

ЗАКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО “НЕФТЕПРОМАВТОМАТИКА”

450047, г. Уфа, ул. Кулибина, 6
тел. (347) 242-39-37, т./факс (347) 242-92-72

e-mail: npa@ufacom.ru <http://www.npaufa.ru>

Инструкция по работе с библиотекой Terminal.dll

Оглавление

1.Установка библиотеки terminal.dll.....	3
2.Подключение библиотеки terminal.dll к среде разработки.....	4
3.Инициализация библиотеки.....	6
4.Изменение сообщений поднести карту к терминалу.....	8
5.Запрос карты	9
6.Чтение номера карты.....	10
7.Изменение MODBUS адреса, записанного в энергонезависимой памяти терминала.....	11
8.Изменение временных уставок для работы с сервером.....	12
9.Создание накладной на сервере.....	13
10.Удаление накладной на сервере.....	15
11.Чтение накладной на сервере.....	16
Приложение А - Описание интерфейса на языке IDL.....	19
Приложение Б – Таблица кодов ошибок.....	21
Приложение В – таблица кодов продуктов.....	22

1. Установка библиотеки terminal.dll

Скопируйте файл `terminal.dll` в любую папку, например `D:\LIB` и выполните команду регистрации ActiveX сервера: **`regsvr32 terminal.dll`**, файл `regsvr32.exe` должен находиться в какой-либо системной папке типа `C:\Windows\System\`. После выполнения данной команды, если все было сделано правильно, будет выведено сообщение об успешной регистрации ActiveX библиотеки в реестре.

2. Подключение библиотеки terminal.dll к среде разработки

Если ваша среда разработки поддерживает раннее связывание ActiveX объектов, то после импорта библиотеки типов terminal.tlb вы можете выполнить следующее:

Пример для Delphi:

```
var
iTerm: ICARD_TERMINAL_NB2;
ErrStr: array[0..26] of string=
  ('Выполнено успешно',
  'Адрес подчиненного не задан',
  'Неверное значение адреса',
  'СОМ-порт не инициализирован',
  'Номер порта должен быть от 1 до 20',
  'Ошибка открытия порта',
  'Неверное значение яркости',
  'Ошибка связи с подчиненным',
  'Ошибка преобразования кодировки',
  'Длина строки больше 20 символов',
  'Карта не была поднесена к терминалу',
  'Ошибка соединения с сервером',
  'Неверное значение хоста',
  'Значение таймаута должно быть от 1000 до 100000 мс',
  'Некорректный номер накладной',
  'Некорректное имя водителя',
  'Некорректный гос/номер автомобиля',
  'Некорректный номер секции',
  'Существует накладная с этим именем, но другим водителем',
  'Существует накладная с этим именем, но другим № автомобиля',
  'Неизвестная ошибка',
  'Существует накладная с этим именем, но другим № карты',
  'Неверное значение дозы',
  'Сервер не инициализирован',
  'Неверное значение кода продукта',
  'Накладная не найдена')
.....
iTerm:= CreateComObject(CLASS_CARD_TERMINAL_NB2) as ICARD_TERMINAL_NB2;
```

Пример для VB:

```
Dim iTerm As terminal.CARD_TERMINAL_NB2
....
Set iTerm = New CARD_TERMINAL_NB2
```

Описание **ICARD_TERMINAL_NB2** и **CLASS_CARD_TERMINAL_NB2** находится в библиотеке типов.

Если ваша среда разработки не поддерживает раннее связывание или вы его не хотите использовать, то вы можете выполнить следующее:

Пример для Delphi:

```
var  
iTerm: Variant;  
.....  
iTerm:= CreateOleObject('terminal.CARD_TERMINAL_NB2');
```

Пример для VB:

```
Dim iTerm As Object  
...  
Set iTerm = CreateObject("Terminal.CARD_TERMINAL_NB2")
```

Пример для среды 1С:

```
iTerm=СоздатьОбъект("terminal.CARD_TERMINAL_NB2");
```

После подключения библиотеки переменная **iTerm** будет использоваться для вызова функций библиотеки.

3. Инициализация библиотеки

Функция инициализации предназначена как для задания, так и изменения ранее заданных параметров работы системы. Она может быть вызвана как один раз, так и несколько, если нужно что-либо перенастроить. Описание функции инициализации на языке IDL (Interface Definition Language) следующее:

```
HRESULT _stdcall InitLib([in] unsigned char Address, [in] unsigned char CommPort, [in] unsigned char Brightness, [in] BSTR HostName, [in] VARIANT_BOOL SetAddress, [in] VARIANT_BOOL SetPort, [in] VARIANT_BOOL SetBrightness, [in] VARIANT_BOOL ClearInitFlag, [in] VARIANT_BOOL ConnectToServer, [out] long * SetAddressErr, [out] long * SetPortErr, [out] long * SetBrightnessErr, [out] long * ClearInitFlagErr, [out] long * ConnectToServerErr, [out, retval] long * Ret );
```

Входные параметры: **Address** – Modbus адрес терминала, к которому мы хотим подключиться; **CommPort** – число в диапазоне 1..20 – номер COM порта компьютера, к которому подключен терминал; **Brightness** – яркость дисплея терминала (0 – 100%, 1 – 75%, 2 – 50%, 3 – 25%); **HostName** – сетевое имя компьютера, на котором установлен сервер;

Следующие логические переменные указывают какие из вышеперечисленных параметров следует изменить, для этого их надо установить в True: **SetAddress** - требование изменить адрес, **SetPort** – требование изменить порт, **SetBrightness** – требование изменить яркость, **ClearInitFlag** – требование инициализировать терминал (у данной модели терминала делать это необязательно), **ConnectToServer** – установить соединение с сервером.

Следующие выходные переменные возвращают результат (код ошибки) затребованных нами действий, расшифровать который можно по таблице Б.1 приложения Б: **SetAddressErr** – результат попытки изменить (не переписать, для этого есть функция ChangeTerminalAddress) MODBUS адрес обращения к терминалу; **SetPortErr** – результат смены COM – порта; **SetBrightnessErr** – результат смены яркости; **ClearInitFlagErr** – результат инициализации терминала; **ConnectToServerErr** – результат соединения с сервером. **Ret** – не используется.

В следующем примере будет показано, как с помощью этой функции можно изменить порт подключения и яркость терминала (порт COM2, яркость – 75%):

Пример для Delphi:

```
var
  SetAddressErr,SetPortErr,SetBrightnessErr,ClearInitFlagErr,ConnectToServerErr: integer;
... ..
iTerm.InitLib(0,2,1,'',false,true,true,false,false,SetAddressErr,SetPortErr,SetBrightnessErr,Clear
InitFlagErr,ConnectToServerErr);
ShowMessage('Результат смены порта: '+ErrStr[SetPortErr]+'#13+'Результат изменения
яркости: '+ErrStr[SetBrightnessErr]);
```

Пример для VB:

```
Dim SetAddressErr As Long
Dim SetPortErr As Long
Dim SetBrightnessErr As Long
Dim ClearInitFlagErr As Long
```

Dim ConnectToServerError As Long

...

ret=iTerm.InitLib(0,2,1,"",false,true,true,false,false,SetAddressErr,SetPortErr, SetBrightnessErr, ClearInitFlagErr, ConnectToServerError)

TextBox1.Text = "Результат смены порта: " + ErrStr(SetPortErr) + " Результат изменения яркости: " + ErrStr(SetBrightnessErr)

4. Изменение сообщений поднести карту к терминалу

Когда терминал просит вас поднести карту, то он выводит 2 строки сообщения: Str1= 'поднесите пожалуйста' и Str2='карту к терминалу'. После прочтения карты терминал выведет другие строки на экран: Str3='номер карты был' Str4='успешно считан'. Если вы хотите вывести другие строки на экран, то при загрузке своей программы вызывайте следующую функцию:

```
HRESULT _stdcall LoadStrings([in] BSTR Str1, [in] BSTR Str2, [in] BSTR Str3, [in] BSTR Str4, [out, retval] long * Ret );
```

Входные параметры **Str1, Str2, Str3, Str4** были рассмотрены выше, длина строк не более 20 символов; код ошибки выводится в выходной параметр **Ret**.

Пример для Delphi:

```
var  
ret: Integer;  
... ..  
ret:= iTerm.LoadStrings('сергей владимирович','поднеси карту','спасибо','большое');  
ShowMessage('Результат изменения строк: '+ErrStr[ret]);
```

Пример для VB:

```
Dim ret As Long  
... ..  
ret = iTerm.LoadStrings("сергей Владимирович","поднеси карту","спасибо","большое")  
TextBox1.Text = "Результат изменения строк: "+ErrStr(ret)
```


5. Запрос карты

После выполнения этой функции на экране терминала появятся строки ‘поднесите пожалуйста’ и ‘карту к терминалу’, которые можно переопределить описанной выше функцией, и терминал перейдет в режим ожидания карты. После того, как кто-либо поднесет карту к терминалу – на экране терминала появятся следующие строки: ‘номер карты был’ и ‘успешно прочитан’, если, конечно эти сообщения не были переопределены описанной выше функцией. Формат функции следующий:

```
HRESULT _stdcall RequestCard([out, retval] long * Ret );
```

Данная функция возвращает только результат выполнения.

Пример для Delphi:

```
var  
  ret: Integer;  
... ..  
ret:= iTerm. RequestCard;  
ShowMessage(‘Результат запроса: ’+ErrStr[ret]);
```

Пример для VB:

```
Dim ret As Long  
... ..  
ret = iTerm. RequestCard  
TextBox1.Text = “Результат запроса: “+ErrStr(ret)
```

6. Чтение номера карты

Чтение номера карты нужно вызывать после запроса карты, описанной выше функцией, и после того, как на терминале появятся строки: 'номер карты был' и 'успешно прочитан'. Если запрос карты был послан (описанной выше функцией), но карта не была поднесена, то функция чтения карты вернет код ошибки *10 – карта не была поднесена к терминалу*. Формат функции следующий:

```
HRESULT _stdcall ReadCardNumber([out] long * Numb, [out, retval] long * Ret );
```

Данная функция содержит выходной параметр **Numb** – прочитанный код карты, и возвращает код результата.

Пример для Delphi:

```
var  
  ret, numb: Integer;  
  ... ..  
ret:= iTerm.ReadCardNumber(numb);  
if ret in [0..26] then LabeledEdit7.Text:='Результат выполнения: '+ErrStr[ret];  
if ret=0 then Edit2.Text:='Код карты: '+IntToStr(numb);
```

Пример для VB:

```
Dim ret As Long  
Dim numb As Long  
... ..  
ret = iTerm.ReadCardNumber(numb)  
if (ret >= 0) And (ret <= 25) Then TextBox1.Text = "Результат выполнения: " + ErrStr(ret)  
if ret = 0 Then TextBox1 = numb
```

7. Изменение MODBUS адреса, записанного в энергонезависимой памяти терминала

Формат функции, осуществляющей переадресацию адреса, следующий:

HRESULT _stdcall ChangeTerminalAddress([in] unsigned char OldAddress, [in] unsigned char NewAddress, [out, retval] long * Ret);

Функция содержит два входных параметра: **OldAddress** – текущий адрес терминала и **NewAddress** – новое значение адреса, которое нужно записать в память. Функция возвращает код результата.

Пример для Delphi:

```
var
  ret: Integer;
... ..
ret:= iTerm. ChangeTerminalAddress(247,1);
if ret in [0..26] then LabeledEdit7.Text:='Результат выполнения: '+ErrStr[ret];
```

Пример для VB:

```
Dim ret As Long
... ..
ret = iTerm. ChangeTerminalAddress(247,1)
If (ret >= 0) And (ret <= 25) Then TextBox1.Text = "Результат выполнения: " + ErrStr(ret)
```

8. Изменение временных уставок для работы с сервером

Данная программа через локальную сеть взаимодействует с особым компьютером – сервером управления отпуском нефтепродуктов. Для смены заданных по умолчанию временных задержек работы с сервером используйте следующую функцию:

```
HRESULT _stdcall ChangeServerTimeouts([in] long ConnTMOU, [in] long FuncTMOU, [out, retval] long * Ret );
```

Входные параметры: **ConnTMOU** – время ожидания соединения с сервером (после разрыва); **FuncTMOU** – время ожидания выполнения функции на сервере (если соединение с сервером уже установлено). Допустимые значения этих параметров от 1000 мс до 100000 мс, типовые: 12000 мс и 7000 мс соответственно. Если сеть работает плохо, то рекомендуется увеличить эти значения и вызывать эту функцию при старте программы. Функция возвращает код результата.

Пример для Delphi:

```
var  
ret: Integer;  
... ..  
ret:= iTerm. ChangeServerTimeouts (15000,10000);  
if ret in [0..26] then LabeledEdit7.Text:='Результат выполнения: '+ErrStr[ret];
```

Пример для VB:

```
Dim ret As Long  
... ..  
ret = iTerm. ChangeServerTimeouts (15000,10000)  
If (ret >= 0) And (ret <= 25) then TextBox1.Text = "Результат выполнения: "+ErrStr(ret)
```

9. Создание накладной на сервере

Формат функции, которая создает накладную на сервере следующий:

```
HRESULT _stdcall CreateNakl([in] BSTR N_Nakl, [in] BSTR Driver, [in] BSTR AutoNum, [in] long Numb, [in] long VolS1, [in] long VolS2, [in] long VolS3, [in] long VolS4, [in] long VolS5, [in] long VolS6, [in] long VolS7, [in] long VolS8, [in] long VolS9, [in] long VolS10, [in] long FuS1, [in] long FuS2, [in] long FuS3, [in] long FuS4, [in] long FuS5, [in] long FuS6, [in] long FuS7, [in] long FuS8, [in] long FuS9, [in] long FuS10, [out] long * S1Err, [out] long * S2Err, [out] long * S3Err, [out] long * S4Err, [out] long * S5Err, [out] long * S6Err, [out] long * S7Err, [out] long * S8Err, [out] long * S9Err, [out] long * S10Err, [out, retval] long * Ret );
```

Входные параметры: **N_Nakl** – номер накладной - строка до 47 символов; **Driver** – имя водителя – строка до 50 символов; **AutoNum** – номер автомобиля – строка до 50 символов; **Numb** – номер карты, считанный функцией **ReadCardNumber**; **VolS1...VolS10** – объемы секций 1..10 или число 0, если какую-либо секцию заливать не надо (или секция отсутствует); **FuS1...FuS10** – коды нефтепродуктов для секций 1..10 (число от 1 до 52, смотри приложение В)

Выходные параметры: **S1Err...S10Err** – коды результата создания в накладной секций 1..10, данные коды следует анализировать только если функция вернула общий код результата - 0. Функция возвращает также общий код результата.

В следующем примере будет создана накладная с номером 001, двумя секциями по 4500 л. и 9000 л. бензина АИ-80.

Пример для Delphi:

```
var
  ret: Integer;
  N_Nakl,Driver,AutoNum: WideString;
  S1Err, S2Err, S3Err, S4Err, S5Err, S6Err,S7Err, S8Err, S9Err, S10Err: Integer;
  Numb: integer;
... ..
N_Nakl:='001';
Driver:='Коземаслов Андрей Валентинович';
AutoNum:='A619YC';
Numb:=-72897564;

ret:= iTerm.CreateNakl(N_Nakl,Driver,AutoNum,Numb,4500,9000,0,0,0,0,0,0,0,0,5,5,1,1,1,1,1,1,1,1, S1Err, S2Err, S3Err, S4Err, S5Err, S6Err,S7Err, S8Err, S9Err, S10Err);

if ret in [0..26] then Edit1.Text:='Результат выполнения: '+ErrStr[ret];
if ret=0 then
begin
  if S1Err in [0..26] then Edit2.Text:='Результат добавления 1-ой секции: '+ErrStr[S1Err];
  if S2Err in [0..26] then Edit3.Text:='Результат добавления 2-ой секции: '+ErrStr[S2Err];
end;
```

Пример для VB:

```
Dim N_Nakl, Driver, AutoNum As String
Dim Numb As Long
```

```

Dim S1Err As Long
Dim S2Err As Long
Dim S3Err As Long
Dim S4Err As Long
Dim S5Err As Long
Dim S6Err As Long
Dim S7Err As Long
Dim S8Err As Long
Dim S9Err As Long
Dim S10Err As Long
Dim ret As Long

... ...
N_Nakl = "001"
Driver = "Коземаслов Андрей Валентинович"
AutoNum = "A619YC"
Numb = -72897564

ret = iTerm.CreateNakl(N_Nakl,Driver,AutoNum,Numb,4500,9000,0,0,0,0,0,0,0,0,5,5,1,1,1,1,1, 1,
1, S1Err, S2Err, S3Err, S4Err, S5Err, S6Err,S7Err, S8Err, S9Err, S10Err)

if (ret >= 0) And (ret <= 25) then TextBox1.Text = "Результат выполнения:" +ErrStr(ret)
if ret=0 then
  if (S1Err >= 0) And (S1Err <= 25) then TextBox2.Text = "Результат добавления 1-ой секции:
"+ErrStr(S1Err)
  if (S2Err >= 0) And (S2Err <= 25) then TextBox3.Text = "Результат добавления 2-ой секции:
"+ErrStr(S2Err)
end if

```

10. Удаление накладной на сервере

Данной функцией можно удалять только те секции накладной на сервере, которые не были исполнены и не исполняются (заправляются) в данный момент:

```
HRESULT _stdcall DeleteNakl([in] BSTR N_Nakl, [in] unsigned char Sect, [out] long * Fnd, [out] long * Dltd, [out, retval] long * Ret );
```

Входные параметры: **N_Nakl** – номер накладной - строка до 47 символов; **Sect** – номер секции от 1 до 10, которую нужно удалить на сервере из накладной или 0, если нужно удалить все секции.

Выходные параметры: **Fnd** – количество секций данной накладной, которые были найдены на сервере; **Dltd** – которые были удалены (остальные либо исполняются в данный момент, либо уже исполнены). Функция возвращает код результата.

Пример для Delphi:

```
var
  ret, Fnd, Dltd: Integer;
  N_Nakl: WideString;
  Sect: byte;
... ..
N_Nakl:='001';
Sect:=0;           //удалить все секции
ret:=DeleteNakl(N_Nakl,Sect,Fnd,Dltd);
if ret in [0..26] then Edit1.Text:='Результат выполнения: '+ErrStr[ret];
if ret=0 then
begin
  Edit1.Text:='Найдено секций: '+IntToStr(Fnd);
  Edit2.Text:='Удалено секций: '+IntToStr(Dltd);
end;
```

Пример для VB:

```
var
Dim ret As Long
Dim Fnd As Long
Dim Dltd As Long
Dim N_Nakl As String
Dim Sect As byte
... ..
N_Nakl = "001"
Sect= 0           //удалить все секции
ret = DeleteNakl(N_Nakl,Sect,Fnd,Dltd)
if (ret >= 0) And (ret <= 25) then TextBox1.Text = "Результат выполнения: "+ErrStr(ret)
if ret=0 then
  TextBox1.Text = "Найдено секций: "+Str(Fnd)
  TextBox2.Text = "Удалено секций: "+Str(Dltd)
end if
```

11. Чтение накладной на сервере

Для чтения накладной на сервере используется следующая функция:

```
HRESULT _stdcall ReadNakl([in] BSTR N_Nakl, [out] unsigned char * Stat1, [out] unsigned char * Stat2, [out] unsigned char * Stat3, [out] unsigned char * Stat4, [out] unsigned char * Stat5, [out] unsigned char * Stat6, [out] unsigned char * Stat7, [out] unsigned char * Stat8, [out] unsigned char * Stat9, [out] unsigned char * Stat10, [out] long * VolOtp1, [out] long * VolOtp2, [out] long * VolOtp3, [out] long * VolOtp4, [out] long * VolOtp5, [out] long * VolOtp6, [out] long * VolOtp7, [out] long * VolOtp8, [out] long * VolOtp9, [out] long * VolOtp10, [out] double * MassOtp1, [out] double * MassOtp2, [out] double * MassOtp3, [out] double * MassOtp4, [out] double * MassOtp5, [out] double * MassOtp6, [out] double * MassOtp7, [out] double * MassOtp8, [out] double * MassOtp9, [out] double * MassOtp10, [out] double * T1, [out] double * T2, [out] double * T3, [out] double * T4, [out] double * T5, [out] double * T6, [out] double * T7, [out] double * T8, [out] double * T9, [out] double * T10, [out] DATE * DateTime1, [out] DATE * DateTime2, [out] DATE * DateTime3, [out] DATE * DateTime4, [out] DATE * DateTime5, [out] DATE * DateTime6, [out] DATE * DateTime7, [out] DATE * DateTime8, [out] DATE * DateTime9, [out] DATE * DateTime10, [out, retval] long * Ret)
```

Входной параметр: **N_Nakl** – номер накладной - строка до 47 символов;

Выходные параметры: **Stat1...Stat10** – состояния секций 1..10 данной накладной, расшифровка значений: 0 – Не исполнена, 1 – Выполняется, 2 – Исполнена, 3 – Секция отсутствует, 4 – Не исполнена и отменена операторами. **VolOtp1..VolOtp10** – объем в литрах, отпущенный в секции 1..10 – имеет смысл анализировать только для исполненных секций. **MassOtp1..MassOtp10** – масса в тоннах, отпущенная в секции 1..10 – имеет смысл анализировать только для исполненных секций. **T1..T10** – температура продукта, залитого в секции 1..10 – имеет смысл анализировать только для исполненных секций. **DateTime1..DateTime10** – время конца заправки секции 1..10 – имеет смысл анализировать только для исполненных секций.

Пример для Delphi:

```
var
```

```
  IspArr: array[0..4] of string=('не исполнена','выполняется','Исполнена','Не найдена',' Не исполнена и отменена');
```

```
  ret: integer;
```

```
  N_Nakl: WideString;
```

```
  Stat1,Stat2,Stat3,Stat4,Stat5,Stat6,Stat7,Stat8,Stat9,Stat10: Byte;
```

```
  VolOtp1,VolOtp2,VolOtp3,VolOtp4,VolOtp5,VolOtp6,VolOtp7,VolOtp8,VolOtp9,VolOtp10: Integer;
```

```
  MassOtp1,MassOtp2,MassOtp3,MassOtp4,MassOtp5,MassOtp6,MassOtp7,MassOtp8,MassOtp9,MassOtp10: Double;
```

```
  T1,T2,T3,T4,T5,T6,T7,T8,T9,T10: double;
```

```
  DT1,DT2,DT3,DT4,DT5,DT6,DT7,DT8,DT9,DT10: TDateTime;
```

```
  ... ..
```

```
  LabeledEdit17.Clear;
```

```
  N_Nakl:=Edit18.Text;
```

```
  Memo2.Lines.Clear;
```

```
  ret:= iTerm.ReadNakl(N_Nakl,Stat1,Stat2,Stat3,Stat4,Stat5,Stat6,Stat7,Stat8,Stat9,Stat10,
```

```
    VolOtp1,VolOtp2,VolOtp3,VolOtp4,VolOtp5,VolOtp6,VolOtp7,VolOtp8,VolOtp9,VolOtp10,MassOtp1,MassOtp2,MassOtp3,MassOtp4,MassOtp5,MassOtp6,MassOtp7,MassOtp8,MassOtp9,MassOtp10,
```



```
T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,
DT1,DT2,DT3,DT4,DT5,DT6,DT7,DT8,DT9,DT10);
if ret in [0..26] then LabeledEdit17.Text:=ErrStr[ret];
if ret <> 0 then Exit;
```

```
if Stat1 <> 3 then Memo2.Lines.Add('Сек.1: V отп=' + IntToStr(VolOtp1) + 'л М отп=' +
    FloatToStrF(MassOtp1, ffFixed, 6, 3) + 'г Темп.:=' + FloatToStrF(T1, ffFixed, 3, 1) + ' дата
'+ DateTimeToStr(DT1) + ' ' + IspArr[Stat1]);
```

... ..

```
if Stat10 <> 3 then Memo2.Lines.Add('Сек.10: V отп=' + IntToStr(VolOtp10) + 'л М отп=' +
    FloatToStrF(MassOtp10, ffFixed, 6, 3) + 'г Темп.:=' + FloatToStrF(T10, ffFixed, 3, 1) + ' дата
'+ DateTimeToStr(DT10) + ' ' + IspArr[Stat10]);
```

Пример для VB:

```
Dim IspArr As Variant
```

```
IspArr = Array("не исполнена", "выполняется", "Исполнена", "Не найдена", "Не исполнена и
отменена")
```

```
Dim ret As Long
```

```
Dim N_Nakl As String
```

```
Dim Stat1 As Byte
```

```
Dim Stat2 As Byte
```

```
Dim Stat3 As Byte
```

```
Dim Stat4 As Byte
```

```
Dim Stat5 As Byte
```

```
Dim Stat6 As Byte
```

```
Dim Stat7 As Byte
```

```
Dim Stat8 As Byte
```

```
Dim Stat9 As Byte
```

```
Dim Stat10 As Byte
```

```
Dim VolOtp1 As Long
```

```
Dim VolOtp2 As Long
```

```
Dim VolOtp3 As Long
```

```
Dim VolOtp4 As Long
```

```
Dim VolOtp5 As Long
```

```
Dim VolOtp6 As Long
```

```
Dim VolOtp7 As Long
```

```
Dim VolOtp8 As Long
```

```
Dim VolOtp9 As Long
```

```
Dim VolOtp10 As Long
```

```
Dim MassOtp1 As Double
```

```
Dim MassOtp2 As Double
```

```
Dim MassOtp3 As Double
```

```
Dim MassOtp4 As Double
```

```
Dim MassOtp5 As Double
```

```
Dim MassOtp6 As Double
```

```
Dim MassOtp7 As Double
```

```
Dim MassOtp8 As Double
```

```
Dim MassOtp9 As Double
```

Dim MassOtp10 As Double

Dim T1 As Double

Dim T2 As Double

Dim T3 As Double

Dim T4 As Double

Dim T5 As Double

Dim T6 As Double

Dim T7 As Double

Dim T8 As Double

Dim T9 As Double

Dim T10 As Double

Dim DT1 As Date

Dim DT2 As Date

Dim DT3 As Date

Dim DT4 As Date

Dim DT5 As Date

Dim DT6 As Date

Dim DT7 As Date

Dim DT8 As Date

Dim DT9 As Date

Dim DT10 As Date

... ..

N_Nakl = TextBox20.Text

**ret = iTerm.ReadNakl(N_Nakl,Stat1,Stat2,Stat3,Stat4,Stat5,Stat6,Stat7,Stat8,Stat9,Stat10,
VolOtp1,VolOtp2,VolOtp3,VolOtp4,VolOtp5,VolOtp6,VolOtp7,VolOtp8,VolOtp9,VolOtp10,
MassOtp1,MassOtp2,MassOtp3,MassOtp4,MassOtp5,MassOtp6,MassOtp7,MassOtp8,MassOtp9,
MassOtp10,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,DT1,DT2,DT3,DT4,DT5,DT6,DT7,DT8,DT9,DT10)
if (ret >= 0) And (ret <= 25) then TextBox17.Text = ErrStr(ret)**

**if Stat1<>3 then TextBox30.text = “Сек.1: V отп=”+Str(VolOtp1)+“л М отп=”+
Str(MassOtp1)+“т Темп.:=”+Str(T1)+“ ”+IspArr(Stat1)**

... ..

**if Stat10<>3 then TextBox40.text = “Сек.10: V отп=”+Str(VolOtp10)+“л М отп=”+
Str(MassOtp10)+“т Темп.:=”+Str(T10)+ “ ” + IspArr(Stat10)**

Приложение А - Описание интерфейса на языке IDL

```
[
  uuid(909A26BE-1D34-454A-AD85-63B62CFF1AA5),
  version(1.0),
  helpstring("terminal Library")
]
library terminal
{

  importlib("stdole2.tlb");

  [
    uuid(8258F872-B119-4C13-8401-641B8A94B8DE),
    version(1.0),
    helpstring("Dispatch interface for CARD_TERMINAL_NB2 Object"),
    dual,
    oleautomation
  ]
  interface ICARD_TERMINAL_NB2: IDispatch
  {
    [
      id(0x000000C9)
    ]
    HRESULT _stdcall InitLib([in] unsigned char Address, [in] unsigned char CommPort, [in] unsigned char
    Brightness, [in] BSTR HostName, [in] VARIANT_BOOL SetAddress, [in] VARIANT_BOOL SetPort, [in]
    VARIANT_BOOL SetBrightness, [in] VARIANT_BOOL ClearInitFlag, [in] VARIANT_BOOL
    ConnectToServer, [out] long * SetAddressErr, [out] long * SetPortErr, [out] long * SetBrightnessErr, [out]
    long * ClearInitFlagErr, [out] long * ConnectToServerErr, [out, retval] long * Ret );
    [
      id(0x000000CA)
    ]
    HRESULT _stdcall LoadStrings([in] BSTR Str1, [in] BSTR Str2, [in] BSTR Str3, [in] BSTR Str4, [out,
    retval] long * Ret );
    [
      id(0x000000CB)
    ]
    HRESULT _stdcall RequestCard([out, retval] long * Ret );
    [
      id(0x000000CC)
    ]
    HRESULT _stdcall ReadCardNumber([out] long * Numb, [out, retval] long * Ret );
    [
      id(0x000000CD)
    ]
    HRESULT _stdcall ChangeTerminalAddress([in] unsigned char OldAddress, [in] unsigned char
    NewAddress, [out, retval] long * Ret );
    [

```

```

    id(0x000000CE)
]
HRESULT _stdcall ChangeServerTimeouts([in] long ConnTMOUT, [in] long FuncTMOUT, [out, retval]
long * Ret );
[
    id(0x000000CF)
]
HRESULT _stdcall CreateNakl([in] BSTR N_Nakl, [in] BSTR Driver, [in] BSTR AutoNum, [in] long
Numb, [in] long VolS1, [in] long VolS2, [in] long VolS3, [in] long VolS4, [in] long VolS5, [in] long VolS6,
[in] long VolS7, [in] long VolS8, [in] long VolS9, [in] long VolS10, [in] long FuS1, [in] long FuS2, [in]
long FuS3, [in] long FuS4, [in] long FuS5, [in] long FuS6, [in] long FuS7, [in] long FuS8, [in] long FuS9,
[in] long FuS10, [out] long * S1Err, [out] long * S2Err, [out] long * S3Err, [out] long * S4Err, [out] long *
S5Err, [out] long * S6Err, [out] long * S7Err, [out] long * S8Err, [out] long * S9Err, [out] long * S10Err,
[out, retval] long * Ret );
[
    id(0x000000D0)
]
HRESULT _stdcall DeleteNakl([in] BSTR N_Nakl, [in] unsigned char Sect, [out] long * Fnd, [out] long
* Dltd, [out, retval] long * Ret );
[
    id(0x000000D1)
]
HRESULT _stdcall ReadNakl([in] BSTR N_Nakl, [out] unsigned char * Stat1, [out] unsigned char *
Stat2, [out] unsigned char * Stat3, [out] unsigned char * Stat4, [out] unsigned char * Stat5, [out] unsigned
char * Stat6, [out] unsigned char * Stat7, [out] unsigned char * Stat8, [out] unsigned char * Stat9, [out]
unsigned char * Stat10, [out] long * VolOtp1, [out] long * VolOtp2, [out] long * VolOtp3, [out] long *
VolOtp4, [out] long * VolOtp5, [out] long * VolOtp6, [out] long * VolOtp7, [out] long * VolOtp8, [out]
long * VolOtp9, [out] long * VolOtp10, [out] double * MassOtp1, [out] double * MassOtp2, [out] double *
MassOtp3, [out] double * MassOtp4, [out] double * MassOtp5, [out] double * MassOtp6, [out] double *
MassOtp7, [out] double * MassOtp8, [out] double * MassOtp9, [out] double * MassOtp10, [out] double *
T1, [out] double * T2, [out] double * T3, [out] double * T4, [out] double * T5, [out] double * T6, [out]
double * T7, [out] double * T8, [out] double * T9, [out] double * T10, [out] DATE * DateTime1, [out]
DATE * DateTime2, [out] DATE * DateTime3, [out] DATE * DateTime4, [out] DATE * DateTime5, [out]
DATE * DateTime6, [out] DATE * DateTime7, [out] DATE * DateTime8, [out] DATE * DateTime9, [out]
DATE * DateTime10, [out, retval] long * Ret)
};

[
    uuid(2E18194D-15DA-4186-9E90-C9A98708DEB2),
    version(1.0),
    helpstring("CARD_TERMINAL_NB2 Object")
]
coclass CARD_TERMINAL_NB2
{
    [default] interface ICARD_TERMINAL_NB2;
};

};

```

Приложение Б – Таблица кодов ошибок

Код	Описание
0	Выполнено успешно
1	Адрес подчиненного не задан
2	Неверное значение адреса
3	СОМ-порт не инициализирован
4	Номер порта должен быть от 1 до 20
5	Ошибка открытия порта
6	Неверное значение яркости
7	Ошибка связи с подчиненным
8	Ошибка преобразования кодировки
9	Длина строки больше 20 символов
10	Карта не была поднесена к терминалу
11	Ошибка соединения с сервером
12	Неверное значение хоста
13	Значение таймаута должно быть от 1000 до 100000 мс
14	Некорректный номер накладной
15	Некорректное имя водителя
16	Некорректный гос/номер автомобиля
17	Некорректный номер секции
18	Существует накладная с этим именем, но другим водителем
19	Существует накладная с этим именем, но другим № автомобиля
20	Неизвестная ошибка
21	Существует накладная с этим именем, но другим № карты
22	Неверное значение дозы
23	Сервер не инициализирован
24	Неверное значение кода продукта
25	Накладная не найдена
26	Накладная с таким номером и секцией уже существует

Таблица Б.1 – Расшифровка кода (кодов) ответа функции библиотеки terminal.dll

Приложение В – таблица кодов продуктов

Код	Нефтепродукт
1	А-76 неэтил.
2	АИ-92 неэтил.
3	А-76 этил.
4	АИ-95 неэтил.
5	Бензин АИ-80 н/эт.
6	Керосин КО
7	Диз. Топливо зимнее
8	Диз. Топливо летнее
9	Бензин АИ-98 н/эт
10	SAE 10W30SE/CC

Таблица В.1 – Расшифровка кода продукта