



Programmeren in Microsoft Visual Basic 6.0, lessenserie voor het voortgezet onderwijs HAVO/VWO

© David Lans, Emmauscollege, Marnix Gymnasium
Rotterdam, februari 2001

Hoofdstuk 2: Werken met variabelen

2.0 Leerdoel

In dit tweede hoofdstuk van de cursus “Programmeren in Visual Basic” leer je:

- kennis maken met de volgende onderdelen van een Visual Basic project:
 - meerdere formulieren in één project (Add Form)
 - afbeeldingsvelden (PictureBoxes)
- nieuwe gebeurtenissen in een project beschrijven met programmacode, zoals:
 - een formulier openen (Show)
 - tekst wegschrijven in een afbeeldingsveld (op een ander formulier)
 - het opschrift van een formulier veranderen (Caption en Icon)
 - weggeschreven tekst in een afbeeldingsveld opmaken (“&”, “,”, BackColor en Format)
- enkele algemene regels voor het programmeren
 - werken met de typen variabelen Single, Integer en Currency
 - herhalingsopdrachten beschrijven met
 - For ... Next opdrachten en
 - Do ... Loop opdrachten











We doen dat aan de hand van een voorbeeld:

A. De spaarrekening

Met dit hoofdstuk ben je, afhankelijk van je tempo, ongeveer 2 lessen van 50 minuten bezig.

2.1 Kort overzicht van mogelijke objecten in Visual Basic

In hoofdstuk 1 heb je kennis gemaakt met enkele objecten of besturingselementen die je in je Visual Basic projecten kunt opnemen. Het volgende overzicht geeft de functie van een wat uitgebreidere verzameling objecten.

| Symbool | Object | Functie |
|---|---------------------------------------|--|
|  | Knop (<i>CommandButton</i>) | Een gebeurtenis in werking zetten. |
|  | Label (<i>Label</i>) | Een ander object van commentaar voorzien |
|  | Tekstvak (<i>TextBox</i>) | Invoer en uitvoer van tekst |
|  | Afbeeldingsveld (<i>PictureBox</i>) | Weergave van afbeeldingen, tekst en grafieken. |
|  | Afbeelding (<i>ImageBox</i>) | Weergave van afbeeldingen |
|  | Keuzerondje (<i>OptionButton</i>) | Keuze uit opties die elkaar uitsluiten |
|  | Selectievakje (<i>CheckBox</i>) | Keuze uit opties, meerdere tegelijk mogelijk |
|  | Keuzelijst (<i>ListBox</i>) | Keuze uit een lijst met tekstelementen |
|  | Keuze/invoerlijst (<i>ComboBox</i>) | Combinatie van een invoervak en een keuzelijst |
|  | Frame (<i>Frame</i>) | Groepering van objecten zoals keuzerondjes |

Tabel 2.1 Overzicht van Visual Basic objecten

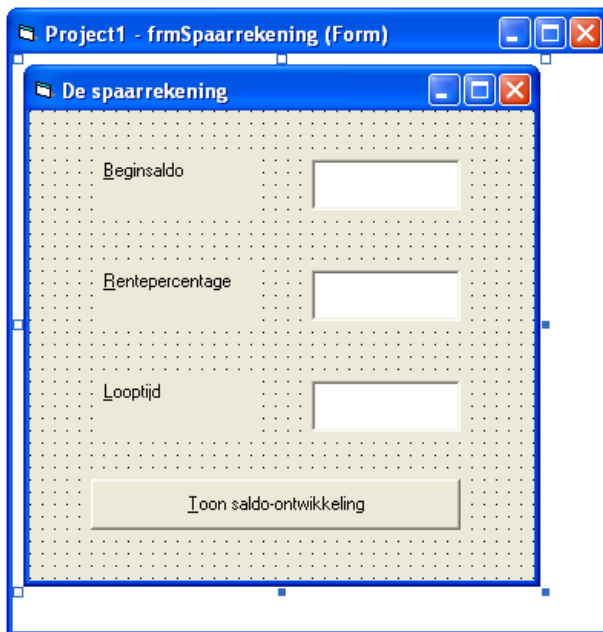
Het overzicht maakt duidelijk waar je de diverse objecten voor gebruikt. We combineren in deze cursus het uitbreiden van onze kennis van omgaan met objecten in Visual Basic met het uitbreiden van onze kennis van programmeertechnieken. In dit hoofdstuk staan centraal:

- het omgaan met variabelen
- het object afbeeldingsveld
- het programmeren van herhalingen

We doen dat aan de hand van een volgend voorbeeld: “De spaarrekening”.

2.2 De gebruikersinterface van “De spaarrekening”

In het project “De spaarrekening” willen we volgen hoe een spaarbedrag groeit tegen een gegeven rentepercentage over een opgegeven looptijd. Door op een knop te drukken krijgt de gebruiker een nieuw formulier voor zich met de saldo-ontwikkeling over de opgegeven looptijd.



☞ Geef het formulier de volgende eigenschappen ([Properties]):

| | | |
|-----------|-------------|------------------|
| naam | [(Name)] | frmSpaarrekening |
| opschrift | [Caption] | De spaarrekening |

We plaatsen nu de tekstvelden en labels op de plaatsen waar de gebruiker Beginsaldo, Rentepercentage en Looptijd in moet gaan voeren:

☞ Plaats een label ([Label]) uit de werkbalk ([General]) op het formulier.
Geef het label de volgende eigenschappen ([Properties]):

| | | |
|-----------|-------------|---------------|
| Naam | [(Name)] | lblBeginsaldo |
| Opschrift | [Caption] | &Beginsaldo |

Welk effect heeft het plaatsen van het teken “&” voor de tekst “Beginsaldo”?

☞ Plaats een tekstveld ([TextBox]) uit de werkbalk ([General]) op het formulier.
Geef het tekstveld de volgende eigenschappen ([Properties]):

| | | | |
|-------|----------|---------------|-------------|
| Naam | [(Name)] | txtBeginsaldo | |
| Tekst | [Text] | | (niets dus) |

De labels en tekstvelden van Rentepercentage en Looptijd maak je op vergelijkbare wijze.

Vervolgens plaatsen we een knop op het formulier:

- Plaats de knop ([CommandButton]) uit de werkbalk ([General]) op het formulier. Geef de knop de volgende eigenschappen ([Properties]):

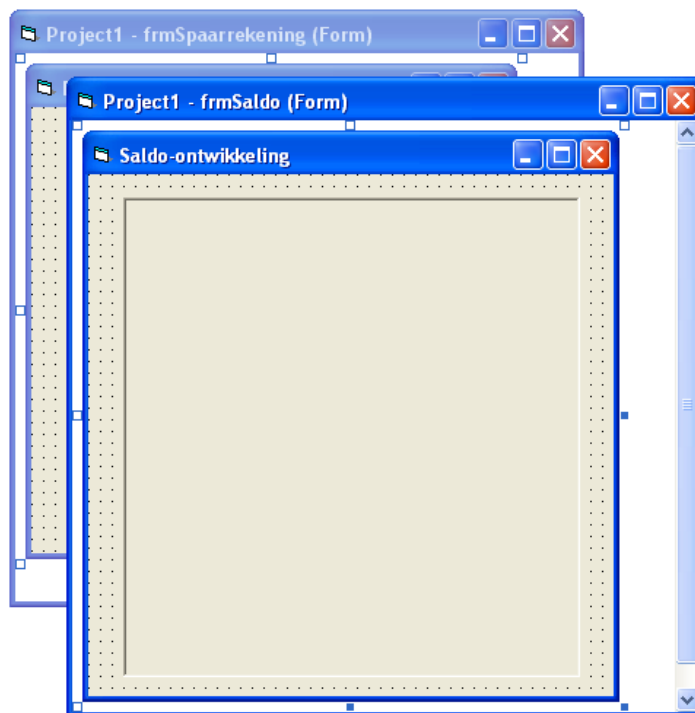
| | | |
|-----------|-------------|--------------------------|
| Naam | [(Name)] | cmdToonsaldo |
| Opschrift | [Caption] | &Toon saldo-ontwikkeling |

We voegen vervolgens een tweede formulier aan het project toe.

- Kies of voor Project / Add Form of voor het icoon  Geef het formulier de volgende eigenschappen ([Properties]):

| | | |
|-----------|-------------|--------------------|
| Naam | [(Name)] | frmSaldo |
| Opschrift | [Caption] | Saldo-ontwikkeling |

Zowel in het projectvenster als in het formuliervenster zie je dat het project nu uit twee formulieren bestaat.



We plaatsen nu op het formulier een afbeeldingsveld waarin de saldo-ontwikkeling kan worden weggeschreven.

- Plaats het afbeeldingsveld ([PictureBox]) uit de werkbalk ([General]) op het formulier. Geef het afbeeldingsveld de volgende eigenschappen ([Properties]):

| | | |
|------|----------|----------|
| Naam | [(Name)] | pctSaldo |
|------|----------|----------|

2.3 De programmastructuur van “De spaarrekening”

We staan nu weer voor een programmeerprobleem. Alle objecten die we nodig hebben zijn weer gemaakt. Aan de knop cmdToonsaldo zal echter weer de nodige programmacode moeten worden toegevoegd. Voordat we deze code toevoegen denken we eerst weer even na over de programmastructuur:

| |
|---|
| Lees Beginsaldo |
| Lees Rentepercentage |
| Lees Looptijd |
| Saldo = Beginsaldo |
| Voor Jaar = 1 ... Looptijd |
| Saldo = Saldo * (1 + Rentepercentage / 100) |
| Schrijf Jaar, Saldo |
| Volgende Jaar |

Elk jaar wordt het saldo weer met een zelfde factor vermenigvuldigd. Bijvoorbeeld, bij een rentepercentage van 5 % is die factor 1,05. Telkens dient het berekende saldo te worden uitgevoerd. Omdat telkens dezelfde bewerking herhaald dient te worden (oude saldo x 1,05 en weer afdrukken) kan hier handig gebruik gemaakt worden van een herhalingslus.

2.4 De variabelen in “De spaarrekening”.

De getallen die in de programmacode van “De spaarrekening” een rol spelen zijn getallen van verschillende typen:

| | | |
|-----------------|--------------|------------------------------------|
| Beginsaldo | geldbedrag | twee cijfers achter de komma |
| Rentepercentage | in procenten | meestal één cijfer achter de komma |
| Looptijd | in jaren | geheel getal |
| Jaar | | geheel getal |

Het gegevenstype is bij het programmeren erg belangrijk. Een gebruiker dient bij een foute invoer (bijvoorbeeld een getal met 5 cijfers achter de komma waar een geldbedrag verwacht wordt of bijvoorbeeld tekst waar een getal verwacht wordt) daarvan melding te krijgen. Ook is het niet zo fraai (zoals bij het project “De euro omzetter” het geval was) om als uitvoer een geldbedrag met 10 cijfers achter de komma te krijgen.

Een andere reden om belang te hechten aan het gegevenstype van een variabele is dat er met de verwerking van het gegevenstype ook geheugenruimte gemoeid gaat. Om astronomisch grote of microscopisch kleine getallen te verwerken is aanzienlijk meer geheugenruimte nodig dan om relatief simpele getallen te verwerken.

Er zijn natuurlijk nog heel veel andere gegevenstypen mogelijk die kunnen worden ingevoerd en uitgevoerd. Een beschrijving van mogelijke gegevenstypen in Visual Basic vind je op de volgende pagina.

| Gegevenstype | Declaratieteken | Bytes | Waardebereik |
|--------------|-----------------|----------|---|
| Boolean | | 2 | logische waarden: 0 (<i>False</i>) of -1 (<i>True</i>) |
| Byte | | 1 | gehele getallen: 0 ... 255 |
| Integer | % | 2 | gehele getallen: - 32768 ... 32768 |
| Long | & | 4 | gehele getallen: - 2 147 483 648 ... 2 147 483 648 |
| Single | ! | 4 | wetenschappelijk getallen, kleine kommagetallen: - 3,402823 E38 ... - 1,401298 E-45 en 1,401298 E-45 ... 3,402823 E38 |
| Double | # | 8 | wetenschappelijke getallen: - 1,79769313486232 E308 ... -4,94065645841247 E-324 4,94065645841247 E-324 ... 1,79769313486232 E308 |
| Decimal | | 12 | decimale getallen, grote kommagetallen: - 79 228 162 514 264 337 593 543 950 335 ... 79 228 etc. - 7,922 816 251 426 433 759 354 395 0335 ... 7,922 etc. |
| Currency | @ | 8 | geldbedragen: - 922 337 203 685 477,5808 ... 922 337 203 685 477,5808 Om te kunnen rekenen zonder afrondingsfouten zijn bij geldbedragen 4 cijfers achter de komma noodzakelijk. |
| Date | | 8 | datum: 1 januari 100 ... 31 december 9999 |
| String | | variabel | tekenreeks: Een string kan van 0 tot 2 miljard tekens bevatten |

Tabel 2.2 Overzicht van gegevenstypen in Visual Basic

De variabelen in het project “De spaarrekening zijn dus van de volgende gegevenstypen:

| | | | |
|-------------------|--------------|------------------------------------|----------|
| Beginsaldo, Saldo | geldbedrag | twee cijfers achter de komma | Currency |
| Rentepercentage | in procenten | meestal één cijfer achter de komma | Single |
| Looptijd | in jaren | geheel getal | Integer |
| Jaar | | geheel getal | Integer |

Voordat we overgaan tot het programmeerwerk bekijken we nog twee mogelijkheden om herhalingen te programmeren in Visual Basic.

2.5 Herhalingen programmeren

Er zijn meerdere mogelijkheden om herhalingen te programmeren in Visual Basic. De meest gebruikte daarvan zijn de For ... Next en de Do ... Loop constructies.

In het project “De spaarrekening” dient een Saldo herhaald te worden vermenigvuldigd met een vermenigvuldigingsfactor ($1 + \text{Rentepercentage} / 100$). Ook staat het aantal keren dat de vermenigvuldiging herhaald moet worden vast. De variabele Looptijd geeft dat aantal aan.

2.5.1 De For ... Next herhalingslus

Met een zogenaamde For ... Next herhalingslus kan deze herhaalde vermenigvuldiging worden geprogrammeerd:

```
Saldo = Beginsaldo
For Jaar = 1 To Looptijd
    Saldo = ( 1 + Rentepercentage / 100) * Saldo
    frmSaldo.pctSaldo.Print Jaar, Saldo
Next
```

De For ... Next herhalingslus zit zo in elkaar:

De lus begint met een “For”. Na de For staat de variabele Jaar, een soort teller. Achter de variabele staat aangegeven tot welke waarde van de variabele wordt doorgerekend. Dan volgt de programmacode die aangeeft welke berekening moet worden uitgevoerd en waar de uitvoer naartoe moet worden weggeschreven. De lus eindigt met een “Next”. Deze geeft aan dat er een volgende waarde van de teller, in dit voorbeeld de variabele Jaar, moet worden doorgerekend.

2.5.1 De Do ... Loop herhalingslus

Ook met een zogenaamde Do ... Loop herhalingslus kan de herhaalde vermenigvuldiging worden geprogrammeerd:

```
Saldo = Beginsaldo
Jaar = 1
Do while Jaar < 11
    Saldo = ( 1 + Rentepercentage / 100) * Saldo
    frmSaldo.pctSaldo.Print Jaar, Saldo
    Jaar = Jaar + 1
Loop
```

De Do ... Loop herhalingslus lijkt erg op de For ... Next lus. Bij de Do ... Loop herhalingslus gebruik je echter een andere stopvoorwaarde. In het voorbeeld moet worden doorgerekend zolang het 11^e jaar nog niet is bereikt. De jaren moeten in de Do ... Loop herhalingslus dan ook apart worden opgehoogd, hetgeen in de For ... Next herhalingslus automatisch gebeurt. Een voordeel van de Do ... Loop herhalingslus is dat je de herhalingslus ook kunt gebruiken als het aantal herhalingen niet vast staat. Met een Do ... Loop herhalingslus kan je bijvoorbeeld berekenen hoe lang het duurt voordat 1000 gulden tegen 5 % rente 10000 gulden wordt.

```
Saldo = Beginsaldo
Jaar = 1
Do while Saldo < 10000
    Saldo = ( 1 + Rentepercentage / 100) * Saldo
    frmSaldo.pctSaldo.Print Jaar, Saldo
    Jaar = Jaar + 1
Loop
```

2.6 De programmacode van project “De spaarrekening”

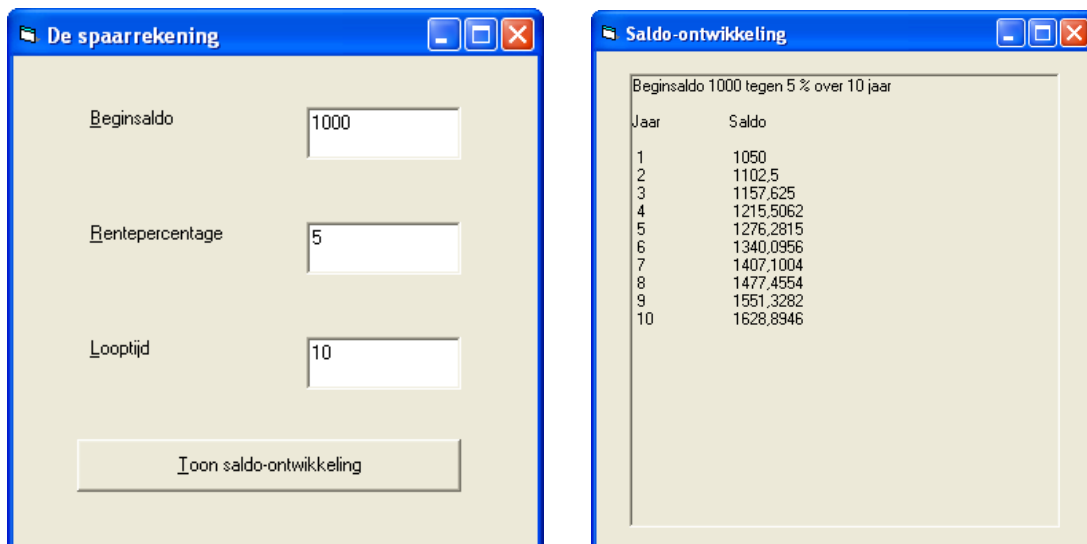
We gaan de opgedane kennis van variabelen en herhalingen in Visual Basic toepassen door de programmacode bij de knop cmdToonsaldo te gaan maken. Daar komt nog eens bij dat we leren hoe we een berekening met waarden uit een eerste formulier, frmSpaarrekening, naar een object (hier: afbeeldingsveld pctSaldo) op een tweede formulier, frmSaldo, moeten schrijven.

☞ Dubbelklik op de knop “cmdToonsaldo” op frmSpaarrekening en voeg de volgende programmacode aan de knop toe:

```
Private Sub cmdToonsaldo_Click()  
Dim Beginsaldo, Saldo As Currency  
Dim Rentepercentage As Single  
Dim Jaar, Looptijd As Integer  
Beginsaldo = txtBeginsaldo.Text  
Rentepercentage = txtRentepercentage.Text  
Looptijd = txtLooptijd.Text  
Saldo = Beginsaldo  
frmSaldo.Show  
frmSaldo.pctSaldo.Print "Beginsaldo " & Beginsaldo & " tegen " & Rentepercentage & " % over " & Looptijd & " jaar "  
frmSaldo.pctSaldo.Print ""  
frmSaldo.pctSaldo.Print "Jaar", "Saldo"  
frmSaldo.pctSaldo.Print ""  
For Jaar = 1 To Looptijd  
    Saldo = Saldo * (1 + Rentepercentage / 100)  
    frmSaldo.pctSaldo.Print Jaar, Saldo  
Next  
End Sub
```

Als je alle code goed hebt ingetypt werkt je programma als volgt.

Bij invoer van de waarden 1000 (gulden), 5(%) en 10 (jaar) op het invoerformulier zoals links afgebeeld toont Visual Basic het uitvoerformulier zoals rechts afgebeeld:



Uitleg van de programmacode:

```
Private Sub cmdToonsaldo_Click()  
Dim Beginsaldo, Saldo As Currency  
Dim Rentepercentage As Single  
Dim Jaar, Looptijd As Integer
```

Hier start de procedure en worden de variabelen opgegeven (gedeclareerd). Zoals je ziet zijn de variabelen Beginsaldo en Saldo gedeclareerd als geldbedrag (Currency), Rentepercentage als klein kommagetal (Single) en Jaar en Looptijd als geheel getal (Integer).

```
Beginsaldo = txtBeginsaldo.Text  
Rentepercentage = txtRentepercentage.Text  
Looptijd = txtLooptijd.Text
```

Hier worden de, voor de verdere berekening benodigde, waarden van Beginsaldo, Rentepercentage en Looptijd uit de tekstvelden gelezen.

```
Saldo = Beginsaldo
```

Hier krijgt de variabele saldo, die steeds weer een nieuwe waarde krijgt, de ingelezen beginwaarde.

```
frmSaldo.Show
```

Deze opdracht zorgt er voor dat formulier frmSaldo in beeld komt.

```
frmSaldo.pctSaldo.Print "Beginsaldo "&Beginsaldo&" tegen " &Rentepercentage&  
" % over "&Looptijd&" jaar "
```

```
frmSaldo.pctSaldo.Print ""  
frmSaldo.pctSaldo.Print "Jaar", "Saldo"  
frmSaldo.pctSaldo.Print ""
```

Deze opdrachten plaatsen een koptekst in het afbeeldingsveld pctSaldo op formulier frmSaldo. Een tekst die letterlijk moet worden afgedrukt staat tussen aanhalingstekens "Beginsaldo", een variabele waarvan de waarde moet worden afgedrukt staat niet tussen aanhalingstekens. Afgedrukte elementen die worden gescheiden door een "&" komen direct naast elkaar te staan, afgedrukte elementen die worden gescheiden door een ",", " worden op tabposities naast elkaar geplaatst. In de eerste regel is er gebruik gemaakt van het scheidingsteken "&" om een vlot lopende zin af te drukken. In de derde regel is er gebruik gemaakt van het scheidingsteken "," om een tabelkop te maken waar de berekende waarden netjes onder komen te staan. In de tweede en vierde regel worden lege regels afgedrukt.

```
For Jaar = 1 To Looptijd
```

```
    Saldo = Saldo * (1 + Rentepercentage / 100)
```

```
    frmSaldo.pctSaldo.Print Jaar, Saldo
```

```
Next
```

Hier is dan de herhalingslus, een zogenaamde "For ... Next" lus. Er wordt tot de Looptijd verstreken is doorgerekend. Telkens wordt het nieuwe saldo berekend en op het afbeeldingsveld pctSaldo op formulier frmSaldo afgedrukt. Ook hier worden de afgedrukte variabelen gescheiden door een "," om het Jaar en Saldo telkens netjes op de tabposities onder de afgedrukte koptekst te krijgen. Na het afdrukken van Jaar en Saldo wordt de variabele Jaar met de code "Next" opgehoogd totdat de variabele Jaar de waarde van de variabele Looptijd heeft bereikt. De code "Next" sluit de herhalingslus af. De programmacode tussen de "For" en "Next" is dus de code die herhaald moet worden uitgevoerd.

```
End Sub
```

Hier eindigt de programmacode.

2.7 De puntjes op de i: “Pimp my Project”:

2.7.1 Een formulier opmaken bij het laden

Een formulier kan je bij het laden eigenschappen toekennen. Zo kunnen we het formulier frmSaldo wat opvrolijken / pimpen door het bij het laden een aantal eigenschappen toe te kennen.



Maak formulier frmSaldo actief door of één keer te klikken op formulier 2 in het formulierenvenster of door te dubbelklikken op Form2 in het projectvenster.

| Jaar | Saldo |
|------|------------|
| 1 | € 1.050,00 |
| 2 | € 1.102,50 |
| 3 | € 1.157,63 |
| 4 | € 1.215,51 |
| 5 | € 1.276,28 |
| 6 | € 1.340,10 |
| 7 | € 1.407,10 |
| 8 | € 1.477,46 |
| 9 | € 1.551,33 |
| 10 | € 1.628,89 |



Dubbelklik op formulier frmSaldo om programmacode aan formulier 2 toe te voegen. Het venster geeft in de bovenste velden aan dat er aan het formulier “Form2” bij de actie “Load” programmacode moet worden uitgevoerd.

```
Private Sub Form_Load()  
Me.BackColor = &HC0&  
Me.Icon = LoadPicture("Sun.ico")  
Me.pctSaldo.BackColor = &H80C0FF&  
End Sub
```

Uitleg programmacode:

Me.BackColor = &HC0&

Me.BackColor maakt de achtergrondkleur van het formulier (that's Me !!) rood (kijk voor de kleurcode in het [palet] bij de eigenschappen ([Properties]) van het formulier.

Me.Icon = LoadPicture("Sun.ico")

Me.Icon plaatst de afbeelding sun.ico in de titelbalk van het formulier. (Je geeft zonnodig het pad op waar de afbeelding gevonden kan worden.)

Me.pctSaldo.BackColor = &H80C0FF&

Deze opdracht maakt de achtergrondkleur van het afbeeldingsveld pctSaldo oranje.

2.7.2 Het format van de af te drukken tekst

Waarschijnlijk vind je het ook niet zo mooi dat de bedragen op het uitvoervenster in vier decimalen worden afgedrukt. Ook dat is programmeerbaar.



Vervang de programmaregel bij de knop cmdToonsaldo:

```
frmSaldo.pctSaldo.Print Jaar, Saldo
```

door:

```
frmSaldo.pctSaldo.Print Jaar, Format(Saldo, "Currency")
```

Met de code Format kan worden aangegeven op welke wijze een variabele afgedrukt dient te worden.

2.8 Programmeeropdracht “De strippenkaart”



Bouw een Visual Basic project “De strippenkaart” dat bestaat uit één formulier. Op het formulier kan de gebruiker in twee tekstvelden opgeven welke prijs hij/zij voor een strippenkaart heeft betaald en hoeveel strippen de kaart bevat. Na een druk op de knop wordt in een afbeeldingsveld op hetzelfde formulier het volgende afgedrukt: wat een rit van 1 t/m het maximale aantal zones op de strippenkaart kost en wat de bijbehorende prijs per zone is. Het bijbehorende PSD is voor het gemak alvast gegeven:

| |
|---|
| Lees Kaartprijs |
| Lees Aantalstrippen |
| $\text{Stripprijs} = \text{Kaartprijs} / \text{Aantalstrippen}$ |
| $\text{Maxzones} = \text{Aantalstrippen} - 1$ |
| Voor Zones = 1 ... Maxzones |
| $\text{Ritprijs} = \text{Stripprijs} * (\text{Zones} + 1)$ |
| $\text{Zoneprijs} = \text{Ritprijs} / \text{Zones}$ |
| Schrijf Zones, Ritprijs, Zoneprijs |

Het resultaat ziet er als volgt uit:



2.9 Programmeeropdracht “Tafels”

☞ Bouw een Visual Basic project “Tafels” dat bestaat uit twee formulieren. Op het eerste formulier kan de gebruiker opgeven welke tafel hij wil zien, bijvoorbeeld de tafel van 5. Op het tweede formulier dient de tafel in een afbeeldingsveld te worden afgedrukt.

Gebruik bij deze opdracht een Do ... Loop herhalingslus.

Bedenk voordat je programmacode gaat intypen hoe het PSD er uit ziet en welke variabelen je gaat gebruiken!

Hieronder zie je een mogelijke gebruikersinterface. Merk op dat ook in het opschrift van het formulier waarop de tafel wordt afgedrukt de waarde van de tafel wordt afgedrukt.

