" : retChRt.ErrorString );
    }
}

```

приклад JSON рядків:

```

strJSONData=
{"ReceiptLst":[{"VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","Barcode":"9999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParameters":null},{"VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","Barcode":"9999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParameters":null}]}

payDataJSON=
{"SumCash":"366","SumPayByCard":null,"SumPayByCredit":null,"SumPayByCertificate":null,"SumPayCheck":"362,34","NumReturnCheck":"180"}

```

Після виконання запиту Cashalot створює діалогове вікно з графічним відображенням результатів реєстрації повернення товару у вигляді чеку.

8. Створення X-Звіту

Для відображення X-Звіту необхідно викликати метод **"ShowXReport"** з параметром:

```

{
    Фіскальний номер PPO
}

```

Специфікація методу:

```

/// <summary>
/// Відобразити X-Звіт (денний звіт без гасіння)
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string ShowXReport( string rroFiscalNumber );

```

Приклад:

```

Object[] parameters = {"11111111111111111111"/*фіскальний номер PPO*/};
var ret = CallCashalotMethod( "ShowXReport", parameters );

```

Результат виконання операції створення X-Звіту відображається на екрані.

ТОВ "Тест-гідрометцентр"	
Тест-гідромет	
УКРАЇНА, м. Київ, Адреса	
ІД 24234227	
ФН ПРРО 4000007619	
X - звіт	
Продаж	
ЗАГАЛЬНА СУМА	38000.10
ГОТІВКА, грн	19000.10
КАРТКА	19000.00
Кількість чеків	1

Обіг О	19000.03
ПДВ О=20%	3166.67
ЗАГАЛЬНА СУМА ПОДАТКУ	3166.67
Повернення	
ЗАГАЛЬНА СУМА	0.00
ГОТІВКА, грн	0.00
КАРТКА	0.00
Кількість чеків	0

ЗАГАЛЬНА СУМА ПОДАТКУ	0.00
Службове внесення	0.00
Службова видача	0.00
06.09.2021 13:09:10	
СЛУЖБОВИЙ ЧЕК	Cashalot
Друк	
Готово	

9. Визначення локального номеру чека по фіскальному номеру

В деяких випадках виникає необхідність визначення локального номеру чека, зареєстрованого в касі, по відомому фіскальному номеру чека. Для цієї операції використовується метод **GetReceiptLocalNumberByFiscalNumber** з параметрами:

```
{
    Фіскальний номер РРО ,
    Фіскальний номер чеку
}
```

Специфікація методу:

```
/// <summary>
/// Отримати фіскальний номер чека по локальному номеру
/// </summary>
/// <param name="roFiscalNumber">Фіскальний номер РРО</param>
/// <param name="receiptFiscalnumber">Фіскальний номер чека</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","
/// "Values":{"ReceiptFiscalNumber":"203366"}}
/// </returns>
public string GetReceiptFiscalNumberByLocalNumber( string
roFiscalNumber, string checkLocalNumber );
```

Приклад:

```

Object[] parameters = { edFiscalNumberRRO.Text.Trim(),
receiptFiscalnumber};
var ret = CallCashalotMethod( "GetReceiptLocalNumberByFiscalNumber",
parameters );

```

10. Отримання XML-структури чеку

Для отримання XML-структури фіскалізованого чеку використовується функція **"GetReceiptXML"** з параметрами

```

{
Фіскальний номер РРО,
Фіскальний номер чеку
}

```

Специфікація методу:

```

/// <summary>
/// Показати чек для друку або відправки на емейл, по фіскальному номеру чека та номеру РРО
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="checkLocalNumber">Фіскальний номер чека</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON з параметром Base64Str1251ReceiptXML - закодована в Base64 строка XML в кодуванні Win-1251
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{"Base64Str1251ReceiptXML":"...."}}
/// </returns>
public string GetReceiptXML(string rroFiscalNumber, string checkFiscalNumber );

```

11. Відображення фіскалізованого чеку

Після фіскалізації чек відображається на екрані. Проте, іноді необхідно відобразити чек повторно. Для цього можна скористатися функцією **"ShowReceipt"** з параметрами

```

{
Фіскальний номер РРО,
Фіскальний номер чеку
}

```

Специфікація методу:

```

/// <summary>
/// Показати чек для друку або відправки на емейл, по фіскальному номеру чека та номеру РРО
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="checkLocalNumber">Фіскальний номер чека</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string ShowReceipt(string rroFiscalNumber, string checkFiscalNumber );

```

12. Закриття зміни та створення Z-Звіту

Для закриття зміни необхідно викликати метод **"CloseShift"** з параметром:

```

{

```

Фіскальний номер PPO

}

Специфікація методу:

```
/// <summary>
/// Закриття зміни
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string CloseShift( string rroFiscalNumber );
```

Приклад:

```
Object[] parameters = {"1111111111111111" /*фіскальний номер PPO*/};
var ret = CallCashalotMethod( "CloseShift", parameters );
```

Перед створенням Z-Звіту та закриттям зміни, описаним в даному розділі, автоматично виконується [метод службового вилучення наявних грошових коштів з каси](#).

Також виконується графічне відображення Z-Звіту для можливості подальшого друку.

ТОВ "Тест-гідрометцентр"	
Тест-гідромет	
УКРАЇНА, м. Київ, Адреса	
ІД 24234227	
ФН ПРРО 4000007619	
Z - звіт	
Продаж	
ЗАГАЛЬНА СУМА	38000.10
ГОТІВКА, грн	19000.10
КАРТКА	19000.00
Кількість чеків	1

Обіг О	19000.03
ПДВ О=20%	3166.67
ЗАГАЛЬНА СУМА ПОДАТКУ	3166.67
Повернення	
ЗАГАЛЬНА СУМА	0.00
ГОТІВКА, грн	0.00
КАРТКА	0.00
Кількість чеків	0

ЗАГАЛЬНА СУМА ПОДАТКУ	0.00
Службове внесення	0.00
Службова видача	19000.10
РЕГІСТРИ ДЕННИХ ПІДСУМКІВ ОБНУЛЕНІ	
ФІСКАЛЬНИЙ ЗВІТНИЙ ЧЕК ДІЙСНИЙ	
494012 06.09.2021 13:10:47	
ФІСКАЛЬНИЙ ЧЕК	Cashalot
Друк	
Готово	

NEW

4. Ломбардні операції

Для фіскалізації ломбардних послуг введено системні номенклатури, які необхідно використовувати при виклику методу фіскалізації чеку `Fiscalizecheck()`. Повернення товару при ломбардних операціях не підтримується.

Також необхідно вказати тип документу `DocType = 4` в JSON-структурі параметрів чеку.

Системні номенклатури для ломбардних операцій.

Артикул	Назва	Тип послуги	Напрямок руху*
"ЛО-00001"	"Повернення страхування"	"3"	-
"ЛО-00002"	"Надання кредиту"	"3"	-
"ЛО-00003"	"Добір кредиту"	"3"	-
"ЛО-00004"	"Надання додаткового кредиту"	"3"	-
"ЛО-00005"	"Зберігання"	"2"	+
"ЛО-00006"	"Страховий платіж"	"2"	+
"ЛО-00007"	"Часткове повернення кредиту (Частковий викуп)"	"2"	+
"ЛО-00008"	"Повне повернення кредиту (Повний викуп)"	"2"	+
"ЛО-00009"	"Повернення кредиту"	"2"	+
"ЛО-00010"	"Користування кредитом"	"2"	+
"ЛО-00011"	"Сплата відсотків по кредиту"	"2"	+
"ЛО-00012"	"Сплата пені"	"2"	+

* "-" видача грошей з каси; "+" повернення грошей в касу.

Напрямок руху необхідно враховувати при обчисленні кінцевого підсумку в чеку: від позицій типу 2 віднімаються позиції типу 3, результат вказується в **Paydata** з врахуванням знаку.

Приклад даних для ломбардної операції

```
Itmdata = {"ReceiptLst":[{"VendorCode":"ЛО-00001","Name":"Повернення
страхування","DocNumber":"1/a12 від
15.08.21","CdUKDZED":null,"CdDKPP":null,"GoodsType":"3","Barcode":null,"UnitType":"шт",
"Quantity":"1","Price":"100","Amount":"100","DiscountPrc":null,"DiscountSum":null,
"IsPriceIncludeVAT":false,"VATRate":null,"SumVAT":null,"IsExcise":false,"OtherParameters":null}],
"Comment":"ТЕСТОВА ЛОМБАРДНА ОПЕРАЦІЯ","DocType":"4"}];
Paydata =
{"SumCash":"100","SumPayByCard":null,"SumPayByCredit":null,"SumPayByCertificate":null,
"SumPayCheck":"100"};
```

5. Приклад використання API для ломбардних операцій

```
LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();
CheckInfo chInfo = new CheckInfo();
```

```

chInfo.VendorCode = "ЛО-00002";
chInfo.Name = "Надання кредиту";
chInfo.Quantity = "1";//кількість
chInfo.Price ="200";//ціна(обов'язково)
chInfo.Amount = "200";//сумма((кількість * ціну) - дисконт)
chInfo.GoodsType = "3";// мінусова операція

//додати строку чеку з товаром №1
lstGoodsInCheck.ReceiptLst.Add( chInfo );

CheckInfo chInfo2 = new CheckInfo( chInfo );
chInfo2.VendorCode = "ЛО-00009";
chInfo2.Name = "Повернення кредиту";
chInfo2.Quantity ="1"; //кількість
chInfo2.Price = "100";//ціна(обов'язково)
chInfo2.Amount = "100";//сумма((кількість * ціну) - дисконт)
chInfo2.GoodsType = "2"; // плюсова операція

//додати строку чеку з товаром №2
lstGoodsInCheck.ReceiptLst.Add( chInfo2 );
lstGoodsInCheck.DocType = 4; // Вказуємо що тип чека ломбардний!!!

PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "-100"; //Сума готівки наданої покупцем(мінусова так як до видачі
100грн )

payInfo.SumPayCheck = "-100";//Сума оплати загальна(мінусова так як до видачі
100грн)

//Конвертація структур до формату джейсон строки
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );

//Параметри функції
Object[] parameters =
{
    edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
    strJSONData, // структура інформації зі строками чека
    payDataJSON // структура інформації про оплату по чеку
};

var ret = CallCashaLotMethod( "FiscalizeCheck", parameters );
if ( ret?.Ret != true )
{
    MessageBox.Show( String.IsNullOrEmpty( ret?.ErrorString ) ? "Не вдалося
фіскалізувати чек" : ret?.ErrorString );
}
else if ( ret?.Values != null )
{
    //отримання параметру в результаті виконання функції
    var vals = ret.Values.ToArray();
    //перебираємо параметри
    for ( int i = 0; i < vals.Length; i++ )
    {
        // отримуємо значення по ключу

```

```

        if ( vals[i].Key.ToLower() == "receiptlocalnumber" )
        {
            edLocalCheckNum.Text = vals[i].Value;
        }
        else
        if ( vals[i].Key.ToLower() == "receiptfiscalnumber" )
        {
            edFiscalCheckNum.Text = vals[i].Value;
        }
    }
    MessageBox.Show( $"Чек було успішно фіскалізовано.\n" +
        $"Чеку присвоєно локальний номер: {edLocalCheckNum.Text},\n" +
        $"та фіскальний номер: {edFiscalCheckNum.Text} "
        );
}

```

6. Приклад реалізації функції для виклику API методів ПРРО Cashalot

//Структура інформації строки чеку(містить всі дані для створення товару в CashaLot)

```

public class CheckInfo
{
    public CheckInfo()
    { }
    public CheckInfo( CheckInfo ckInfo )
    {
        this.VendorCode = ckInfo.VendorCode;
        this.Name = ckInfo.Name;
        this.CdUKDZED = ckInfo.CdUKDZED;
        this.CdDKPP = ckInfo.CdDKPP;
        this.GoodsType = ckInfo.GoodsType;
        this.Barcode = ckInfo.Barcode;
        this.UnitType = ckInfo.UnitType;
        this.Quantity = ckInfo.Quantity;
        this.Price = ckInfo.Price;
        this.Amount = ckInfo.Amount;
        this.DiscountPrc = ckInfo.DiscountPrc;
        this.DiscountSum = ckInfo.DiscountSum;
        this.VATRate = ckInfo.VATRate;
        this.IsPriceIncludeVAT = ckInfo.IsPriceIncludeVAT;
        this.SumVAT = ckInfo.SumVAT;
        this.IsExcise = ckInfo.IsExcise;
        this.OtherParametrs = ckInfo.OtherParametrs;
        this.VATLetter = ckInfo.VATLetter;
        this.ExciseLetter = ckInfo.ExciseLetter;
        this.ExciseStampBarcode = ckInfo.ExciseStampBarcode;
    }

    //Артикул товару
    public String VendorCode { get; set; } = String.Empty;
    public String Article { get => VendorCode; set { VendorCode = value; } }

    //Назва товару
    public String Name { get; set; } = String.Empty;
    //Код УКДЗЕД

```



```

public String CdUKDZED { get; set; } = String.Empty;
//Код ДКПП(для послуг)
public String CddKPP { get; set; } = String.Empty;
//Тип (1-товар/2-послуга(+)|3-послуга ломбардна(-)). За замовчуванням товар,
якщо нічого не задано.
public String GoodsType { get; set; } = String.Empty;
//Штрихкод
public String Barcode { get; set; } = String.Empty;
//Код одиниці виміру
public String UnitType { get; set; } = String.Empty;
//Кількість товару
public String Quantity { get; set; } = String.Empty;
//Ціна товару
public String Price { get; set; } = String.Empty;
//Сума
public String Amount { get; set; } = String.Empty;
//Процент знижки
public String DiscountPrc { get; set; } = String.Empty;
//Сума знижки
public String DiscountSum { get; set; } = String.Empty;
//Процент податку ПДВ
public String VATRate { get; set; } = String.Empty;
//Ознака - чи включає сума ПДВ
public bool? IsPriceIncludeVAT { get; set; } = true;
//Сума ПДВ
public String SumVAT { get; set; } = String.Empty;
//Ознака - підакцизного товару
public bool? IsExcise { get; set; } = false;
//Інші параметри у вигляді строки XML
//Приклад: <?xml version="1.0" encoding="UTF-8"?><Table><Record VAT
="20" IsExcisable="False" Nomenclature="СТЕП Пазл 104 Винни Пух" UKTZED=""
VendorCode="82106" Barcode="" Dimension="шт" Discount="1"
DiscountAmount="0.56" PriceIncludeVAT="1" AmountVAT="9.24"/></Table>
public String OtherParametrs { get; set; }
//Літера ставки податку ПДВ
public String VATLetter { get; set; } = String.Empty;
//Літера ставки податку Акцизу
public String ExciseLetter { get; set; } = String.Empty;
//Штрихкод акцизної марки
public String ExciseStampBarcode { get; set; } = String.Empty;
};

//Структура з масивом срок чеку
public class LstCheckInfoRsp
{
public List<CheckInfo> ReceiptLst { get; set; }
//Коментар
public String Comment { get; set; }
//Тип чека(0 - продаж, 4 - ломбард)
public long DocType { get; set; } = 0;
};

//шлях до каталогу ПРРО Cashalot

```

```

public string edCashalotPath = "C:\\ProgramData\\Cashalot\\Cashalot";

//Фіскальний номер РРО
public string edFiscalNumberRRO = "11111111111111";

//завантажена бібліотека
private Assembly asmCashalotDll;

//тип класу Апі
private Type asmCashalotClass;

//екземпляр класу Апі
private object instCashalotApi;

//клас відповіді апі для десеріалізації
public class CashalotApiRsp
{
    //Ознака успішності виконання функції (true - успішно/false - не успішно)
    public bool Ret { get; set; } = false;
    //Рядок з текстом помилки
    public string ErrorString { get; set; } = String.Empty;
    //Масив параметрів, які можуть повернути функції АПІ
    public Dictionary<string, string> Values { get; set; }
}

/// <summary>
/// Функція для виклику метода з АПІ Cashalot
/// </summary>
/// <param name="methodName"> Назва методу </param>
/// <param name="parameters"> Список параметрів методу</param>
/// <returns></returns>
public CashalotApiRsp CallCashalotMethod(string methodName, Object[] parameters )
{
    CashalotApiRsp ret = new CashalotApiRsp();

    if ( edCashalotPath.Trim().Length == 0 )
        MessageBox.Show( "Не вказано шлях до каталогу Cashalot" );

    if ( edFiscalNumberRRO.Trim().Length == 0 )
        MessageBox.Show( "Не вказано фіскальний номер РРО" );

    string strPath = $"{edCashalotPath}\\CashalotApiBridge.dll";
    if ( !File.Exists( strPath ) )
    {
        MessageBox.Show( "За вказаним каталогом не знайдено CashalotApiBridge.dll"
    );
        return ret;
    }

    if ( asmCashalotDll == null )
        asmCashalotDll = System.Reflection.Assembly.LoadFrom( strPath
); //Завантаження бібліотеки

    if ( asmCashalotDll != null )
    {

```

```

        if( asmCashalotClass == null)
            asmCashalotClass = asmCashalotDll.GetType(
"CashalotApiBridge.BridgeApi" );

        if ( asmCashalotClass != null )
        {
            if( instCashalotApi == null)
                instCashalotApi = Activator.CreateInstance( asmCashalotClass
);//Створення об'єкту класу

            System.Reflection.MethodInfo mi =
asmCashalotClass.GetMethod(methodName );//отримання інформації по вказанному методу
            if ( mi != null )
            {
                var retVal = mi.Invoke( instCashalotApi, parameters ); // виклик
методу з передачею параметрів
                if( retVal !=null )
                {
                    try
                    {
                        //десеріалізація отриманої відповіді
                        ret = JsonConvert.DeserializeObject<CashalotApiRsp>(
retVal as String );
                    }
                    catch
                    {
                        MessageBox.Show( "Не вдалося десеріалізувати відповідь від
Cashalot" );
                    }
                }
            }
        }
    }
    return ret;
}

```

Приклад проведення чеку “ПЕРЕДОПЛАТИ”

NEW

```

private void buttonPrepayment_Click( object sender, EventArgs e )
{
    LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
    lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();

    //Заповнюємо дві позиції товарів в чек
    CheckInfo chInfo = new CheckInfo();// структура з параметрами першої
позиції товару
    chInfo.Name = "ТоварПредоплатыПослеплаты1";
    chInfo.UnitType = "шт";
    chInfo.Quantity = "2";
    chInfo.Price = "1000";
    chInfo.Amount = "2000";
    //додати строку чеку з товаром №1
    lstGoodsInCheck.ReceiptLst.Add( chInfo );
}

```

```

        CheckInfo chInfo2 = new CheckInfo( chInfo );//Структура з параметрами
другої позиції товару
        chInfo2.Name = "ТоварПредоплатыПослеплаты2";
        chInfo.Quantity = "1";
        chInfo.Price = "1000";
        chInfo.Amount = "1000";
//додати строку чеку з товаром №2
        lstGoodsInCheck.ReceiptLst.Add( chInfo2 );

//присвоїти номер договору/замовлення передоплати(друкується у чеку)
        lstGoodsInCheck.SaleOrderNumber = "555444333";

//Заповнюємо інформацію про оплату
        PayInfoRsp payInfo = new PayInfoRsp();
        payInfo.SumCash = "1500"; //Сума готівки наданої покупцем( може бути
більшою за суму до сплати, з неї вираховується решта)

        //встановлюємо порядок оплати "ПЕРЕДОПЛАТА"
        payInfo.PaymentOrderType = 1; //Спосіб оплати (0-звичайна | 1 -
передоплата | 2 - післяплата | 3 - інтернет продаж )
        //вказуємо суму передоплати
        payInfo.SumPreparePayed = "1500";

        //вказуємо загальну суму по чеку
        payInfo.SumPayCheck = "1500";//Сума оплати загальна з урахуванням
заокруглення

        //Конвертація структур до формату джейсон строки
        string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
        string payDataJSON = JsonConvert.SerializeObject( payInfo );

        //Параметри функції
        Object[] parameters =
        {
            edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
            strJSONData, // структура інформації зі строками чека
            payDataJSON // структура інформації про оплату по чеку
        };

        var ret = CallCashaLotMethod( "FiscalizeCheck", parameters );
        if ( ret?.Ret != true )
        {
            MessageBox.Show( String.IsNullOrEmpty( ret?.ErrorString ) ? "Не
вдалося фіскалізувати чек" : ret?.ErrorString );
        }
        else if ( ret?.Values != null )
        {
            //отримання параметру в результаті виконання функції
            var vals = ret.Values.ToArray();
            //перебираємо параметри
            for ( int i = 0; i < vals.Length; i++ )
            {
                // отримуємо значення по ключу
                if ( vals[i].Key.ToLower() == "receiptlocalnumber" )
                {

```

```

        edLocalCheckNum.Text = vals[i].Value;
    }
    else
    if ( vals[i].Key.ToLower() == "receiptfiscalnumber" )
    {
        //Отримуємо фіскальний номер створеного чека ПЕРЕДОПЛАТИ,
та зберігаємо його для можливості здійснити "післяплату"
        edFNRcpPrepayment.Text = vals[i].Value;
    }
    }
    MessageBox.Show( $"Чек було успішно фіскалізовано.\n" +
        $"Чеку присвоєно локальний номер:
{edLocalCheckNum.Text},\n" +
        $"та фіскальний номер: {edFiscalCheckNum.Text}
"
        );
    }
}

```

Приклад проведення чеку "ПІСЛЯПЛАТИ"

```

private void buttonAfterpayment_Click( object sender, EventArgs e )
{
    LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
    lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();

    //присвоїти номер договору/замовлення передоплати
    lstGoodsInCheck.SaleOrderNumber = "555444333";

    PayInfoRsp payInfo = new PayInfoRsp();
    payInfo.SumCash = "1600"; //Сума готівки наданої покупцем( може бути
більшою за сумму до сплати, з неї вираховується решта/здача)

    //встановлюємо спосіб оплати "ПІСЛЯПЛАТА"
    payInfo.PaymentOrderType = 2; //Спосіб оплати (0-звичайна | 1 -
передоплата | 2 - післяплата | 3 - інтернет продаж )

    //Вказуємо фіскальний номер чека "ПЕРЕДОПЛАТИ", на основі якого буде
післяплата(поля чека будуть автоматично зачитані з нього, передані користувачем
будуть проігноровані)
    payInfo.ParentReceiptFiscalNumber = edFNRcpPrepayment.Text;
    //вказуємо суму передоплати
    payInfo.SumPreparePayed = "1500";
    //вказуємо загальну суму по чеку
    payInfo.SumPayCheck = "1500";//Сума оплати загальна з урахуванням
заокруглення

    //Конвертація структур до формату джейсон строки
    string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
    string payDataJSON = JsonConvert.SerializeObject( payInfo );

    //Параметри функції
    Object[] parameters =
    {
        edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
        strJSONData, // структура інформації зі строками чека
    }
}

```

```

        payDataJSON // структура інформації про оплату по чеку
    };

    var ret = CallCashaLotMethod( "FiscalizeCheck", parameters );
    if ( ret?.Ret != true )
    {
        MessageBox.Show( String.IsNullOrEmpty( ret?.ErrorString ) ? "Не
вдалося фіскалізувати чек" : ret?.ErrorString );
    }
    else if ( ret?.Values != null )
    {
        //отримання параметру в результаті виконання функції
        var vals = ret.Values.ToArray();
        //перебираємо параметри
        for ( int i = 0; i < vals.Length; i++ )
        {
            // отримуємо значення по ключу
            if ( vals[i].Key.ToLower() == "receiptlocalnumber" )
            {
                edLocalCheckNum.Text = vals[i].Value;
            }
            else
            if ( vals[i].Key.ToLower() == "receiptfiscalnumber" )
            {
                edFiscalCheckNum.Text = vals[i].Value;
            }
        }
        MessageBox.Show( $"Чек було успішно фіскалізовано.\n" +
            $"Чеку присвоєно локальний номер:
{edLocalCheckNum.Text},\n" +
            $"та фіскальний номер: {edFiscalCheckNum.Text}
"
            );
    }
}

```

Приклад проведення Чеку "Інтернет-продаж"

```

private void btnInternetPayment_Click(object sender, EventArgs e)
{
    LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
    lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();

    //Заповнюємо дві позиції товарів в чек
    CheckInfo chInfo = new CheckInfo(); // структура з параметрами першої
позиції товару
    chInfo.Name = "ТоварІнтернетПродажу1";
    chInfo.UnitType = "шт";
    chInfo.Quantity = "2";
    chInfo.Price = "1000";
    chInfo.VATRate = "20";
    chInfo.Amount = "2000";

    //додати строку чеку з товаром №1
    lstGoodsInCheck.ReceiptLst.Add(chInfo);
}

```

```

        CheckInfo chInfo2 = new CheckInfo(chInfo); // структура з параметрами
другої позиції товару
        chInfo2.Name = "ТоварІнтернетПродажу2";
        chInfo.Quantity = "1";
        chInfo.Price = "1000";
        chInfo.Amount = "1000";

        //додати строку чеку з товаром №2
        lstGoodsInCheck.ReceiptLst.Add(chInfo2);

        //присвоїти номер договору/замовлення (друкується в чеці)
        lstGoodsInCheck.SaleOrderNumber = "555444333555";

        PayInfoRsp payInfo = new PayInfoRsp();
        //вказуємо порядок оплати "Інтернет-продаж"
        payInfo.PaymentOrderType = 3; //Спосіб оплати (0-звичайна | 1 -
передоплата | 2 - післяплата | 3 - інтернет-продаж )

        payInfo.SumPayByCard = "1500"; //Сума оплати картою або на рахунок
        //сума передоплати, якщо вона менша за суму всіх позицій чека то
решта додається в рядок ПІСЛЯПЛАТА
        payInfo.SumPreparePayed = "1500";

        payInfo.SumPayCheck = "3000"; //Сума оплати по чеку загальна

        //Конвертація структур до формату джейсон строки
        string strJSONData = JsonConvert.SerializeObject(lstGoodsInCheck);
        string payDataJSON = JsonConvert.SerializeObject(payInfo);

        //Параметри функції
        Object[] parameters =
        {
            edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
            strJSONData, // структура інформації зі строками чека
            payDataJSON // структура інформації про оплату по чеку
        };

        var ret = CallCashaLotMethod("FiscalizeCheck", parameters);
        if (ret?.Ret != true)
        {
            MessageBox.Show(String.IsNullOrEmpty(ret?.ErrorString) ? "Не
вдалося фіскалізувати чек" : ret?.ErrorString);
        }
        else if (ret?.Values != null)
        {
            //отримання параметру в результаті виконання функції
            var vals = ret.Values.ToArray();
            //перебираємо параметри
            for (int i = 0; i < vals.Length; i++)
            {
                // отримуємо значення по ключу
                if (vals[i].Key.ToLower() == "receiptlocalnumber")
                {
                    edLocalCheckNum.Text = vals[i].Value;
                }
            }
        }
    }
}

```

```

        else
            if (vals[i].Key.ToLower() == "receiptfiscalnumber")
            {
                edFNRcpPrepayment.Text = vals[i].Value;
            }
        }
        MessageBox.Show($"Чек було успішно фіскалізовано.\n" +
            $"Чеку присвоєно локальний номер:
{edLocalCheckNum.Text},\n" +
            $"та фіскальний номер: {edFiscalCheckNum.Text}
"
            );
    }
}

```

Приклад формування JSON “ПрикладЧекПередплати”

Виконання функції *FiscalizeCheck/FiscalizeReturnCheck*: “ПрикладЧекПередплати”

JSONGoodsLstData=

```

{"ReceiptLst": [{"VendorCode": "", "Article": "", "Name": "ТоварПредоплатыПо
слеплаты1", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "1", "Barcode": "", "
UnitType": "шт", "Quantity": "1", "Price": "1000", "Amount": "1000", "Disco
untPrc": "", "DiscountSum": "", "VATRate": "", "IsPriceIncludeVAT": true, "Su
mVAT": "", "IsExcise": false, "OtherParams": null, "VATLetter": "", "ExciseL
etter": "", "ExciseStampBarcode": "", "DocNumber": ""}, {"VendorCode": "", "
Article": "", "Name": "ТоварПредоплатыПослеплаты2", "CdUKDZED": "", "CdDKPP
": "", "GoodsType": "1", "Barcode": "", "UnitType": "шт", "Quantity": "2",
"Price": "1000", "Amount": "2000", "DiscountPrc": "", "DiscountSum": "", "
VATRate": "", "IsPriceIncludeVAT": true, "SumVAT": "", "IsExcise": false, "OtherP
arameters": null, "VATLetter": "", "ExciseLetter": "", "ExciseStampBarcode": ""
", "DocNumber": ""}], "Comment": null, "DocType": "0", "DocSubType": "0", "Sa
leOrderNumber": "555444333"}

```

JSONPayData =

```

{"SumCash": "1500", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCerti
ficate": null, "SumPayCheck": "1500", "CustomerEmail": null, "TerminalID": null,
"ApprovalCode": null, "RRN": null, "IssuerName": null, "PAN": null, "TransactionDat
e": null, "SignVerif": null, "AcquireName": null, "InvoiceNumber": null, "ParentRRN
": null, "PaymentOrderType": "1", "SumPreparePayed": "1500", "ParentReceiptFis
calNumber": null, "OtherParams": null}

```

Приклад формування JSON “ПрикладрЧекПісляплати”

Виконання функції *FiscalizeCheck/FiscalizeReturnCheck*: “ПрикладрЧекПісляплати”

JSONGoodsLstData =

```
{ "ReceiptLst": [], "Comment": null, "DocType": "0", "DocSubType": "0", "SaleOrderNumber": "555444333" }
```

JSONPayData =

```
{ "SumCash": "1600", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "1500", "CustomerEmail": null, "TerminalID": null, "ApprovalCode": null, "RRN": null, "IssuerName": null, "PAN": null, "TransactionDate": null, "SignVerif": null, "AcquireName": null, "InvoiceNumber": null, "ParentRRN": null, "PaymentOrderType": 2, "SumPreparePayed": "1500", "ParentReceiptFiscalNumber": "604779", "OtherParameters": null }
```

Важливо: обов'язковим параметром є **ParentReceiptFiscalNumber**

Приклад формування JSON “ПрикладРЧекІнтернетПродажу”

Виконання функції *“FiscalizeCheck/FiscalizeReturnCheck.”*

Чек інтернет-продажу:

JSONGoodsLstData =

```
{ "ReceiptLst": [ { "VendorCode": "", "Article": "", "Name": "ТоварІнтернетПродажу1", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "1", "Barcode": "", "UnitType": "шт", "Quantity": "1", "Price": "1000", "Amount": "1000", "DiscountPrc": "", "DiscountSum": "", "VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "", "IsExcise": false, "OtherParameters": null, "VATLetter": "", "ExciseLetter": "", "ExciseStampBarcode": "", "DocNumber": "" }, { "VendorCode": "", "Article": "", "Name": "ТоварІнтернетПродажу2", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "1", "Barcode": "", "UnitType": "шт", "Quantity": "2", "Price": "1000", "Amount": "2000", "DiscountPrc": "", "DiscountSum": "", "VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "", "IsExcise": false, "OtherParameters": null, "VATLetter": "", "ExciseLetter": "", "ExciseStampBarcode": "", "DocNumber": "" } ], "Comment": null, "DocType": "0", "DocSubType": "0", "SaleOrderNumber": "555444333555" }
```

JSONPayData =

```
{ "SumCash": null, "SumPayByCard": "1500", "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "3000", "CustomerEmail": null, "TerminalID": null, "ApprovalCode": null, "RRN": null, "IssuerName": null, "PAN": null, "TransactionDate": null, "SignVerif": null, "AcquireName": null, "InvoiceNumber": null, "ParentRRN": null, "PaymentOrderType": "3", "SumPreparePayed": "1500", "ParentReceiptFiscalNumber": null, "OtherParameters": null }
```

Важливо: Сума післяплати розраховується автоматично від різниці всіх позицій чеку, мінус сума передоплати.

Результат формування XML “ЧекПередплати”

```
<?xml version="1.0" encoding="windows-1251"?>
```

```

<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>0</DOCSUBTYPE>
    <UID>5CE7FE06-2BCB-4B3F-9ECC-6983EE225009</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>15122021</ORDERDATE>
    <ORDERTIME>142131</ORDERTIME>
    <ORDERNUM>2887</ORDERNUM>
    <CASHDESKNUM>1</CASHDESKNUM>
    <CASHREGISTERNUM>4000007619</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <!--CASHALOTCOMMENT>
  <PAYMENTSTATUS>4</PAYMENTSTATUS>
  <PAYMENTORDER>1</PAYMENTORDER>
  <PREPAYORDERNUM>555444333</PREPAYORDERNUM>
</CASHALOTCOMMENT-->
</CHECKHEAD>
<CHECKTOTAL>
  <SUM>1500.00</SUM>
  <DISCOUNTSUM>1500.00</DISCOUNTSUM>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТІВКА</PAYFORMNM>
    <SUM>1500.00</SUM>
  </ROW>
</CHECKPAY>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>4611686020234877719</CODE>
    <NAME>ТоварПредоплатиПослеплати1</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>1000.00</PRICE>
    <COST>1000.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>500.00</DISCOUNTSUM>
    <COMMENT>Передоплата 500,0 грн.</COMMENT>
    <!--CASHALOTDESCRIPTION>
  <LEFTTOPAY>500.00</LEFTTOPAY>
</CASHALOTDESCRIPTION-->
  </ROW>
  <ROW ROWNUM="2">
    <CODE>4611686020660047365</CODE>
    <NAME>ТоварПредоплатиПослеплати2</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>2.000</AMOUNT>
    <PRICE>1000.00</PRICE>
    <COST>2000.00</COST>

```

```

<DISCOUNTTYPE>0</DISCOUNTTYPE>
<DISCOUNTSUM>1000.00</DISCOUNTSUM>
<COMMENT>Передплата 1000,0 грн.</COMMENT>
<!--<CASHALOTDESCRIPTION>
<LEFTTOPAY>1000.00</LEFTTOPAY>
</CASHALOTDESCRIPTION-->
</ROW>
</CHECKBODY>
</CHECK>

```

ТОВ "Тест-гідрометцентр"
 Тест-гідромет
 УКРАЇНА, м. Київ, Адреса
 ІД 24234227
 ФН ПРРО 4000007619
 КАСИР Лабораторний П. П.

Чек №604634

Номер замовлення 555444333

ПЕРЕДОПЛАТА

Товар	Предоплати	Послеплати	1
1 X	1000,00		1000,00
Товар	Предоплати	Послеплати	2
2 X	1000,00		2000,00

Сума по чеку			3000,00 грн
Передплата			1500,00 грн
Післяплата			1500,00 грн

СУМА			1500,00 ГРН

ГОТІВКА			1500,00 ГРН
604634			15.12.2021 14:21:31



ФІСКАЛЬНИЙ ЧЕК

ОНЛАЙН
Cashalot

Чек післяплати:

```

<?xml version="1.0" encoding="windows-1251"?>
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>0</DOCSUBTYPE>
    <UID>68E640A2-A29F-4D55-9E87-9BC3C4580B52</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>15122021</ORDERDATE>
    <ORDERTIME>142223</ORDERTIME>
    <ORDERNUM>2888</ORDERNUM>

```

```

<CASHDESKNUM>1</CASHDESKNUM>
<CASHREGISTERNUM>4000007619</CASHREGISTERNUM>
<CASHIER>Лабораторний П. П.</CASHIER>
<VER>1</VER>
<!--<CASHALOTCOMMENT>
<PAYMENTSTATUS>1</PAYMENTSTATUS>
<PAYMENTORDER>2</PAYMENTORDER>
<PREPAYORDERNUM>555444333</PREPAYORDERNUM>
<PARENTID>5ce7fe06-2bcb-4b3f-9ecc-6983ee225009</PARENTID>
<PARENTFISCALNUMBER>604779</PARENTFISCALNUMBER>
</CASHALOTCOMMENT>-->
</CHECKHEAD>
<CHECKTOTAL>
  <SUM>1500.00</SUM>
  <DISCOUNTSUM>1500.00</DISCOUNTSUM>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТІВКА</PAYFORMNM>
    <SUM>1500.00</SUM>
    <PROVIDED>1600.00</PROVIDED>
    <REMAINS>100.00</REMAINS>
  </ROW>
</CHECKPAY>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>4611686020234877719</CODE>
    <NAME>ТоварПредоплатыПослеплаты1</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>1000.00</PRICE>
    <COST>1000.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>500.00</DISCOUNTSUM>
    <COMMENT>Післяплата 500,0 грн.</COMMENT>
    <!--<CASHALOTDESCRIPTION>
  <ALREADYPAYED>500.00</ALREADYPAYED>
</CASHALOTDESCRIPTION>-->
  </ROW>
  <ROW ROWNUM="2">
    <CODE>4611686020660047365</CODE>
    <NAME>ТоварПредоплатыПослеплаты2</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>2.000</AMOUNT>
    <PRICE>1000.00</PRICE>
    <COST>2000.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>1000.00</DISCOUNTSUM>
    <COMMENT>Післяплата 1000,0 грн.</COMMENT>
    <!--<CASHALOTDESCRIPTION>
  <ALREADYPAYED>1000.00</ALREADYPAYED>
</CASHALOTDESCRIPTION>-->
  </ROW>
</CHECKBODY>
</CHECK>

```

ТОВ "Тест-гідрометцентр"
Тест-гідромет
УКРАЇНА, м. Київ, Адреса
ІД 24234227
ФН ПРРО 4000007619
КАСІР Лабораторний П. П.

Чек №604635

Номер замовлення 555444333

ПІСЛЯПЛАТА

Товар	Предоплати	Послеплати	1
1 X	1000,00		1000,00
Товар	Предоплати	Послеплати	2
2 X	1000,00		2000,00

Сума по чеку			3000,00 грн
Передоплата			1500,00 грн
Післяплата			1500,00 грн

СУМА			1500,00 ГРН

ГОТІВКА			1600,00 ГРН
Решта			100,00 грн
604635			15.12.2021 14:22:23



ФІСКАЛЬНИЙ ЧЕК

ОНЛАЙН
Cashalot

Чек інтернет-продажу:

```
<?xml version="1.0" encoding="windows-1251"?>
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>0</DOCSUBTYPE>
    <UID>F3EAA733-74C0-40E6-93A8-FDCF7E136F03</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>15122021</ORDERDATE>
    <ORDERTIME>142258</ORDERTIME>
    <ORDERNUM>2889</ORDERNUM>
    <CASHDESKNUM>1</CASHDESKNUM>
    <CASHREGISTERNUM>4000007619</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <!--<CASHALOTCOMMENT>
  <PAYMENTSTATUS>5</PAYMENTSTATUS>
  <PAYMENTORDER>3</PAYMENTORDER>
  <PREPAYORDERNUM>555444333555</PREPAYORDERNUM>
```

```
</CASHALOTCOMMENT>-->
</CHECKHEAD>
<CHECKTOTAL>
  <SUM>3000.00</SUM>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>201</PAYFORMCD>
    <PAYFORMNM>ПІСЛЯПЛАТА</PAYFORMNM>
    <SUM>1500.00</SUM>
  </ROW>
  <ROW ROWNUM="2">
    <PAYFORMCD>2</PAYFORMCD>
    <PAYFORMNM>ПЕРЕДОПЛАТА</PAYFORMNM>
    <SUM>1500.00</SUM>
  </ROW>
</CHECKPAY>
<CHECKTAX>
  <ROW ROWNUM="1">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>0</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>3000.00</TURNOVER>
    <SOURCESUM>3000.00</SOURCESUM>
    <SUM>500.00</SUM>
  </ROW>
</CHECKTAX>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>4611686020891480948</CODE>
    <NAME>Товар ІнтернетПродажу1</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>1000.00</PRICE>
    <LETTERS>0</LETTERS>
    <COST>1000.00</COST>
  </ROW>
  <ROW ROWNUM="2">
    <CODE>4611686019445393394</CODE>
    <NAME>Товар ІнтернетПродажу2</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>2.000</AMOUNT>
    <PRICE>1000.00</PRICE>
    <LETTERS>0</LETTERS>
    <COST>2000.00</COST>
  </ROW>
</CHECKBODY>
</CHECK>
```

ТОВ "Тест-гідрометцентр"
Тест-гідромет
УКРАЇНА, м. Київ, Адреса
ІД 24234227
ФН ПРРО 4000007619
КАСІР Лабораторний П. П.

Чек №604636

Номер інтернет-замовлення 555444333555

ТоварІнтернетПродажу1	
1 X 1000,00	1000,00 О
ТоварІнтернетПродажу2	
2 X 1000,00	2000,00 О
-----	-----
СУМА	3000,00 ГРН
ПДВ О=20%	500,00
-----	-----
ПЕРЕДОПЛАТА	1500,00 ГРН
ПІСЛЯПЛАТА	1500,00 ГРН
604636	15.12.2021 14:22:58



ФІСКАЛЬНИЙ ЧЕК

ОНЛАЙН
Cashalot