

EuroApteka

Dokumentacja techniczna

Spis treści:

PROPONOWANY PLAN WDROŻENIA.....	5
Instalacja.....	5
Konwersja danych IA do EA.....	5
Konfiguracja.....	5
Wybór formatów wydruku i weryfikacja podstawowych dokumentów i zestawień, ustalenie liczby kopii.....	5
Szkolenie użytkowników.....	5
Wprowadzenie.....	5
Ekspedycja.....	6
Sprzedaż.....	6
Wytwarzanie.....	6
Rachunki refundacyjne.....	6
Zakupy.....	6
Raporty finansowe.....	6
Raporty farmaceutyczne.....	6
Magazyn.....	7
Płatności.....	7
Administrator.....	7
INSTALACJA JEDNOSTANOWISKOWA - SYSTEM MS WINDOWS NT/95.....	7
Czynności wykonywane przez program instalacyjny:.....	7
Opis przeznaczenia i położenia plików:.....	8
Program główny.....	8
Bazy danych.....	8
Raporty.....	9
Pliki SQL Anywhere 5.5.....	9
Pliki ODBC.....	9
Biblioteki Power++ v 2.5.....	9
Pliki HTML.....	9
Pliki DataWindow v 6.0.....	9
Inne pliki.....	10
ODINSTALOWANIE PROGRAMU.....	10
PONOWNA INSTALACJA PROGRAMU.....	11
USTAWIENIA ODBC.....	11
Baza startowa – EuroAptekaStart.....	11
Baza demo – EuroAptekaDemo.....	11
KONWERSJA BAZY DANYCH: INFOAPTEKA->EUROAPTEKA.....	11

Eksport danych z InfoApteki:.....	11
Wczytanie danych do bazy EuroApteki:.....	12
PLIKI KONFIGURACYJNE *.INI.....	12
Sekcje.....	12
Lista.....	12
Formularz.....	13
DataColumns.....	13
SQL.....	14
User, Date, Status.....	14
Default.....	15
Większy przykład.....	15
Nazwa pliku.....	16
ListColumnN.....	16
Kopiowanie dokumentów.....	17
Inne pola występujące w pliku ini.....	17
Zerowanie numeru dokumentu.....	17
Liczba numerków.....	17
Opłata za dyżur nocny.....	18
Ostrzeżenie przed końcem daty ważności.....	18
Sekcje Stylu – plik “iastyle.ini”.....	18
ZAMÓWIENIA ZE STANÓW MAGAZYNOWYCH.....	19
OBLICZANIE CENY DETALICZNEJ.....	19

Proponowany plan wdrożenia

Instalacja

SQL Anywhere w sieci
Instalacja EA na serwerze

SQL Anywhere na stanowisku
EA na każdym stanowisku

Konwersja danych IA do EA

Weryfikacja ilościowa i wartościowa magazynu.

Konfiguracja

Wprowadzenie danych Właściciela.
Wprowadzenie danych użytkowników i kwalifikacja użytkowników do grup.
Ustawienie dla stanowisk typów drukarek fiskalnych.
Sprawdzenie poprawności nazw fiskalnych stawek VAT.
Weryfikacja formuł numerów dokumentów i wprowadzenie początkowych numerów dokumentów sprzedaży.

Wybór formatów wydruku i weryfikacja podstawowych dokumentów i zestawień, ustalenie liczby kopii

Rejestr zakupów VAT
Faktura VAT
Rejestr sprzedaży VAT
Protokół zwrotu
Rejestr zwrotów
Taśma refundacyjna
Rachunek refundacyjny
Zbiorcze zestawienie refundacji
Rejestr obrotów psychotropów i narkotyków

Szkolenie użytkowników

Wprowadzenie

Organizacja programu:

Rejestry dokumentów, Kartoteki danych

Zakres programu:

Zakupy, Sprzedaż, Magazyn, Wytwarzanie, Płatności, Administrator

Ekspedycja

Sprzedaż leku:

- na receptę
- z fakturą

Ustawienia ekspedycji:

- Druk wyceny
- Druk numerka

Przyjęcie zlecenia na lek recepturowy

Przyjęcie zaliczki na lek

Wydanie leku recepturowego

Sprzedaż

Zwroty

Tworzenie dokumentów zwrotu

Wystawianie faktury do paragonu sprzedaży

Wytwarzanie

Określanie składu leku recepturowego

Wydruk wyceny leku recepturowego

Rachunki refundacyjne

Tworzenie rachunków refundacyjnego

Wydruk rachunku refundacyjnego

- dla wybranej kasy chorych
- zbiorczego

Zakupy

Przejęcie faktury zakupu

odczyt z pliku

pasowanie leku

Tworzenie nowego artykułu

Raporty finansowe

Rejestr zakupów VAT

Rejestr sprzedaży VAT

Rejestr zwrotów

Raport kasowy

Raport finansowy

Sprzedaż w poszczególnych dniach

- miesiąca
- tygodnia

Raporty farmaceutyczne

Rejestr obrotów psychotropów i narkotyków

Magazyn

Dokumenty magazynowe

Likwidacje

Wprowadzanie zmian w magazynie za pomocą dokumentów magazynowych

Płatności

Wpłaty i wypłaty z kasy

Przelewy za faktury zakupu

Administrator

Dodawanie nowych użytkowników

Przydzielanie użytkowników do grup

Instalacja jednostanowiskowa - system MS Windows NT/95

Program EuroApteka instaluje się poprzez uruchomienie programu Setup.exe znajdujący się na płycie CD. Program ten został wygenerowany przy pomocy programu InstallShield Express pochodzącego z pakietu Powersoft Power++ v. 2.5.

Czynności wykonywane przez program instalacyjny:

Instalacja:

0ODBC

1środowiska Sybase SQL Anywhere 5.5

Konfiguracja:

2ODBC -tworzenie źródeł danych, modyfikacja rejestrów systemowych.

Kopiowanie plików:

3pliku wykonywalnego ea.exe

4bazy danych i odpowiedzialnych za połączenie z bazą

5bibliotek dll, z których bezpośrednio korzysta plik wykonywalny ea.exe

62 plików konfiguracyjny ea.ini

7bibliotek pbl zawierających definicje raportów

8bibliotek dll Power++

9HTML wraz z plikami bmp zawierającymi instrukcję obsługi programu oraz informacje na jego temat

Opis przeznaczenia i położenia plików:

W wyniku instalacji umieszczane są na dysku wymienione poniżej pliki (domyślny katalog, w którym instaluje się program D:\Program Files\EuroSoft\EuroApteka\):

Program główny

(położenie - katalog wybrany podczas instalacji programu)

10ea.exe
11ea.ini
12iaStyle.ini
13Admin.dll
14Exped.dll
15Request.dll
16Report.dll
17Compute.dll
18Farma.dll
19Fiscal.dll
20Invoice.dll
21PayDoc.dll
22Packs.dll
23Process.dll
24Receipt.dll
25StoreDoc.dll
26Table.dll
27Document.dll
28PGlobals.dll
29IAGlobal.dll
30IAExpr.dll
31IAComm.dll
32IAFiscal.dll
33IAProcess.dll
34IAApp.dll
35IAText.dll
36IAReport.dll
37IAAction.dll
38IADoc.dll

Bazy danych

(położenie – podkatalog \dbStart\ i \dbDemo katalogu bieżącego programu - ea.exe)

39EuroApteka.db
40EuroApteka.log

Raporty

(położenie - podkatalog \pbl\ katalogu bieżącego programu)

41pliki o rozszerzeniu pbl

Pliki SQL Anywhere 5.5

(położenie - podkatalog \Sqlany50\Win32 katalogu bieżącego programu)

42dbl50t.dll
43rtdsk50.exe
44wl50ent.dll
45wod50t.dll
46wtr50t.dll
47dbeng50.exe

Pliki ODBC

(położenie – katalog systemowy Windows)

48ODBCAD32.exe
49ODBC32.dll
50ODBCCP32.cpl
51ODBCCP32.dll
52ODDCINT.dll
53ODBCCR32.dll
54ODBC16GT.dll
55ODBC32GT.dll
56DS16GT.dll
57DS32GT.dll

Biblioteki Power++ v 2.5

(położenie - katalog bieżący programu)

58WCMDDA26.dll
59WCPDDA26.dll

Pliki HTML

(położenie – podkatalog \html\ katalogu głównego programu)

60Pliki o rozszerzeniu bmp
61Pliki o rozszerzeniu html

Pliki DataWindow v 6.0

(położenie – katalog systemowy Windows)

62Psdwe.ocx v 5.0.03.01
63Msvcirt.dll v 6.0.8186
64Psdwe1.lic
65Ivbp.lic
66Mfc40.dll v 4.1.0.6139
67Olepro32.dll v 5.0.4262.1

68Msvcert40.dll v 4.10.6038

69Msvcert.dll v 6.0.8186

70Msvcert.dll v 2.11.0.0

Inne pliki

(położenie – katalog systemowy Windows)

71Ea.ini

72CTL3D32.dll

73Ole32.dll

UWAGA:

Podczas instalacji programu EuroApteka (dokładnie podczas kopiowanie plików na dysk) może wystąpić problem rejestrowania plików o rozszerzeniu dll i ocx. Objawia się to wyświetleniem odpowiedniej informacji przez program instalacyjny. Należy wtedy po zainstalowaniu programu samemu zarejestrować w systemie pliki w stosunku do których został wyświetlony komunikat. W celu wykonania powyższego należy użyć programu znajdującego się w katalogu systemowy Windows o nazwie regsvr32.exe w katalogu systemowym.

Odeinstalowanie programu

Odeinstalowanie programu następuje przez uruchomienie opcji w Panelu Sterowania \Dodaj usuń programy\EuroApteka – odeinstaluj.

Odeinstalowanie programu powoduje:

74odeinstalowanie sterownika bazy danych SQL Anywhere 5.5 bez względu na to czy został on zainstalowany przez program instalacyjny czy nie (UWAGA. Inne programy korzystające z tego sterownika nie będą działać)

75pozostawienie na dysku struktury katalogów utworzonych przez program instalacyjny.

76pozostawienie na dysku plików bazy danych tj. EuroApteka.db i EuroApteka.log w katalogu Program Files\EuroSoft\EuroApteka\dbStart i dbDemo

77pozostawienie w katalogu Windows pliku konfiguracyjnego ea.ini

78pozostawianie wpisu konfiguracyjnego (rejestry systemowe) w ODBC tworzonego przy instalacji programu.

Ponowna instalacja programu

W przypadku, ponownej instalacji programu EuroApteka, program instalacyjny wyświetla komunikat informacyjny o potrzebie uaktualnianie plików bazy danych. W celu zapewnienia poprawnego działania programu należy wybrać opcję “Tak na wszystkie”.

Ustawienia ODBC

Baza startowa – EuroAptekaStart

Data Source Name: EuroAptekaStart

User ID: dba

Password: sql

Database Name: EuroApteka

Database File: ...\\EuroSoft\\EuroApteka\\dbStart\\EuroApteka.db

Database Startup: Custom

Options:

Start Command: ...\\Sybase\\SQL Anywhere 5.0\\win32\\dbeng50.exe

Baza demo – EuroAptekaDemo

Ustawienia j/w za wyjątkiem:

Data Source Name: EuroAptekaDemo

Database File: ...\\EuroSoft\\EuroApteka\\dbDemo\\EuroApteka.db

Konwersja bazy danych: InfoApteka->EuroApteka

Przeniesieniu podlegają dane o kontrahentach i lekach oraz bilans otwarcia magazynu

Eksport danych z InfoApteki:

79 Za pomocą programu dbexp.exe znajdującym się w katalogu f:\\farma\\sys należy wyeksportować dane z Farmacy3. Program uruchomić z katalogu f:\\farma\\system/ przez komendę:

```
f:\\farma\\sys\\dbexp -x farma3 company drg_rec cmp_rec oth_rec
```

company.txt – kontrahenci

drg_rec.txt – leki gotowe

cmp_rec.txt – substraty

oth_rec.txt – inne leki

80 Za pomocą programu magfm.exe (P:\\Farma3\\util\\mag_fm\\) stworzyć bilans otwarcia: magfm /1 /2 /3 bo.tfm (można generować pliki oddzielnie z opcjami /1)

/1 - leki gotowe

/2 - substraty

/3 – leki umowne

Wczytanie danych do bazy EuroApteki:

81Uruchomić program converter.exe znajdujący się w katalogu głównym programu EuroApteka (parametr: dsn <nazwa źródła danych>) i wskazać pliki company.txt itd. Zaznaczyć opcje Cena urzędowa za opakowanie fabryczne jeśli w pliku f:\farma\system\farma.ini wartość parametru CONST_PRICE_PER_PACK jest ustawiona na 1.

82Wczytać bilans otwarcia za pomocą programu EuroApteka (Magazyn->Dokumenty magazynowe->Odczyt) wskazując pliki zawierające bilans otwarcia wygenerowany z IA. Następnie zatwierdzić dokument. Należy pamiętać o wskazaniu magazynu docelowego na formularzu dokumentu.

Pliki konfiguracyjne *.ini

Pliki konfiguracyjne ini zawierają sekcje, sekcje zawierają pola, w których wpisywane są odpowiednie wartości.

Sekcje

Lista

Nazwa sekcji:

Form<Nazwa>s, np. dla przeglądarki klientów (tabela/akcja Customer) będzie FormCustomers

Pola:

SQL (wymagane) kwerenda, z której będą pobierane dane

Title (wymagane) tytuł okienka listy

Text (wymagane)

DataColumns (opcja) kolumny, które lista będzie wykorzystywała do szybkiego sortowania, lub filtrowania.

ListColumnN (opcja) kolumny, które będzie pokazywała lista. W przypadku, jeżeli nie jest podane, wykorzystuje się DataColumns

Default (opcja) filtr nałożony na listę; zob. dalej

Przykład:

```
[StyleName]
```

```
Font = 10.Arial.Bold.CharSet=238
```

```
[StyleFullName]
```

```
Font = 7.Arial.Italic.CharSet=238
```

```
[FormSponsors]
```

```
ListColumn0 = Kontrahent;name,StyleName;long_name,StyleFullName
```

```
ListColumn1 = Address1;name;+;address2;
```

```
DataColumns="1;2;3;4;5"
```

```
SQL="SELECT * FROM sponsor"
```

```
Title="Refundatorzy"
```

Text = "Ref&undatorzy"
Default = "id=1;name=abc"

Formularz

Nazwa sekcji:
Form<Nazwa>, np. FormCustomer

Pola:

Title (wymagane) tytuł okienka rekordu
User (opcja) nazwa pola zawierającego identyfikator użytkownika (user_id)
Status (opcja) nazwa pola zawierającego status dokumentu (status)
Date (opcja) nazwa pola zawierającego bieżącą datę (date)
ItemNumber (opcja) nazwa pola zawierającego numer pozycji dokumentu
Number (opcja) nazwa pola zawierającego numer dokumentu
DefaultFilter (opcja) domyślna kolumna do szybkiego wyszukiwania
DefaultSort (opcja) domyślna kolumna do sortowania

Przykład:

```
[FormInvoice]
Title = "Faktura"
User = "user_id"
Status = "status"
Date = "makeout_date;delivery_date"
Number = "id"
DefaultFilter = "contrator_id"
DefaultSort = "3"
```

DataColumns

Pole zawiera identyfikatory kolumn oddzielone średnikami.

W DataColumns powinny występować wszystkie pola wykorzystane w ListColumn.. Jeżeli nie ma sekcji ListColumn to lista zawiera pola wymienione w DataColumns. Jeżeli nie ma

DataColumns to lista zawiera wszystkie pola tabel wymienione w SQL.

Identyfikator kolumny to liczba całkowita dodatnia lub nazwa kolumny bez nazwy tabeli lub nazwa tabeli i nazwa kolumny oddzielone kropką:

```
POLE_DATACOLUMN ::= DataColumn = "<KOLUMNA>{;<KOLUMNA>}"
KOLUMNA ::=      <LICZBA_DODATNIA>
                  | <NAZWA_KOLUMNY>
                  | <NAZWA_TABELI>.<NAZWA_KOLUMNY>
```

UWAGA: Nazwa tabeli NIE MOŻE być aliasem użytym w kwerendzie.

Przykład:

```
DataColumns = "1;2;customer_id;invoice.name;4;customer.name"
```

SQL

WAŻNE: Kwerenda nie może kończyć się średnikiem!

SELECT

W klauzuli SELECT można używać * i nazw kolumn.

Nazwy kolumn można kwalifikować nazwami tabel, także aliasami.

Nie można natomiast używać zapisu <tabela>.*

FROM

W klauzuli FROM można używać aliasów oraz KEY/NATURAL JOIN.

Główna tabela musi występować jako pierwsza.

WHERE

Można używać dowolnych warunków.

W kwerendach dla details pytajnik używany jako parametr do łączenia z masterem musi występować na samym końcu.

Przykład:

```
SQL="SELECT * FROM invoice_item I KEY LEFT OUTER JOIN physical_drug D WHERE  
D.physical_drug_id>=0 AND I.invoice_id=?"
```

User, Date, Status

Pole User zawiera nazwę kolumny, która opisuje klucz obcy do tabeli "user_account". Operacja Nowy umieszcza w tej kolumnie identyfikator bieżącego użytkownika.

Jeżeli kolumna opisująca użytkownika nazywa się "user_id" to nie trzeba jej podawać w pliku ini.

Pole Date zawiera nazwy kolumn, oddzielone średnikami, które powinny zawierać bieżącą datę. Jeżeli kolumna opisująca datę nazywa się "date" to nie trzeba jej podawać w pliku ini.

Pole Status zawiera nazwę kolumny, która opisuje status dokumentu. Operacja Zatwierdź umieszcza w tej kolumnie "Z".

Jeżeli kolumna opisująca status nazywa się "status" to nie trzeba jej podawać w pliku ini.

Identyfikator kolumny to liczba całkowita dodatnia lub nazwa kolumny bez nazwy tabeli lub nazwa tabeli i nazwa kolumny oddzielone kropką:

Default

Pole Default zawiera formułę do filtrowania rekordów. Formuła ta to niepusty ciąg napisów postaci <kolumna>=<wartość>, oddzielonych średnikami, np.

```
Default = "group_id=2;member_of=13"
```

Większy przykład

Dla typowego master/detail potrzeba 4 sekcji. Rozważmy to na przykładzie dokumentu faktury.

1. Najpierw tworzymy sekcję dla listy faktur, czyli:

```
//----- lista dokumentów
[FormInvoices]
SQL="SELECT * FROM invoice"
Text = "&Faktury"
Title = "Faktury"
// Ten default nie ma sensu, to tylko przykład
Default = "invoice_id=1"
```

2. Potem tworzymy dwie sekcje dla master/detail:

```
//----- formularz mastera
[FormInvoice]
Title = "Faktura"
Number = "id"
User = "user_id"
Status = "status"
Date = "makeout_date;delivery_date"
Number = "id"
DefaultFilter = "2"
DefaultSort = "3"
// albo...
DefaultFilter = "customer_id"
DefaultSort = "drug_id"
```

//----- lista pozycji, detail

```
[FormInvoiceItems]
DataColumns="1;2;invoice_item.physical_drug_id;physical_drug.physical_drug_id"
SQL="SELECT * FROM invoice_item I KEY LEFT OUTER JOIN physical_drug D WHERE
D.physical_drug_id>=0 AND I.invoice_id=?"
Text="&Pozycje Faktury"
Title="Pozycje Faktury"
```

3. Następnie tworzymy sekcję dla pojedynczej pozycji faktury:

```
//----- formularz pozycji dokumentu
[FormInvoiceItem]
Title="Pozycja Faktury"
ItemNumber="item_number"
```

Nazwa pliku

Każda aplikacja powinna mieć swój plik ini,

Przykład:

EuroApteka EA.ini.

Najlepiej aby plik był w katalogu, z którego uruchamiana jest aplikacja.

Przykład:

Dla ea jest katalog bieżący aplikacji.

Nazwę pliku konfiguracyjnego określa w kodzie programu:

```
IAApplication::SetProfileFileName()
```

Przykład:

```
static const WString ProfileFileName=".\\EA.ini";  
    // kropka oznacza bieżący katalog.  
  
_application.SetProfileFileName( ProfileFileName );
```

patrz również IAAction\Test\FormMain

ListColumnN

N = 0, 1, 2,

ListColumnN = NazwaKolumny ';' LisColumnDef

ListColumnDef = NazwaPolaBD ';' SekcjaStylu ';' ListColumnDef

| NazwaPolaBD ';' SekcjaStylu

| NazwaPolaBD

(*) | ListColumnDef '+'; ListColumnDef

Jeżeli w ListColumnN podane kilka pól z bazy danych, to ich wartości są wyświetlane jedno pod drugim, chyba że jest '+' (przypadek *) . Wtedy pole po plusie jest wyświetlane z prawej strony w tym samym wierszu co pole przed plusem.

Kopiowanie dokumentów

Możliwe jest r tworzenie dokumentów na podstawie treści dokumentów pochodzących z innych rejestrów.

Podczas kopiowania, przepisuje się tylko część pól określona w pliku konfiguracyjnym ini.

CopyMaster: kolumny docelowe główki

CopyDetail: kolumny docelowe pozycji

CopyMasterSource: kolumny źródłowe główki

CopyDetailSource: kolumny źródłowe pozycji

CopySQLSource: pytanie SQL dla źródła

gdzie Source nazwa listy z danymi źródłowymi
Kolejność kolumn ma znaczenie, treść jest kopiowana z kolumny źródła do odpowiadającej kolumnie docelowej. Jeżeli w jakimś źródle nie ma kolumny, która występuje w kolumnach docelowych, należy w źródle wpisać nazwę kolumny nieistniejącej np "no_data"

Dla przykładu fragment sekcji opisującej kopiowanie do zamówienia sprzedaży

```
[FormSaleRequest]
```

```
..
```

```
CopyMaster="drawer_id;contractor_id;note_id"
```

```
CopyDetail="physical_drug_id;quantity;price_netto;price_brutto;value_netto;value_brutto;value_vat;vat_id"
```

```
CopyMasterPurchaseInvoice="seller_id;purchaser_id;note_id"
```

```
CopyDetailPurchaseInvoice="physical_drug_id;quantity;price_netto;price_brutto;netto;brutto;vat_vat_id"
```

```
CopySQLPurchaseInvoice="SELECT * FROM invoice KEY JOIN invoice_item"
```

Inne pola występujące w pliku ini

Zerowanie numeru dokumentu

```
NumberPeriod =[Year, Month, Day]
```

Przykład:

```
[FormReceipt]
```

```
..
```

```
NumberPeriod = Day
```

Liczba numerków

Przykład:

```
[FormExpedManuOrders]
```

```
..
```

```
PrintNumberCount = 2
```

Opłata za dyżur nocny

Przykład:

```
[FormExpedInvoiceItem]
```

```
ExtraPayment = -1 [ physical_drug.physical_drug_id ]
```

Ostrzeżenie przed końcem daty ważności

```
[FormExpedSelectStoreLineRps]
```

```
LimitValidDays = <całkowita liczba>
```

domyślnie 182 (pół roku)

Sekcje Stylu – plik “iastyle.ini”

Style definiowane są w pliku iaStyle.ini

Wszystkie pola w sekcjach stylu są opcjonalne. W przypadku, gdy nie jest podane pole, jest ustawiany styl domyślny.

Pola

Font = Definicja czcionki

Alignment = 'left' | 'right' | 'center'

Width = 'fixed' | liczba

Style = 'number' | 'label' | 'switch'

IAAction ver 01.28 (22-09-1999)

Format = definicja formatu (tylko dla Style = number)

Label = definicja tekstu (tylko dla Style= label)

W przypadku gdy pole Width ma wartość 'fixed', szerokość kolumny ustawia się tylko raz. Zmiana filtra lub sortowania nie modyfikuje szerokości kolumny.

Style = number

Pole Style o wartości 'number' wyświetla tekst kolumny tak jak liczbę (na przykład 123456789.09 będzie wyświetlone w postaci 123 456 789,09).

Pole Format

Pole Format jest uwzględniane tylko dla Style = 'Number'.

Domyślną wartością pola jest "%.2f". Składnia języka C

Style = label

Style 'label' określa stały napis.

Pole Label

Pole Label jest uwzględniane tylko dla Style = 'Label'.

Wartością tego pola jest stały tekst wyświetlany dla każdego wiersza.

Na przykład:

Style = switch

Style 'switch' zamienia napis z zapytania na inny napis.

Pola CaseN, ValueN, gdzie N - numer kolejnego zamiennika numerowany od 0

Wartością pola CaseN jest napis, który może powstać w wyniku zapytania

Wartością pola ValueN jest napis, na który trzeba zamienić napis z CaseN.

Zamówienia ze stanów magazynowych

Na zamówieniu sporządzonym na podstawie stanów magazynowych znajdują się towary:

a) znajdujące się w aktualnym widoku listy stanów magazynowych

b) o ilości mniejszej niż deklarowane minimum

Zamówiona zostanie taka ilość, by poziom towaru był w połowie między min i max. Ilość towaru jest zamawiany w opakowaniach całkowitych.

Przykład:

Stan mag. = 20; zarez.=5, min=21, max=70, rozm. opak. =7

Rzeczywisty stan=20-5=15 pastylek.

W magazynie powinno się znaleźć $\min + ((\max - \min) / 2)$, czyli $21 + 24 = 45$ pastylek. Trzeba zamówić 30 pastylek. Zamawia się 35 pastylek (5 pełnych opakowań).

Obliczanie ceny detalicznej

N - cena zakupu NETTO

B - cena zakupu BRUTTO

V - stawka VAT

FM - funkcja licząca cenę z marżą na podstawie podanej ceny

DN - cena detaliczna NETTO

DB - cena detaliczna BRUTTO

Z - podana cena zakupu (brutto lub netto)

if (Z.brutto)

```
{  
    N = Z / ( 1 + V );  
    B = Z;
```

```
}  
else
```

```
{  
    N = Z;  
    B = Z * ( 1 + V );
```

```
}
```

M – marża (liczona od ceny brutto lub netto)

if (M.brutto)

```
{  
    DB = FM(B);  
    DN = DB / ( 1 + V );
```

```
}  
else
```

```
{  
    DN = FM(N);  
    DB = DN * ( 1 + V );
```

```
}
```

Opis algorytmu funkcji obliczającej marżę FM:

Marża może być liczona od góry (marże procentowe) lub od dołu (marża degresywna).

```
if ( M.liczOdDoŁu )  
    linear = 1 + linearCoof;  
else  
    linear = 1 / ( 1 - linearCoof );
```

$CM = FM(CBM) = CBM * linear + constCoof$

Gdzie:

CM - cena z marżą

CBM - cena bez marży

LinearCoof – współczynnik liniowy marży

ConstCoof – współczynnik stały marży (marża kwotowa)