

Програмне забезпечення OLE сервер роботи з терміналом збору даних «PidkovaShopAssistance»

1. Для роботи з терміналом збору даних (далі ТЗД) потрібно створити об'єкт OLE сервера «ProzaSoftMobileAccounting.InvoiceAudit».

Приклад

```
sole:=CreateOleObject('PidkovaSA.Terminaldata');
```

2. Обробка складається з двох режимів:

- формування вихідних даних;
- обмін даними з ТЗД;
- завантаження результатів роботи ТЗД;

Вступ

Бібліотека призначена обміну даними з мобільним додатком «Pidkova Shop Assistance» Форма файлу бази даних – sqllite.

Для того щоб термінал збору даних разом із додатком «Pidkova Shop Assistance» міг здійснювати операції необхідно підготувати дані і переписати файл на пристрій. По завершенні роботи з цього файлу слід імпортувати створені документи.

Бібліотека дозволяє підготувати довідникові дані для проведення ревізії і оформлення приходу товару. Для цього використовуються методи

- InsertDepot – список підрозділів
- InsertSupplier список постачальників
- Insertitem список номенклатури
- InsertBarcode список штрих кодів, цін і кількості

Також бібліотека дозволяє імпортувати створені документи інвентаризації і приходу. Для цього існують методи

- OpenTable відкрити таблицю
- OpenTableWithParam відкрити таблицю з параметром
- GetTablerow наступний рядок
- GetTableValueAsFloat отримати значення поля типу число
- GetTableValueAsinteger отримати значення поля типу ціле
- GetTableValueAsVariant отримати значення поля типу варіант
- GetTableValueAsString отримати значення поля типу строка
- GetTableValueAsDate отримати значення поля типу дата

Опис методів

CREATEFILE (FILENAME:STRING):BOOLEAN

Створити файл обміну

FileName ім'я файлу бази даних, який необхідно створити. Якщо файл бази даних вже існує, то його буде відкрито.

Результат True або False

OpenFile(const AFileName: WideString): WordBool; stdcall;

Відкрити файл обміну.

FileName ім'я файлу бази даних

Результат True або False

CloseFile: WordBool; stdcall;

Закрити файл обміну.

Результат True або False

InsertDepot(idkey: Integer; Text: WideString): WordBool; stdcall;

Експортувати підрозділ

- Idkey – унікальне значення Integer
- text: назва підрозділу WideString

InsertSupplier(idkey: Integer; Text: WideString): WordBool; stdcall;

Експортувати постачальника

- Idkey – унікальне значення Integer
- text: назва постачальника WideString

Insertitem(idkey: Integer; const code1, text, idmeasure: WideString): WordBool;

- Idkey – унікальне значення Integer
- Code1 – код товару WideString
- text: назва постачальника WideString
- idmeasure: одиниця виміру WideString

InsertBarcode(idkey, idcommodity: Integer; const barcode: WideString; costprice, saleprice, quaintity: Double): WordBool; stdcall;

- idkey – унікальне значення (або 0, тоді унікальне значення буде створено автоматично)
- idcommodity посилання на таблицю товарів: Integer;
- const barcode штри код: WideString;
- costprice ціна закупки,
- saleprice ціна продажу
- quaintity: кількість

DeleteFromTable(const TableName: WideString): WordBool; stdcall;

Видалити дані з таблиці. Значення TableName

- DEALERSHIP01 зміст прихідних накладних

- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

OpenTable(const TableName: WideString): Integer; stdcall;

Відкрити таблицю. Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Результат кількість записів в таблиці

OpenTableWithParam(const TableName, ParamName: WideString; ParamValue: Integer): Integer; stdcall;

Видалити дані з таблиці з параметром. Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Paramname – ім'я параметра

ParamValue – значення параметра

Результат кількість записів в таблиці

GetTablerow(const TableName: WideString; RecordID: Integer): WordBool; stdcall;

Перейти на запис RecordID

Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Результат True False

GetTableValueAsFloat(const Tablename, FieldName: WideString): Single; safecall;

Отримати значення поля як число

Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Значення fieldname залежить **від таблиці**

GetTableValueAsinteger(const TableName, Fieldname: WideString): Integer;

stdcall;

Отримати значення поля як ціле число

Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Значення fieldname залежить **від таблиці**

GetTableValueAsVariant(const TableName, Fieldname: WideString): OleVariant;

safecall;

Отримати значення поля як варіант

Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Значення fieldname залежить **від таблиці**

GetTableValueAsString(const TableName, FieldName: WideString): WideString;

safecall;

Отримати значення поля як строка

Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи
- SUPPLIER постачальники
- BARCODES штрих коди

Значення fieldname залежить **від таблиці**

DeleteAll; safecall;

Очистити всі дані

GetTableValueAsDate(const Tablename, Fieldname: WideString): TDateTime; stdcall;

Отримати значення поля як дата

Значення TableName

- DEALERSHIP01 зміст прихідних накладних
- INVENTORYDATA зміст інвентаризації
- INVOICE01 заголовки прихідних накладних
- INVENTORY заголовки інвентаризації
- COMMODITY' товари
- DEPOTS підрозділи

- SUPPLIER постачальники
- BARCODES штрих коди

Значення fieldname залежить від таблиці

Структура бази даних

- Підрозділи

```
CREATE TABLE IF NOT EXISTS Depots
```

```
(idkey integer PRIMARY KEY,TEXT char(50));
```

- Постачальники

```
CREATE TABLE IF NOT EXISTS Supplier
```

```
(idkey integer PRIMARY KEY,TEXT char(50));
```

- Товари

```
CREATE TABLE IF NOT EXISTS Commodity
```

```
(idkey integer PRIMARY KEY,Code1 char(50),Text char(50),idmeashure char(50));
```

- Штрихкоди

```
CREATE TABLE IF NOT EXISTS BarCodes
```

```
(idkey integer PRIMARY KEY AUTOINCREMENT,idcommodity int,BarCode char(50),saleprice NUM,costprice num,param1 char(50),param2 num,quantity num);
```

- Заголовок прихідної накладної

```
CREATE TABLE IF NOT EXISTS Invoice01
```

```
(idkey integer PRIMARY KEY AUTOINCREMENT,iddepot int,idsupplier int,iddate char(10),idno char(50),CostAmount num,SaleAmount num,idcomment char(10));
```

- Зміст прихідної накладної

```
CREATE TABLE IF NOT EXISTS Dealership01 (idkey integer PRIMARY KEY AUTOINCREMENT,idinvoice01 int,idcommodity int,barcode text,saleprice NUM,costprice num,param1 char(50),param2 num,quantity num);
```

- Заголовок документу інвентаризації

```
CREATE TABLE IF NOT EXISTS Inventory
```

```
(idkey integer PRIMARY KEY AUTOINCREMENT,iddepot int,iddate char(10),idno char(50),idcomment char(10),CostAmount num,SaleAmount num);
```

- Зміст документу інвентаризації

```
CREATE TABLE IF NOT EXISTS inventoryData
```

```
(idkey integer PRIMARY KEY AUTOINCREMENT,idinvoice int,idcommodity int,barcode text,saleprice NUM,costprice num,param1 char(50),param2 num,quantity num);
```

Приклади

Експорт даних

```
sole:=CreateOleObject('PidkovaSA.Terminaldata');

if sole<>null then

    if sole.CreateFile('c:\temp \psa_data.db3') then

        begin

            sole.deleteAll; // очистити базу

            sole.insertDepot(1,'Магазин'); // підрозділ магазин

            sole.insertDepot(1,'Склад');

            sole.insertSupplier(1, 'Постачальник 1'); // Постачальник 1

            sole.insertSupplier(1, 'Постачальник 2');

            sole.insertItem(1,'1','Вода Шайянська','пл 0,5') // товар з кодом 1

            sole.insertItem(2,'2','Вода Живіця','пл 0,5') // товар з кодом 2

            sole.insertbarcode(0,1,'4820001010',10.00,12.00,5); // Вода Шайянська код товару 1, закупка 10, продаж 12, кількість 5

            sole.insertbarcode(0,2,'482000133',11.00,12.50,6);

            sole.CloseFile;

        end;
```

Імпорт даних Інвентаризація

Створюємо об'єкт

```
sole:=CreateOleObject('PidkovaSA.Terminaldata');
```

Відкриваємо файл

```
if sole.OpenFile('c:\temp \psa_data.db3') then
```

```
begin
```

```
    Rc1:=0;
```

```
    Rc2:=0;
```

```
MasT:='INVENTORY';
```

```
DetT:='INVENTORYDATA';
```

Відкриваємо заголовки документів

```
Rc1:=Sole.OpenTable(MasT);
```

Rc1 – кількість документів

```
for i:=1 to Rc1 do
```

Перебираємо записи

```
if Sole.GetTableRow(MasT,i) then
```

```
begin
```

```
    ID підрозділа:= sole.GetTableValueAsString(MasT,'IDDEPOT');
```

```
    Дата:=sole.GetTableValueAsString(MasT,'IDDATE');
```

```
    Примітка:=sole.GetTableValueAsString(MasT,'idcomment');
```

```
    Номер документа:=sole.GetTableValueAsString(MasT,'idno');
```

Відкриваємо зміст документа з параметром ID документа

```
    RC2:=sole.OpenTableWithParam(DetT,'idinvoice',sole.GetTableValueAsString(MasT,'idkey'));
```

```
    for j:=1 to Rc2 do
```

```
        if Sole.GetTableRow(DetT,j) then
```

```
            begin
```

```
                ID товару:= sole.GetTableValueAsString(DetT,'IDCOMMODITY');
```

```
                Штрих код:=Sole.GetTableValueAsString(DetT,'BARCODE');
```

```
                Ціна продажу:=ConvertDec(sole.GetTableValueAsString(DetT,'SALEPRICE'));
```

```
                Кількість:= sole.GetTableValueAsString(DetT,'QUANTITY');
```

```
            end;
```

```
        end;
```

```
sole.CloseTable('INVENTORY');
```

```
sole.CloseTable('INVENTORYDATA');
```

```
sole.CloseFile;
```

Імпорт даних Приходи

```
sole:=CreateOleObject('PidkovaSA.Terminaldata');
```

```
if sole.OpenFile('c:\temp\psa_data.db3') then
```

```
begin
```

```
    Rc1:=0;
```

```
    Rc2:=0;
```

```
MasT:='INVOICE01';
```

```
DetT:='DEALERSHIP01';
```



```
RC1:=Sole.OpenTable(MasT);

  for i:=1 to Rc1 do

    if Sole.GetTableRow(MasT,i) then

      begin

        ID підрозділа:= sole.GetTableValueAsString(MasT,'IDDEPOT');

        ID постачальника:= sole.GetTableValueAsString(MasT,'IDSUPPLIER');

        Дата:=sole.GetTableValueAsString(MasT,'IDDATE');

        Примітка:=sole.GetTableValueAsString(MasT,'idcomment');

        Номер документа:=sole.GetTableValueAsString(MasT,'idno');

        RC2:=sole.OpenTableWithParam(DetT,'idinvoice',sole.GetTableValueAsString(MasT,'idkey'));

        for j:=1 to Rc2 do

          if Sole.GetTableRow(DetT,j) then

            begin

              ID товару:= sole.GetTableValueAsString(DetT,'IDCOMMODITY');

              Штрих код:=Sole.GetTableValueAsString(DetT,'BARCODE');

              Ціна закупки:=ConvertDec(sole.GetTableValueAsString(DetT,'COSTPRICE'));

              Ціна продажу:=ConvertDec(sole.GetTableValueAsString(DetT,'SALEPRICE'));

              Кількість:= sole.GetTableValueAsString(DetT,'QUANTITY'));

            end;

          end;

        end;

      sole.CloseTable(MasT);

      sole.CloseTable(DetT);

      sole.CloseFile;
```