Двустранно счетоводство с MS Access

Ползвате ли Access?

"Ами малко..." или "Да, но предпочитам Excel" – това са най-често срещаните отговори, които дават потребителите на MS Office, дори и когато имат дългогодишен опит.

***Евгения Христова** (София)*
info@shevitza.com

Интерфейсът на MS Access или защо въпреки всичко ми е трудно

MS Access е програма с невероятен интерфейс, но трябва човек да се сблъска с други приложения за база данни (например MySQL в един класически хостинг-пакет например!) за да го оцени. Точно този интерфейс, като че ли, създава у потребителя усещане, че с него се прави всичко, както в Word. И ако нещо не става точно така, той е склонен да се откаже. А броят на задачите в последните години расте. И все повече служители на фирми се чудят дали да се захванат сами с възникналия труден проблем или да поискат някой да им конструира приложение на Access!

След като човек се е запознал с правилата за създаване на таблици, заявки, форми и отчети и решава да се заеме с нещо по-близко до практиката, често установява, че всъщност не може да реши даже и проста наглед задача, като отчетност на малък склад или счетоводство с пет сметки. Проблемът много често идва от това, че с едно въвеждане искаме да генерираме записи на няколко места. С нанасянето на данните за новопостъпила стока, искаме да изчислим среднопретеглената цена и да я нанесем в таблицата за наличностите. Или с изписването на стоката по метода FIFO или LIFO искаме многократно да променяме наличността, в зависимост от изписваното количество и броя доставки.

Задачата на двустранното счетоводство, която от столетия се е решавала на ръка и едва в последните двадесетина години – с компютър, също може да бъде разгледана като пример за несложна задача от групата на споменатите по-горе.

Моделът на двустранното счетоводство

За всички, които не са счетоводители или икономисти ще кажем, че това е един невероятно прост и красив модел. И така, разполагаме със сметкоплан, в който са записани номерата на сметките и техните имена. Всяка счетоводна операция се реализира, като се дебитират и кредитират две различни сметки с една и съща сума. Ако отида до банкомата и изтегля пари, това се изразява с дебитиране на сметката на касата, (в случая портмонето ☺) и кредитиране на банковата сметка. А когато плащам стоката в супера, дебитирам склада (или хладилника) като кредитирам касата (или портмонето).

Въвеждането на данните трябва да стане еднократно, като запишем сметките за дебит, кредит, сумата, номера и дата на документа и описание в един запис в таблицата на операциите, а да получим като резултат и още два записа в друга таблица – първия с номера на операцията, сметка за дебит и сумата за дебит, и втория – със същия номер на операцията, но със сметка за кредит и същата сума. Възможно е да се създадат няколко заявки, които да вземат последния запис от таблицата с операциите и с помощта на добавяне на записи и ъпдейт на празни полета до 0, да доведат до резултат. Наличието на много заявки, в групата на заявките, плаши потребителя, когато той самия иска да създаде заявка за някакъв конкретен, частен случай. Е, може да се зададат правила за имената, може за всяка заявка да се добави описание в свойствата, изпълняваните от макроси заявки може да се преобразуват в код на VB, но всичко все пак изисква грижа, внимание и четене на "указанията за употреба".

А начинът на осчетоводяване си остава един и същ, неизменен, независимо какви допълнителни задачи ще решава потребителя в своята база.

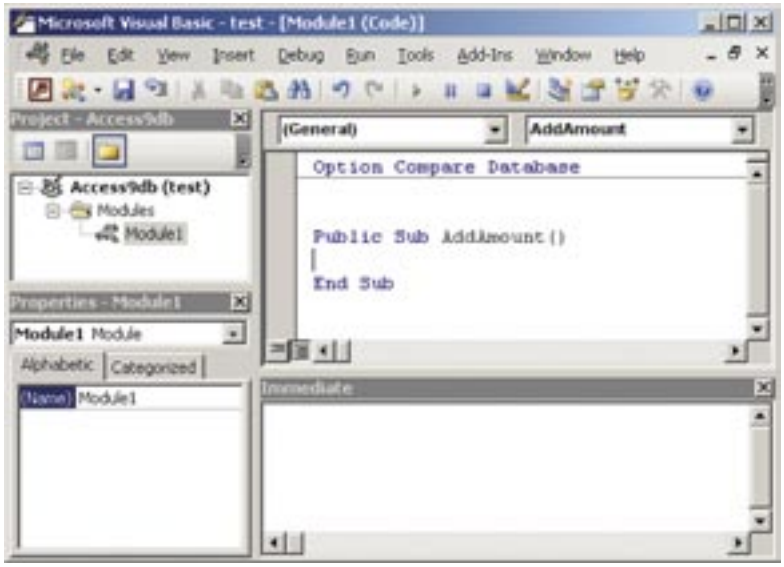
Тук ще решим задачата с модела на двустранното счетоводство, като изнесем "осчетоводяването" и "сторното" в код, извикван от формите, където се въвеждат данните. Ще се стараем да съблюдаваме правилото за "НЕпреследване на заека с оръдие!" и ще работим в рамките на един единствен, обикновен .mdb файл, без да мислим за другите вариантите, като "клиент – сървър" например.

Как да пишем в таблиците, без да ги отваряме директно или през форма?

Да, има такава "магия" и тя се прави с метода OpenRecordset() на обекта Database, след което се обикалят полетата на записа и се прилага метода AddNew – добави нов. Нашата база данни е обект от типа Database, а записите в нея се виждат и манипулират с помощта на подобектите Recordset – record е запис, а set– поставям. Метод на един обект е някакво действие, в случая OpenRecordset отваря местата на записите. След като сме отворили нещо, не бива да забравяме да го затворим обаче! Един обект от типа Recordset има и метод Close - затваряне.

Създайте една проста база например с името test.mdb. В нея създайте таблица под името tblOperazia, с поле за Primary Key- OperaziaID от типа Auto Number и още едно поле под името Amount – в него ще нанасяме сумата на операцията.

Преминете в групата на модулите и създайте нов модул с бутона New от прозореца Database. Ще попаднете в редактора на Visual Basic. Извикайте Insert – Procedure и създайте процедура под името AddAmount().



Фиг. 1

Дименсиониране на променливите

Трябва ни променлива за база данни.

```
Dim db As Database
```

Нуждаем се от променлива за достъп до записите

```
Dim rstOperazia As Recordset
```

Amount е променливата, в която ще записваме данните за операцията или в случая - парите.

```
Dim Amount As Currency
```

Забелязваме, че още щом напишем As редакторът на VB се опитва да ни съдейства, като вади падащ списък за избор на типа – значи сме на прав път!

Задаване на стойности на променливите

Променливите, които са обекти като базата данни и рекордсета, ще получат стойности с думата Set, а останалите – просто с равно.

И така, със следващия ред казваме, че базата данни всъщност е текущата база данни. Но като нищо, можем да отворим и някоя друга, стига да знаем пътя до нея ;-)

```
Set db = CurrentDb()
```

След това насочваме рекордсета към таблицата с операциите.

```
Set rstOperazia =
db.OpenRecordset("tblOp
erazia")
```

Решили сме да действаме със 100 лева и обявяваме това така:

```
Amount = 100
```

Манипулиране на рекордсета

Това става като зададем начало на действието с:

```
With rstOperazia
```
'с рекордсета (прави...)

И обявим, че приключваме действието с:

```
End With
```
' край на "правенето"

Апострофът предшества коментар.

```
With rstOperazia
```
'с рекордсета (прави...)

```
.AddNew
```

```
!Amount = Amount
```
'обръщение към полето на таблицата

```
.Update
```
'ако не кажем "обнови" не може!

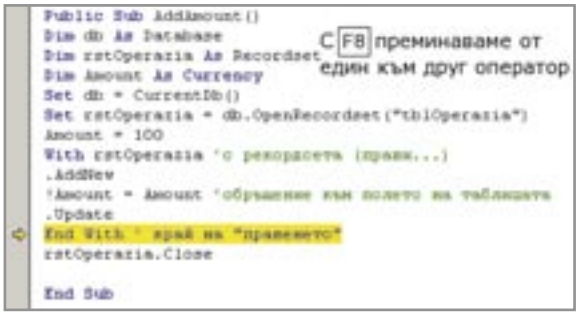
```
End With
```
' край на "правенето"

И да не забравим да затворим вратата:

```
rstOperazia.Close
```

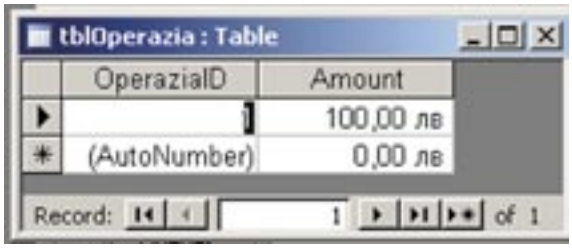
Я да видим как работи!

Позиционирайте се в началото на процедурата и натискайте клавиша F8 – така ще преминавате плавно от оператор към оператор.



Фиг. 2

След като сте преминали през кода така до края на процедурата, отворете си таблицата – ще видите това:



Фиг. 3

Ами да, "магията хвана" и добавихме в таблицата запис без да я отваряме!

Задайте на променливата друга стойност – например 200 и отново обходете с F8. Ще добавите запис с номер 2 и сума – 200 лв.

Обхождане на рекордсета

Обхождането става отново с With End With, но вътре има един оператор от следния вид:

```
Do While Not .EOF
```
'До като не стигнеш края на файла

```
Do While Not .EOF
```
' печати полета 0 и 1

```
.MoveNext
```
' премести се на следващото

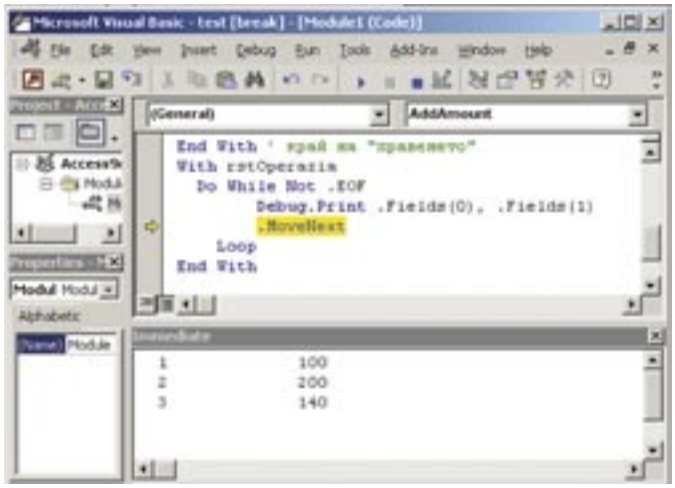
```
Loop
```
'Завий обратно или направи лупинг

```
Debug.Print
```
ще разпечата в прозореца Immediate съдържанието на полетата 0 и 1

Обхождането ще поставим преди да сме затворили рекордсета, естествено!

Прозорецът Immediate може да покажете от командата View, ако не го виждате.

Резултатът е по-долу.



Фиг. 4

Как "да вземем" стойност от рекордсета

Да, с ***.AddNew*** сложихме нещо в рекордеста, а сега искаме да вземем номера на операцията, който се генерира автоматично от полето, което е първичен ключ. Нали искаме този номер на операцията, а не всички допълнителни данни за номер и дата на документа и описанието да запишем в таблицата с двойните редове – ще я наречем ***tblTrans***

Необходима ни е "бележка" – ***Bookmark*** за мястото, от където ще вземем стойност.

.Bookmark=.LastModified ще ни установи на последния добавен запис. Ще вземем номера на операцията ***OperaialD*** в променливата ***OperaziaNom***. Обхождането ще мине само през последния запис защото там сме фиксирали "бележката"

Ето и резултатът:

```
Amount = 160
Dim OperaziaNom As Long
With rstOperazia 'с рекордсета (прави...)
.AddNew
!Amount = Amount 'обръщение към полето на табл
.Update
.Bookmark = .LastModified
OperaziaNom = !OperaziaID
Debug.Print OperaziaNom
End With ' край на "правенето"
With rstOperazia
  Do While Not .EOF
        Debug.Print .Fields(0), .Fields(1)
        .MoveNext
    Loop
End With
rstOperazia.Close
End Sub
```

```
Immediate
 6
 6              160
```

Фиг. 5

Вече имаме всичко необходимо за да продължим с реалната задача защото:

· можем да добавим данни в таблица без да я отваряме;

· следователно ще можем да добавим едни и същи или малко променени данни едновременно в две таблици;

· можем да вземем последната генерирана данна от едната таблица и да я запишем в другата;

· имаме контрол на резултатите през редактора на VB с оператора Debug.Print.

Таблици в базата за двустранното счетоводство

Те са характерни с това, че в тях почти никога не се пише директно – освен в сметкоплана и при задаване на началните салда.

· ***tblOperazia*** – съдържа номера на операцията, сума, сметка за дебит, сметка за кредит, вид и номер на документа, дата, описание;

· ***tblTrans*** – съдържа поле за номера на операциите, поле за номер на сметка, поле за дебит, поле за кредит, като едното или дебита, или кредита е винаги 0;

· ***tblSmetkoplan*** – съдържа поле за номера на сметката, което е индесирано и поле за името на сметката

· ***tblNachalnoSaldo*** – съдържа номера на всяка сметка, поле за дебитно салдо, поле за кредитно салдо. Данните за началното салдо ще съответстват на операция с номер 0.

Заявки в базата за счетоводството

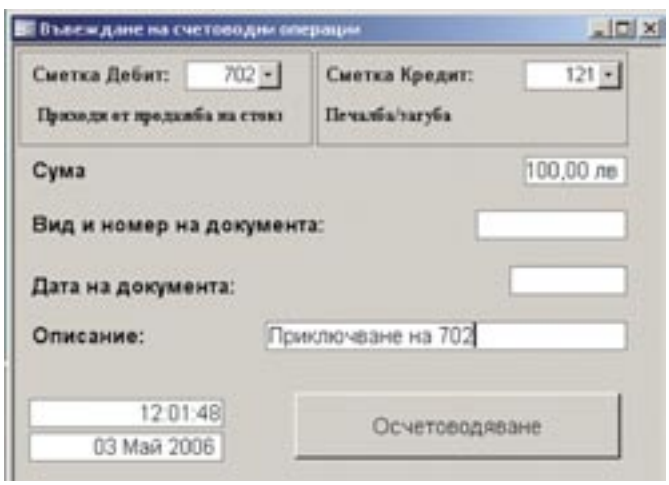
· ***qryOborotnaVedomost*** – заявка за оборотната ведомост

· ***Query1*** – скрита заявка, която се стартира през макро и добавя началните салда към таблицата tblTrans

Форми

Форма за нанасяне на счетоводните операции

Това е frm_f1. Получава се като се създаде празна форма, несвързана със никаква таблица или заявка и лишена от селектори за записи и навигационни бутони. Върху тази форма има две контроли ComboBox за избор на номера на сметката и контроли от типа текстово поле за нанасяне на сумата на операцията, вида и номера на документа, датата, описанието на операцията и бутон за осчетводяване. Този бутон задейства код, който записва данните за операцията в таблиците tblOperazia и tblTrans - именно с особеностите на този код се занимавахме в началото!



Фиг. 6

Следва код за нанасяне на данните в двете таблици:

```
With rstOperazia
   .AddNew
   !SmetkaDebit = intSmetkaDebit
   !SmetkaCredit = intSmetkaCredit
   !Amount = curAmount
   !DocNumber = strDocNum
   !DocDate = dDocDate
   !Description = strDescription
   .Update
   .Bookmark = .LastModified
   intOperaziaID = !OperaziaID
   Debug.Print intOperaziaID
End With

'Update debit in tblTrans
With rstTrans
   .AddNew
   !Account = intSmetkaDebit
   !Debit = curAmount
   !Credit = 0
   !OperaziaN = intOperaziaID
   .Update
End With
'End of update Dåbit

'UpdateCredit in tblTrans
With rstTrans
   .AddNew
   !Account = intSmetkaCredit
   !Debit = 0
   !Credit = curAmount
   !OperaziaN = intOperaziaID
   .Update
End With
'End Of Update Credit
```

А по-долу имената на полетата на самите таблици:

Фиг. 7

Форма за сторно

Тя съдържа контрола ComboBox за избор на операция от таблицата на операциите tblOperazia, етикет чиито Caption съдържа всички данни на операцията, която ще сторнираме и бутон за сторно. Бутонът задейства код, който намира операцията, която потребителят иска да сторнира, показва нейните данни върху етикета, показва кутия за съобщения, с която пита потребителя дали иска да сторнира точно тази операция и при отговор Yes извършва сторното като счетоводна операция с отрицателна стойност на сторнираната. За описание на операцията се записва "Сторно" и номера на сторнираната операция.

Форма Switchboard

Тя се генерира по обичайния за Access начин, като съдържа бутони за стартиране на формите за счетоводни операции и сторно, оборотната ведомост и списъка на операциите като отчети.

Краят на приключението

Струва ми се също, че времето на готовите проекти на Access, които потребителят само ползва, без да смее да ги пипне отвътре, отминава. Във няколко форума срещнах изказвания, че Access не бил подходящ за правене на приложения по поръчка, защото като нарастне броя на записите, работел лошо. Ами всички програми за база данни почват да работят лошо, когато им се налага "да търкалят" прекалено много записи. В един момент, на края на тримесечието или годината ще трябва да освободите таблиците от старите данни, за да ги пълните с нови. Архивирането на данните също може да се автоматизира, но нали си обещахме нещо за зайците и оръдията! Копирате файла в друга папка, триете записите, нанасяте нови начални салда, стартирате макро Dobavia_Nachalni_Salda и всичко е готово. Импортирането и експортирането на данни, когато е строго зададено от разработчика, обикновено създава проблеми на потребителя, по-лесно е последния да си го прави сам. И както и за всяка друга работа с компютър и за тази ще кажем : **АРХИВИРАЙТЕ, АРХИВИРАЙТЕ, АРХИВИРАЙТЕ!**

Въпреки, че ако сте счетоводител, вероятно си имате програма, всичко казано дотук показва, че нещата с базите данни не са толкова страшни. Надявам се, че сега вие ще имате смелост да решавате и други подобни задачи.