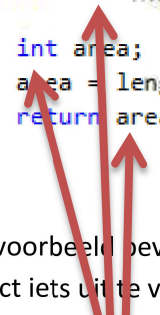


6.3. Functiemethoden en resultaten

In onze eerdere voorbeelden van argumenten en parameters werden de waarden die de methode gebruikte aan de methode doorgegeven. We moeten echter vaak ook methoden programmeren die een berekening uitvoeren en een resultaat naar de rest van het programma terugsturen, zodat het resultaat in de daaropvolgende berekeningen gebruikt wordt. In dat geval moeten we een *return*-statement in een methode gebruiken. Dit noemt men een **functie**. Bijvoorbeeld :

```
private void button1_Click(object sender, EventArgs e)
{
    int a;
    a = OppervlakteRechthoek(10, 20);
}

private int OppervlakteRechthoek(int length, int width)
{
    int area;
    area = length * width;
    return area;
}
```



Dit voorbeeld bevat een aantal nieuwe elementen. Nu gebruikt men een methode (functie) niet om direct iets uit te voeren, maar om iets in een variabele te steken. Bovendien hebben we in plaats van **void** het type gespecificeerd dat de functie van de aanroeper moet retourneren. Dit zou bijvoorbeeld ook een double of string kunnen zijn, als je één van deze twee types terugstuurt. Je moet de geretourneerde waarde 'opvangen', je steekt dit dus in een variabele.

OEFENINGEN

- 1) Schrijf een functiemethode EuroEquivalent die bij een gegeven aantal dollars het aantal euro's berekent. Een voorbeeld van de aanroep is :
double euro = EuroEquivalent(37.50);

Zoek zelf de wisselkoers op.

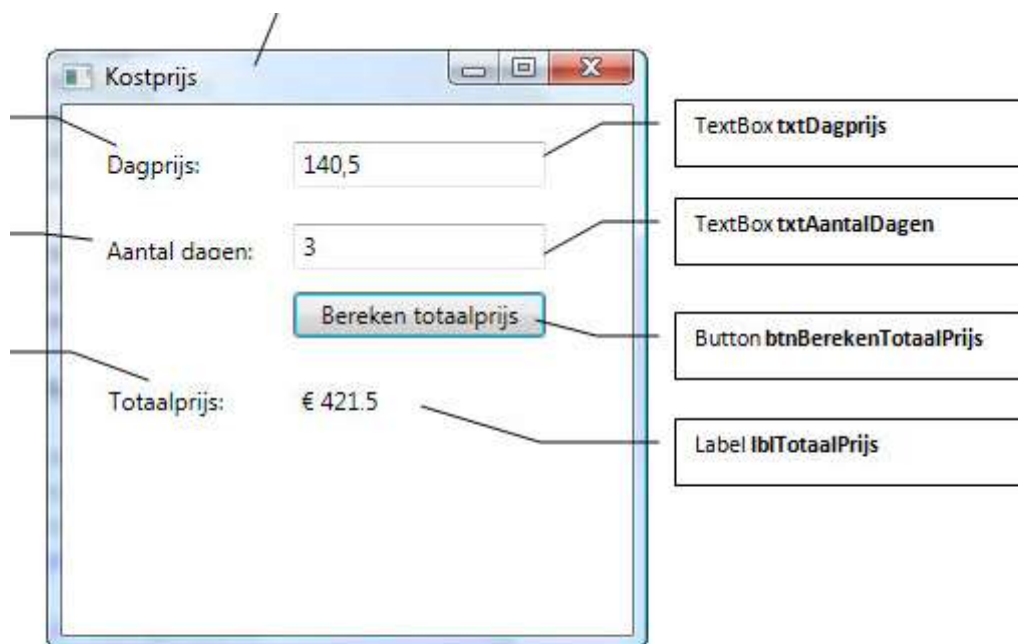
- 2) Schrijf een functiemethode KubusVolume die bij gegeven lengte van een ribbe (zijde van een kubus) het volume berekent. Een voorbeeld van een aanroep is :
double vol = KubusVolume(1.2);

//Gebruik hierbij de formule van de inhoud van een kubus

- 3) Schrijf een funciemethode `TijdInSec` met drie integers als argumenten, die de tijd in uren, minuten en seconden voorstelt. Deze moet de totale tijd in seconden retourneren.
Bijvoorbeeld:
`int aantSeconden = TijdInSec(1,1,2); // retourneert 3662`

UITBREIDING

- 4) Schrijf een funciemethode `TijdsVerschilInSec` met zes argumenten en met een integer als resultaat. De invoer bestaat uit twee tijdstippen in uren, minuten en seconden en retourneert het verschil tussen deze tijdstippen in seconden.
Dus `TijdsVerschilInSec(int uur1, int minuut1, int seconden1, int uur2, int minuut2, int seconden2);`
- 5) Creëer volgende oefening met een methode (het totaal **wordt geretourneerd** naar de hoofdcode) :



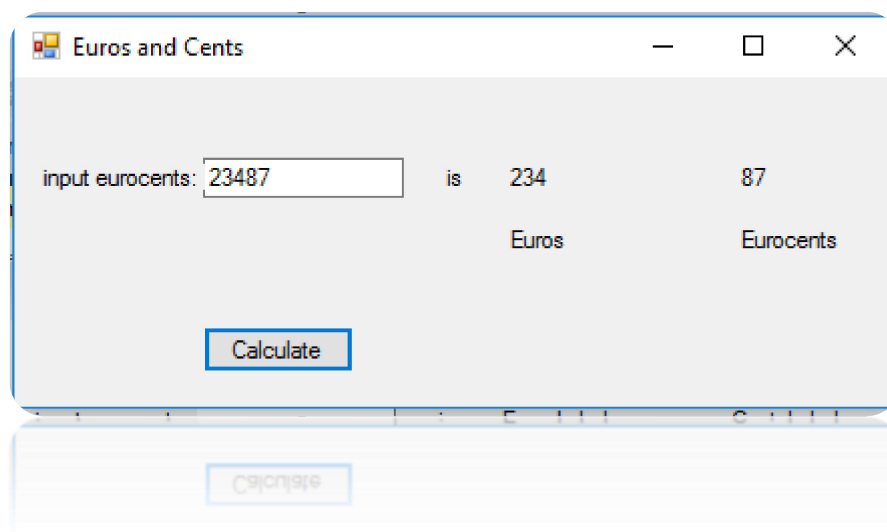
6.4. Argumenten doorgeven als referentie

Tot dusver hebben we steeds het concept van het doorgeven van argumenten als waarde gebruikt, ofwel in combinatie met een (void-)methode ofwel in combinatie met een functie. We gebruikten een functie wanneer er één enkele waarde aan de aanroeper moest worden teruggegeven. Tot nu toe werkte dit prima, maar er is een situatie die we nog niet behandeld hebben : wat moet er gebeuren als onze methode meer dan één resultaat moet retourneren? Een voorbeeld:

Maak een methode die bij een gegeven aantal centen het overeenkomstige gehele aantal euro's en het resterende aantal centen berekent en dus teruggeeft. We hebben dus één invoer en twee resultaten.

6.4.1. Het out- argument

We pakken nu het probleem uit de vorige paragraaf aan : een methode die een aantal centen omzet in een aantal euro's en het resterende aantal eurocenten. Hiervoor moeten we een methode schrijven met een invoerparameter en twee resultaten. Zie hieronder :



- 1) Schrijf nu een methode waar je het totaal aantal centen meegeeft als argument.
- 2) Je zit al met een probleem : Je moet de hoeveelheid euro's en eurocents terugkrijgen. Dit zijn twee variabelen en gaan niet op die manier als de vorige hoofdstukken!

Zie code hieronder :

```
namespace Euros_and_Cents
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void calculateButton_Click(object sender, EventArgs e)
        {
            //declareren

            int aantalCenten;
            int volledigeEurosOver;
            int overgeblevenCenten;

            invoer
            aantalCenten = Convert.ToInt32(inputTextBox.Text);

            //verwerking

            EurosAndCents(aantalCenten, out volledigeEurosOver, out
overgeblevenCenten);

            // in plaats van volledigeEurosOver = EurosAndCents(aantalCenten); want je krijgt TWEE
variabelen terug!

            // uitvoer

            eurosLabel.Text = Convert.ToString(volledigeEurosOver);
            centsLabel.Text = Convert.ToString(overgeblevenCenten);
        }

        private void EurosAndCents(int totalCents,
out int euros,
out int centsLeft)
        {
            euros = totalCents / 100;
            centsLeft = totalCents % 100;
        }
    }
}
```



Door gebruik te maken van het out-argument kan je meerdere variabelen terughalen. Ook hier maakt men gebruik van **PARAMETER - BINDING**. `aantalCenten` wordt doorheen het programma gelijk gesteld aan `totalCents`. Ook `Euros` & `centsLeft` worden gelijkgesteld aan `volledigeEurosOver` en `overgeblevenCenten`.



Pas nu deze oefening aan zodat je tweemaal een invoer hebt en tweemaal een output. Gebruik daarvoor dezelfde methode als eerder gemaakt – Zie hieronder :

OEFENINGEN

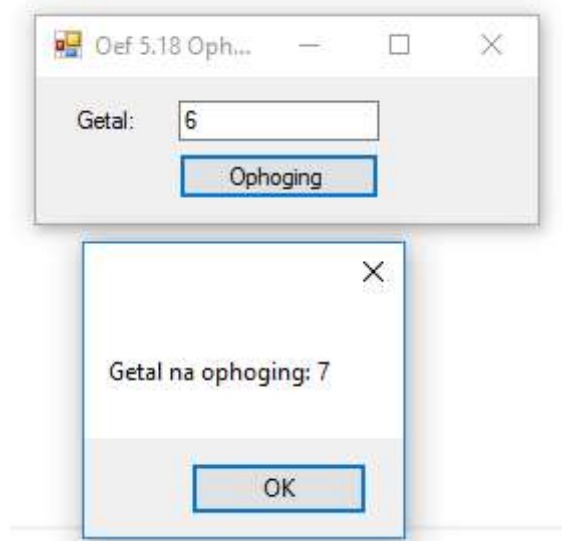
- 1) Schrijf een methode Ophoging, die zijn argument met 1 ophoogt. De messagebox wordt **NIET** in de methode opgeroepen maar in de hoofdcode. Daarmee bedoel ik :

```
private void ophogenButton_Click(object sender, EventArgs e)
{
    [redacted]

    Ophoging([redacted]);
    MessageBox.Show("Getal na ophoging: " + getal2);
}

private void Ophoging([redacted])
{
    [redacted] + 1;
}
```

Een voorbeeld van een aanroep is :



- 2) Schrijf een methode genaamd `SecNaarUMS`, die een aantal seconden als invoer heeft en dit omzet naar uren, minuten en seconden. Maak gebruik van de modulo operator. Een voorbeeld : 3662 seconden is 1 uur, 1 minuut en 2 seconden.
- 3) Schrijf een methode waar je als parameters twee getallen moet meegeven. In deze methode berekent die zowel de som , als het verschil als de vermenigvuldiging van deze twee getallen. De resultaten van deze drie berekeningen geeft hij alle drie terug en worden weergegeven in een label.