



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

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

Hoofdstuk 1: Een eerste Visual Basic project

1.0 Leerdoel

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

- kennis maken met de programmeeromgeving van Visual Basic
- kennis maken met de volgende onderdelen van een Visual Basic project:
 - formulieren (Forms)
 - tekstvelden (TextBoxes)
 - labels (Labels)
 - knoppen (CommandButtons)
 - afbeeldingen (ImageBoxes)
- gebeurtenissen in een project (event handlers) beschrijven met programmacode, zoals:
 - programmacode toevoegen aan een knop
 - tekst ophalen uit een tekstveld
 - tekst wegschrijven in een tekstveld
 - een plaatje wegschrijven in een afbeelding
 - een afbeelding schoonmaken
 - tekst wegschrijven in een label
- enkele algemene regels voor het programmeren
 - werken met variabelen en toekenningsopdrachten
- een programma uitvoeren in de programmeeromgeving van Visual Basic
- hoe je een zelfstandig uitvoerbaar programma in Visual Basic maakt

We doen dat aan de hand van twee voorbeelden:

- A. De euro omzetter
- B. Het weerstation

Met dit hoofdstuk ben je, afhankelijk van je tempo, 1 a 2 lessen van 50 minuten bezig.

Voordat je aan de slag kan met dit hoofdstuk dien je:

- A. een persoonlijke map aan te maken op de C:\ schijf van je computer
- B. een viertal bestanden in deze map te plaatsen:
 - Sun.ico
 - Rain.ico
 - Snow.ico
 - Cloud.ico

1.1 Het ontwerpen van programma's in Visual Basic

Een algemene omschrijving van het begrip programmeren is het geven van instructies die in een bepaalde volgorde (programmastructuur) moeten worden uitgevoerd. Zo spreken we van het programmeren van een videorecorder wanneer we het apparaat instellen om op vrijdagavond van 20:00 tot 20:45 een soapserie op te nemen. Ook spreken we van programmeren wanneer een leerling als deze een aantal instructies krijgt van een docent om daarmee een bepaald leerdoel te bereiken. In deze cursus beperken we ons tot het programmeren van een computer. Ook met een computerprogramma wil je altijd een bepaald doel bereiken. Het is belangrijk om, voordat je echt gaat programmeren, goed na te denken hoe je programma er uit moet komen te zien: het programma moet eerst ontworpen worden.

Het ontwerpen van een programma in Visual Basic bestaat uit twee onderdelen:

A. *Het ontwerpen van de programmastructuur*

Welke handelingen moeten achtereenvolgens worden verricht om het gewenste doel te bereiken en op welke manier kan dat zo handig mogelijk?

B. *Het ontwerpen van de gebruikersinterface*

Welke onderdelen gebruik je om de gebruiker informatie in te laten voeren en het programma informatie uit te laten voeren?

In dit eerste hoofdstuk werken we een tweetal voorbeelden uit:

A. *De euro omzetter*

Doel: Dit programma dient een hoeveelheid guldens (invoer) om te zetten in euro's (uitvoer)

B. *Het weerstation*

Doel: Dit programma dient, aan de hand van invoer van de weersoort, met een plaatje weer te geven wat voor weer het wordt.

1.2 De programmastructuur

In een Programma Structuur Diagram beschrijf je in welke volgorde de handelingen in een programma dienen te worden verricht. De PSD's van de euro omzetter en het weerstation zijn achtereenvolgens:

invoer aantal guldens
aantal euro := aantal guldens / 2,20371
uitvoer aantal euro

PSD euro omzetter

invoer weersoort
uitvoer weersoort

PSD weerstation

Natuurlijk zijn de bovenstaande PSD's bijna te kinderachtig voor woorden. Toch is het, vooral bij ingewikkelde programma's, altijd aan te raden om een PSD te maken.

PSD's maken namelijk niet alleen maar duidelijk via welke volgorde het gewenste doel bereikt wordt (**effectiviteit** van de programmastructuur) maar ook of dat op een handige manier gebeurt (**efficiency** van de programmastructuur). De computer moet niet onnodig veel of geen onnodig lange opdrachten uit te hoeven voeren als dat niet persé noodzakelijk is. Je zou kunnen stellen dat voor jou de eis van efficiënt programmeren nauwelijks een rol speelt: de rekenkracht van moderne PC's is enorm groot en de programmeeropdrachten in deze cursus zijn kleinschalig. Toch is het van belang dat je gevoel ontwikkelt voor "netjes programmeren".

1.3 De gebruikersinterface ontwerpen met programmeeronderdelen: projecten, formulieren en objecten.

Een Visual Basic programma noemen we ook wel een project (*project*). Een project kan uit een groot aantal verschillende onderdelen bestaan. De onderdelen die je gebruikt hangen af van de manier waarop je het programma wilt laten werken. Meestal telt een project één of meer formulieren (*forms*). De formulieren en de onderdelen die je daarop zet (knoppen, tekstvensters, etc.) vormen samen de *interface* met de gebruiker van jouw programma's.

We kunnen ons nu wat beter voorstellen wat er in het Visual Basic project (VB project) allemaal moet gebeuren. Op de formulieren uit het project moeten we objecten opnemen zoals invoervensters, knoppen, uitvoervensters, lijsten etc. Elk object (*object*) heeft bepaalde eigenschappen (*properties*) zoals grootte, positie, kleur etc. Aan een object kan ook een gebeurtenis (*event*) worden gekoppeld. In het algemeen verander je bij het laten uitvoeren van een gebeurtenis of event de eigenschappen van een object. Dit is een nogal abstracte formulering van wat er allemaal moet gebeuren in een VB project. Laten we aan de hand van de voorbeelden “euro omzetter” en “weerstation” eens gaan kijken hoe je concreet beslist welke objecten je opneemt in je VB project. Voor het gemak bestaan deze VB projecten uit één formulier.



Het *formulier*: De euro omzetter

Bij dit project ligt het voor de hand te kiezen voor een *tekstveld* voor de invoer van de hoeveelheid guldens. Achter dit tekstveld plaatsen we een *label* met daarop de tekst “guldens” om de gebruiker duidelijk te maken dat in het tekstveld guldens dienen te worden ingevoerd. Onder het tekstveld en de label plaatsen we een *knop* waaraan de gebeurtenis of event omrekenen van de ingevoerde hoeveelheid guldens moet worden gekoppeld. Voor de uitvoer van euro's kiezen we weer een *tekstveld* en een *label*.



Het *formulier*: Het weerstation

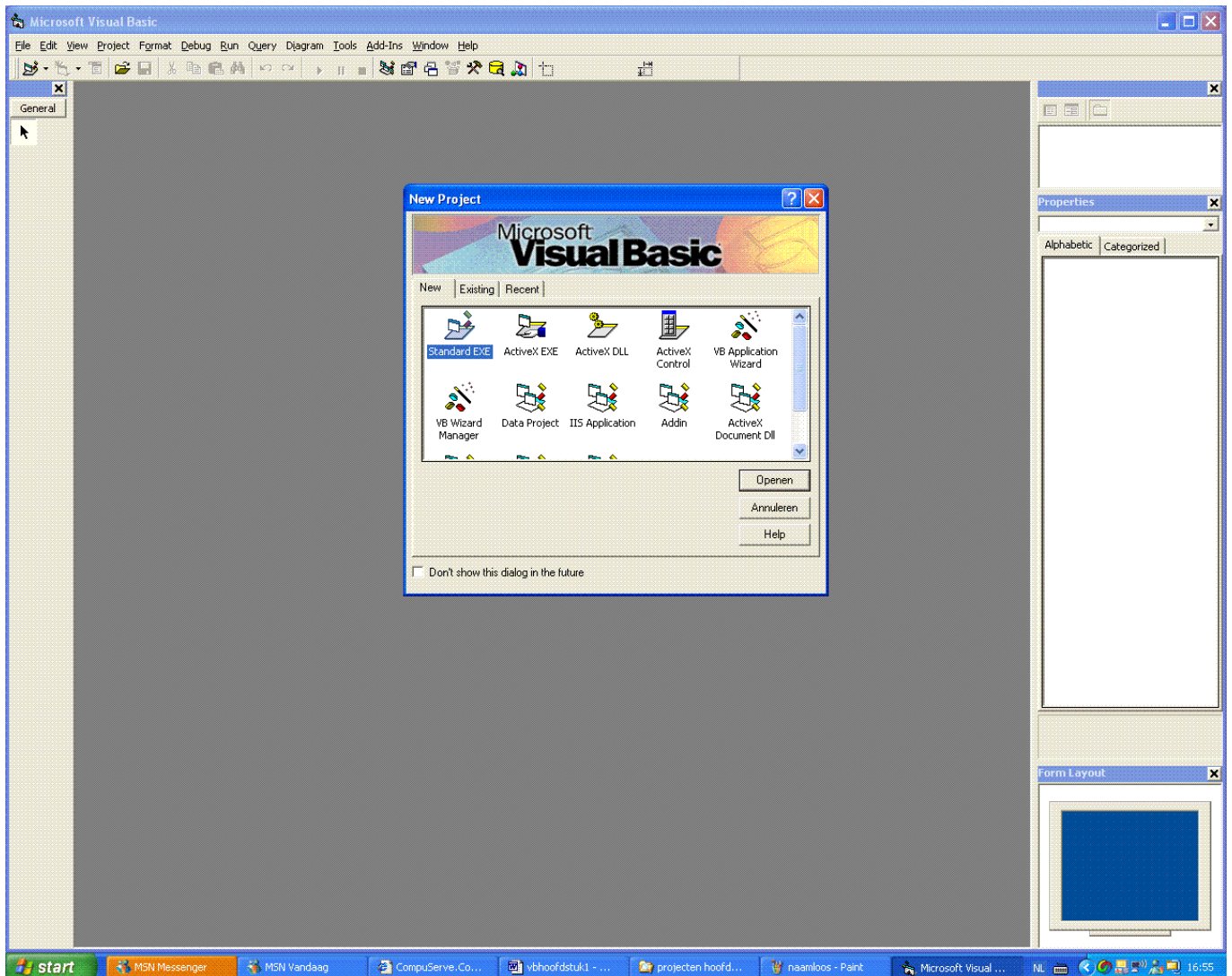
Voor de invoer van het weer is bij het weerstation niet gekozen voor tekstvelden maar voor *knoppen*. De gebruiker zal niet elk willekeurig weertype kunnen aangeven maar alleen de weertypen waarvan een plaatje beschikbaar is. Aan elke knop wordt de gebeurtenis of event gekoppeld dat er een plaatje wordt geplaatst in de *afbeelding* naast de knoppen. Het weerstation is afgemaakt met een *label* bovenin met daarop de tekst “De weersvoorzichten”

Het programmeren in Visual Basic kunnen we nu kort samenvatten als het bouwen van een Visual Basic project waarbij de programmeur vooraf heeft bepaald (1) hoe zijn interface er uit komt te zien (welke formulieren en objecten gebruikt hij?) en (2) welke mogelijke gebeurtenissen in het project met programmacode moeten worden beschreven (*event handlers*).

We gaan nu eens uitzoeken wat dat precies inhoudt. Tijd om aan de slag te gaan met Visual Basic 6.0.

1.4 De programmeeromgeving

☞ Start het programma Visual Basic 6.0



Voor je verschijnt de programmeeromgeving van Visual Basic 6.0 zoals hierboven is weergegeven.

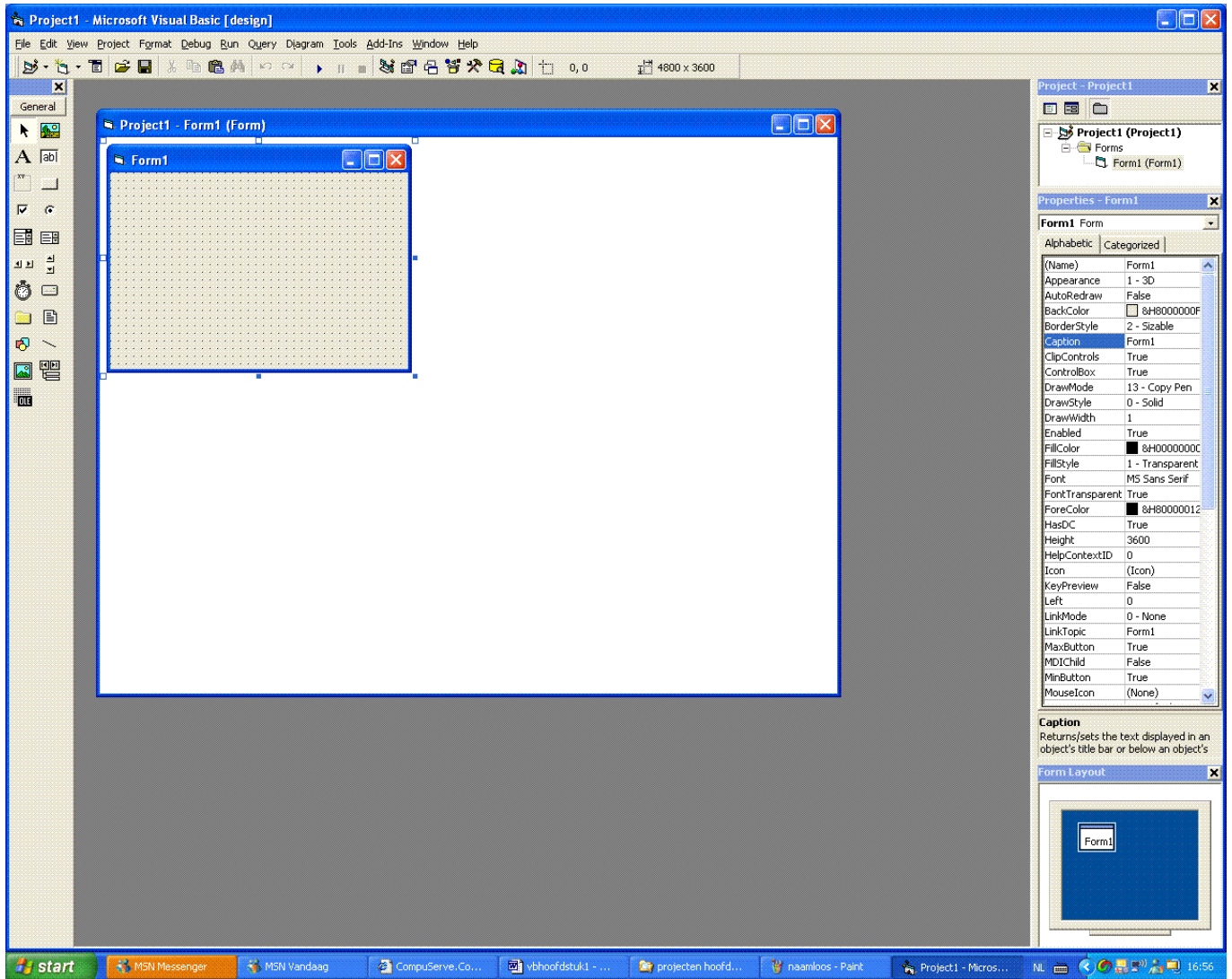
Op het tabblad **New** kan je aangeven wat voor soort nieuw project je wilt gaan maken.

Op het tabblad **Existing** kan je bestaande projecten opvragen.

Op het tabblad **Recent** kan je recent gemaakte projecten opvragen.

☞ Kies voor Standard EXE (op het tabblad New).

Met deze optie start je de bouw van een direct uitvoerbaar (*executable*) Visual Basic project. Een Visual Basic EXE bestand is een programma dat, zoals we later in dit hoofdstuk zullen zien, zelfstandig uitgevoerd kan worden.



Op het scherm verschijnen verschillende vensters:

- **Form 1:** het formuliervenster met daarin het (eerste) formulier waaruit het project gaat bestaan.
- **Project – Project 1:** het projectvenster dat overzicht geeft van alle formulieren van het project.
- **Properties – Form 1:** het eigenschappenvenster waarmee je de eigenschappen van een object (formulier, knop, etc.) kunt veranderen.
- **Form Layout:** het uitvulvenster waarmee je de positie van een object (formulier, knop, etc.) kunt veranderen.
- **General:** de standaard werkbalk waaruit je allerlei objecten naar je formulier kunt overbrengen.

1.5 Plaatsing en naamgeving van objecten op het formulier

We starten met het project “De euro omzetter”.

We bouwen het project volgens het ontwerp op bladzijde 3.



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

naam	[(Name)]	frmEuro_omzetter
opschrift	[Caption]	De euro omzetter

We plaatsen nu eerst het tekstveld waar de gebruiker de hoeveelheid guldens moet gaan invoeren.



Plaats het tekstveld ([TextBox]) uit de werkbalk ([General]) op het formulier.

Geef het tekstveld de volgende eigenschappen ([Properties]):

Naam	[(Name)]	txtGuldens	
Tekst	[Text]		(niets dus)

De naamgeving zoals “frmEuro_omzetter” en “txtGuldens” noemen we wel de **Hongaarse naamgeving**. Het geeft aan dat het om een tekstveld gaat en tegelijkertijd dat in het veld guldens dienen te worden ingevuld. Vervolgens plaatsen we het label met de uitleg “guldens” naast het tekstveld.



Plaats het label ([Label]) uit de werkbalk ([General]) op het formulier.

Geef het label de volgende eigenschappen ([Properties]):

Naam	[(Name)]	lblGuldens
Opschrift	[Caption]	guldens
Lettertype	[Font]	MS Sans Serif, 24 pt

Na het tekstveld en label voor de invoer plaatsen we de knop op het formulier.



Plaats de knop ([CommandButton]) uit de werkbalk ([General]) op het formulier.

Geef de knop de volgende eigenschappen ([Properties]):

Naam	[(Name)]	cmdOmzetten
Opschrift	[Caption]	Zet om in euro's
Lettertype	[Font]	MS Sans Serif, 24 pt

Als laatste kunnen we nu een tekstveld en een label voor de uitvoer op het formulier plaatsen.



Plaats het tekstveld en het label voor de uitvoer van de hoeveelheid euro's op het formulier.

Gebruik vergelijkbare naamgeving en eigenschappen als bij het tekstveld en het label voor de invoer van de hoeveelheid guldens.

De gebruikersinterface is nu gebouwd. Het wordt tijd voor het echte programmeerwerk!

1.6 Programmacode invoeren

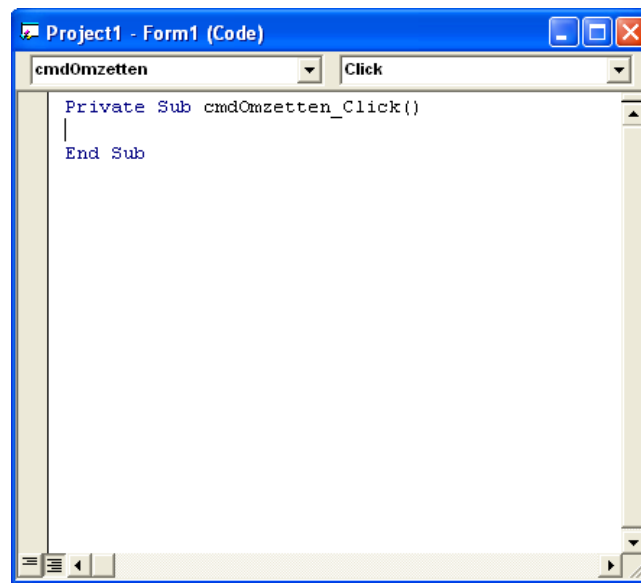
Bij het opbouwen van de interface, het formulier voor “De Euro omzetter”, heb je waarschijnlijk nog niet echt het idee gehad met programmeren bezig te zijn. Toch is dat wel degelijk het geval. Je hebt met de werkbalk [General] op het formulier [Form] een aantal objecten aangemaakt, er een bepaalde positie en grootte aan toegekend en er, bij de [Properties], namen en andere eigenschappen aan toegekend. Er is dus al heel wat ingesteld!

We hebben alleen het belangrijkste deel van het programmeerwerk, de omzetting van guldens naar euro's, nog niet ingevuld. Aan de knop “cmdOmzetten” moet een gebeurtenis (*event*) worden toegevoegd die de omzetting van guldens naar euro's moet gaan afhandelen. Hoe voeg je programmacode aan de knop toe?



Dubbelklik op de knop “cmdOmzetten”

Op je scherm verschijnt het volgende venster:



Het venster geeft in de bovenste velden aan dat er aan de knop “cmdOmzetten” bij de actie “Click” programmacode (een klein programmaatje of subroutine) dient te worden uitgevoerd. De programmacode dient in de lege regels tussen het begin en einde van de subroutine cmdOmzetten_Click() te worden ingetypt. We noemen de subroutine cmdOmzetten_Click() ook wel een *event handler* omdat de subroutine beschrijft welke gebeurtenis bij het klikken op de knop cmdOmzetten moet worden afgehandeld.

```
Private Sub cmdOmzetten_Click()
```

```
End Sub
```

De programmacode, die bij het klikken op de knop “cmdOmzetten” dient te worden uitgevoerd, typen we in de lege regels tussen het begin en einde van de subroutine cmdOmzetten_Click().



Voer in de lege regels de programmacode verder in zoals hieronder is weergegeven.

```
Project1 - Form1 (Code)
cmdOmzetten Click
Private Sub cmdOmzetten_Click()
Dim guldens
Dim euros
guldens = txtGuldens.Text
euros = guldens / 2.20371
txtEuros.Text = euros
End Sub
```

Uitleg van de programmacode:

Private Sub cmdOmzetten_Click()

De Sub instructie start een procedure of subroutine, de naam van de subroutine staat er altijd direct achter. Hier start dus de procedure die plaats vindt wanneer gedrukt wordt op de knop cmdOmzetten. De toevoeging Private staat erbij om ervoor te zorgen dat de procedure alleen maar binnen het formulier frmEuro_omzetter kan worden gebruikt. Als je de Private instructie weglaat, dan kan de procedure ook in andere projecten worden gebruikt.

Dim guldens

Dim euros

De instructie Dim gaat vooraf aan de opgave van variabelen die in het programma gebruikt gaan worden. Het opgeven van de variabelen heet ook wel declaratie van de variabelen.

guldens = txtGuldens.Text

Hier wordt met een instructie aan de variabele guldens een waarde toegekend. Moeilijk geformuleerd: de variabele guldens krijgt de waarde van de eigenschap [Text] van het object [txtGuldens]. Makkelijk geformuleerd: heeft de gebruiker in het tekstveld een getal ingevoerd, dan krijgt de variabele guldens dus deze waarde.

euros = guldens / 2.20371

Hier wordt met een instructie aan de variabele euro's een waarde toegekend. Om van guldens euro's te maken dien je te delen door 2.20371. Merk op dat een punt gebruikt wordt, géén komma!

txtEuros.Text = euros

Hier wordt aan de eigenschap [Text] van het object [txtEuros] de waarde van het berekende aantal euro's toegekend: in het uitvoerveld verschijnt de berekende hoeveelheid euro's.

End Sub

Hier eindigt de procedure cmdOmzetten_Click()

1.7 Een Visual Basic project opslaan

- ☞ Kies voor “File” / “Save Project As” uit de menubalk om het project op te slaan.
Verzin zelf een passende naam.
Sla het bestand op in je eigen map!

Visual Basic slaat bestanden van verschillende soorten op, waaronder: een *.VBP bestand waarin het hele project wordt opgeslagen, *.FRM bestand(en) waarin de formulieren uit het project worden opgeslagen.

Het is verstandig om een programma (Visual Basic project) pas uit te laten voeren wanneer je het programma hebt opgeslagen. Bij het uitvoeren van het programma kan namelijk best een en ander fout gaan.

1.8 Een programma uit laten voeren

- ☞ Kies of voor (1) “Run” / “Start” uit de menubalk of voor (2) de F5-toets of voor (3) icoon 

Als je fouten hebt gemaakt in de programmacode, dan meldt Visual Basic dat.

- ☞ Probeer je programma “De euro omzetter” uit.

Bij het drukken op de knop btnOmzetten wordt de programmacode geladen en uitgevoerd. In het uitvoervenster verschijnt steeds het berekende aantal euro's. Probeer ook eens tekst in te voeren

- ☞ Probeer ook eens tekst in te voeren in plaats van getallen. Wat gebeurt er?

Visual Basic geeft aan dat de invoer van een verkeerd type is. Op dit probleem komen we later in de cursus terug.

1.9 Een zelfstandig uitvoerbaar programma maken

Bij het opstarten van Visual Basic heb je aangegeven een nieuw, zelfstandig uitvoerbaar programma (executable) te willen gaan maken. Tot nu toe heb je het Visual Basic project “Euro omzetter” alleen maar in de ontwikkelomgeving van Visual Basic laten werken. Om een zelfstandig uitvoerbaar programma te maken heb je nog één handeling nodig.

- ☞ Kies voor “File” / “Make euroomzetter.exe”

Opmerking: Het programma, dat je nu gemaakt hebt, werkt niet helemaal zelfstandig. Op een PC waar je het programmaatje wilt laten werken dient een zogenaamde runtime bibliotheek aanwezig te zijn die de Visual Basic programmacode uit kan voeren. Dit bestand heet MSVBVM60.DLL (Microsoft Visual Basic Virtual Machine 6.0). Met deze bibliotheek kan jouw programma dus ook op computers, waarop geen Visual Basic is geïnstalleerd, uitgevoerd worden.

- ☞ Sluit Visual Basic en probeer via “Start” / “Programma's” / “Windows Verkenner” of het programma euroomzetter.exe, dat je zojuist in je map hebt opgeslagen, ook “zelfstandig” werkt door te dubbelklikken op de bestandsnaam. Sluit na het uitproberen het programma weer af.

1.10 Een bestaand project aanpassen



Open Visual Basic weer en kies voor het tabblad “Recent”
Kies voor het project “De euro omzetter”.
Dubbelklik zonnodig in het projectvenster op “Forms” / “Form 1” om het formulier weer op het beeldscherm te laten verschijnen.

We gaan nu de programmacode aanpassen zodat “De euro omzetter” niet alleen guldens naar euro’s kan omrekenen maar ook weer andersom.

De bijbehorende gebruikersinterface moet worden:



Wijzig het formulier zodat de bovenstaande gebruikersinterface ontstaat.
Voeg code toe aan de knop “van euro’s naar guldens”.
Probeer of je gewijzigde project ook echt werkt.
Sla het project onder de naam “De uitgebreide euro omzetter” in je persoonlijke map op.

1.11 Even reflecteren: programmacode

In procedures komen verschillende soorten instructies voor. Gebruikers zullen informatie moeten invoeren in *objecten* zoals TextBoxes (tekstvelden etc.). Bij het aanroepen van deze objecten zullen we rekening moeten houden met grammaticaregels, ook wel **syntaxregels** genoemd. Misschien is je opgevallen dat Visual Basic tijdens het typen van de instructies al aangeeft hoe deze regels luiden.

Variabelen zijn belangrijke geheugens in de programmacode om ingevoerde informatie te kunnen onthouden en om met deze informatie door te kunnen rekenen. Variabelen dienen wel, aan het begin van de procedure waarin ze worden gebruikt, met de programmacode

Dim < variabele >

Dim guldens
Dim euros

te worden gedeclareerd. Na declaratie kunnen er, al dan niet via objecten als TextBoxes door de gebruiker, waarden aan de variabelen worden toegekend:

< variabele > = waarde

guldens = txtGuldens.Text
euros = guldens / 2.20371

Ook het wegschrijven van een resultaat naar een object is een aparte instructiesoort. In het voorbeeld gebruikten we de txtEuros.Text instructie. In het algemeen gebruiken we de syntaxbeschrijving:

< object >.< eigenschap > = < inhoud >

txtEuros.Text = euros

In het verdere verloop van de cursus staan twee zaken centraal:

- A. *het uitbreiden van onze kennis van programmastructuren en het omgaan met variabelen*
- B. *het uitbreiden van onze kennis van het omgaan met objecten*

We starten met een tweede voorbeeldproject, “Het weerstation”. In dit voorbeeld zullen we ons minder dan in het voorbeeld van “De euro omzetter” concentreren op het omgaan met de programmeeromgeving en meer concentreren op het programmeren van gebeurtenissen.

1.12 Het weerstation

We gaan nu verder met het project “Het weerstation”.

☞ Open een nieuw Visual Basic project.

We bouwen de gebruikersinterface van het project volgens het ontwerp op bladzijde 3.

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

naam	[(Name)]	frmWeerstation
opschrift	[Caption]	Het weerstation

We plaatsen eerst de knoppen op het formulier.

☞ We plaatsen knoppen ([CommandButton]) uit de werkbalk ([General]) op het formulier. Geef de knoppen de volgende eigenschappen ([Properties]):

Naam	[(Name)]	cmdZon
Opschrift	[Caption]	Zon

Naam	[(Name)]	cmdRegen
Opschrift	[Caption]	Regen

Naam	[(Name)]	cmdSneeuw
Opschrift	[Caption]	Sneeuw

Naam	[(Name)]	cmdBewolkt
Opschrift	[Caption]	Bewolkt

Een onderdeel, dat we nog niet eerder zijn tegengekomen en aan het project gaan toevoegen, is een kader waarin een afbeelding kan worden opgenomen.

☞ Plaats een afbeelding ([ImageBox]) uit de werkbalk ([General]) naast de knoppen. Geef het afbeelding de volgende eigenschappen ([Properties]):

Naam	[(Name)]	imgWeer
Uitrekken	[Stretch]	True

Een plaatje dat te klein is voor het afbeelding wordt met de waarde:

Stretch = True (Uitrekken = Waar)

automatisch uitgerekte. Als laatste plaatsen we nu een label voor de kopstekst op het formulier.

☞ Plaats een label met de kopstekst “De weersvoorzichten” op het formulier.

De gebruikersinterface is nu gebouwd. Het wordt weer tijd voor programmeerwerk.

Aan elke knop moet een gebeurtenis worden toegevoegd. Die gebeurtenis bestaat uit het toekennen van een inhoud aan de afbeelding bij het drukken op de betreffende knop. Als voorbeeld werken we de gebeurtenis bij het drukken op de knop cmdZon uit.

☞ Dubbelklik op de knop cmdZon en voeg er de volgende code aan toe:

```
Private Sub cmdZon_Click()  
imgWeer.Picture = LoadPicture("Sun.ico")  
End Sub
```

(Zorg ervoor dat de plaatjes in dezelfde map als je project staan)

Met de opdracht `imgWeer.Picture` wordt aan de eigenschap `Picture` van de afbeelding `imgWeer` het bestand `Sun.ico` toegekend.

☞ Voeg nu ook code toe aan de knoppen `cmdRegen`, `cmdSneeuw` en `cmdBewolkt` zodat bij het drukken op deze knoppen respectievelijk de bestanden `Rain.ico`, `Snow.ico` en `Cloud.ico` in de afbeelding `imgWeer` verschijnen.

Het weerstation is nu klaar! Tijd om het project op te slaan en uit te proberen.

☞ Sla het project “Het weerstation” op en probeer het programma uit.

Ook nu weer zijn er allerlei mogelijkheden om “Het weerstation” uit te breiden:



We breiden “Het weerstation” uit met een knop `cmdMaakschoon` die de afbeelding `imgWeer` schoon maakt en een label `lblCommentaar` met daarop het weertype.

☞ Plaats een knop `cmdMaakschoon` op het formulier. Voeg de volgende code toe:

```
Private Sub cmdMaakschoon_Click()  
imgWeer.Picture = LoadPicture(None)  
lblCommentaar.Caption = “ ”  
End Sub
```

Ook wordt een label lblCommentaar toegevoegd met daarop het weertype in 24 pt, gecentreerde tekst.



Plaats een label lblCommentaar op het formulier, onder de afbeelding imgWeer. Wijzig de eigenschappen ([Properties]) uitlijning ([Alignment]) en lettertype ([Font]). Voeg aan de knop cmdZon code toe. De code wordt nu:

```
Private Sub cmdZon_Click()  
imgWeer.Picture = LoadPicture("C:\DavidLans\Sun.ico")  
lblCommentaar.Caption = "Zon"  
End Sub
```

Voeg aan de knoppen cmdRegen, cmdSneeuw en cmdBewolkt op vergelijkbare wijze code toe.



Voeg aan de knop cmdMaakschoon code toe zodat niet alleen de afbeelding maar ook het label bij een druk op de knop wordt schoongemaakt.

1.14 Progammeeropdracht



Ontwerp en bouw een Visual Basic project “De warmte omzetter” dat:

- Fahrenheit in Celsius om kan zetten en andersom volgens de formule: $F = 1,8 C + 32$
- Hint: wijzig project “de Euro-omzetter” en bewaar het gewijzigde project onder een andere naam

1.15 Progammeeropdracht



Ontwerp en bouw een Visual Basic project “De verwisselaar” dat:

- met twee tekstvelden om invoer vraagt en
- bij een druk op de knop deze teksten van veld kan laten verwisselen
- Hint: zie Instruct paragraaf 9.7.2 opgave 3

1.16 Programmeeropdracht



Ontwerp en bouw een Visual Basic project “De rekenmachine” dat:

- twee getallen als invoer vraagt en
- deze getallen kan optellen, aftrekken, vermenigvuldigen en delen en
- het antwoord als uitvoer geeft
- Hint: bijna alle knoppen zullen een goed resultaat geven. Er gaat echter iets mis bij de “+” !? Wat hier mis gaat is dat de “+” ook een opdracht is om teksten aan elkaar te plakken. Visual Basic behandelt de invoer dus in principe als tekst. Je zult Visual Basic moeten laten weten dat de invoer in de tekstvelden bestaan uit getallen. De code die je toevoegt aan de “+” knop moet beginnen met:

```
Dim getal1 As Single  
Dim getal2 As Single  
Dim antwoord As Single
```

De toevoeging “As Single” maakt Visual Basic duidelijk dat de variabelen getallen zijn. In hoofdstuk 2 zullen we nader ingaan op verschillende typen variabelen in Visual Basic.