

```
int GrootsteGemeneDeler(int getal1, int getal2)
{
    while (getal2 != 0)
    {
        int rest = getal1 % getal2;
        getal1 = getal2;
        getal2 = rest;
    }

    return getal1;
}

static void Main(string[] args)
{
    int getal1 = Int32.Parse(Console.ReadLine());
    int getal2 = Int32.Parse(Console.ReadLine());
    int ggd = GrootsteGemeneDeler(getal1, getal2);
    Console.WriteLine("{0} is de GGD van {1} en {2}.",
        ggd, getal1, getal2);
    Console.ReadKey();
}
```

CO
- ORIGIN
'warn'.

monitor

thing 2
duties 3
picture

Programmeren 1 (C#)

Gerwin van Dijken (gerwin.vandijken@inholland.nl)

Programma periode 1.1 (Programmeren 1)

01 (wk-36)	Inleiding / Visual Studio 2017/2019
02 (wk-37)	Sequentie
03 (wk-38)	Selectie
04 (wk-39)	Iteratie
05 (wk-40)	Array's
06 (wk-41)	Methoden
07 (wk-42)	Herhaling / oefententamen
08 (wk-43)	<i>roostervrije week</i>
09 (wk-44)	praktijktentamen (<i>computer opdrachten</i>)
10 (wk-45)	-

Inleiding – **Analyse** / Ontwerp

- Programmeren: software/programma schrijven voor een bepaald probleem
- Programma: reeks instructies voor de computer om uit te voeren
 - welke instructies?
 - in welke volgorde?
 - welke gegevens?

Inleiding – Analyse / **Ontwerp**

- Ontwerp: instructies/algorithmen vastleggen in pseudocode (of in een PSD)
- Pseudocode: 'onechte' programmeertaal / informeel karakter
- PSD: Programma Structuur Diagram
(Nassi-Schneidermann diagram, 1972)
- *"Een algoritme is een eindige reeks instructies die vanuit een gegeven begintoestand naar een beoogd doel leidt".*
(Wiki, n.d.)

Fasen van software ontwikkeling

- 1) analyse
 - Wat voor gegevens heb ik nodig?
 - Waar haal ik deze gegevens vandaan?
 - Wat moet er met deze gegevens gebeuren?

- 2) ontwerp
 - Beschrijf het algoritme m.b.v. pseudocode / een PSD
(testen: controle op logische fouten)

- 3) realisatie
 - Zet pseudocode om in een programma (instructies)
(testen: controle op syntax-fouten en logische fouten)

Voorbeeld - analyse

(1/4)

- *Bereken het bedrag voor getankte benzine.*
- Analyse
 - Welke gegevens nodig?
 - prijs van één liter benzine
 - aantal liters getankt
 - Wat moet hiermee gedaan worden?
 - aantal liters vermenigvuldigen met de prijs van één liter

- *Bereken het bedrag voor getankte benzine.*
 1. Vraag liter prijs van benzine
 2. Vraag aantal liters benzine getankt
 3. Bereken het bedrag door het aantal liters te vermenigvuldigen met de prijs van één liter
 4. Toon het bedrag

Voorbeeld - analyse

(3/4)

- *Bereken het bedrag voor getankte benzine.*

(gebruik nu variabelen en operatoren)

1. lees literprijs
2. lees aantal_liters
3. bedrag = aantal_liters * literprijs
4. toon bedrag

Geef variabelen
altijd betekenisvolle
namen!!

- *Bereken het bedrag voor getankte benzine.*

(gebruik eventueel een constante)

1. LITERPRIJS = 1.79 ← constante
2. lees aantal_liters
3. bedrag = aantal_liters * LITERPRIJS
4. toon bedrag

Voorbeeld – ontwerp (pseudocode)

- Een van de 3 basisstructuren in een (imperatieve) programmeertaal is de sequentie (“opeenvolging”)

```
LITERPRIJS = 1.79
```

```
lees aantalLiters
```

```
bedrag = aantalLiters * LITERPRIJS
```

```
toon bedrag
```

Voorbeeld – ontwerp (PSD)

liters

LITERPRIJS \leftarrow 1.79
lees aantal_liters
bedrag \leftarrow aantal_liters * LITERPRIJS
toon bedrag

3 basisstructuren - pseudocode

- Sequentie (*opeenvolging*)

```
instruction 1  
instruction 2  
instruction 3
```

- Selectie (*keuze*)

- *als ... dan ... anders ...*

```
if condition(s)  
    instruction(s)  
else  
    instruction(s)
```

- Iteratie (*herhaling*)

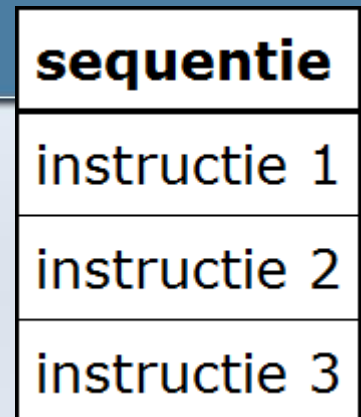
- *zolang ... doe*
- *doe ... zolang*

```
while condition(s)  
    instruction(s)
```

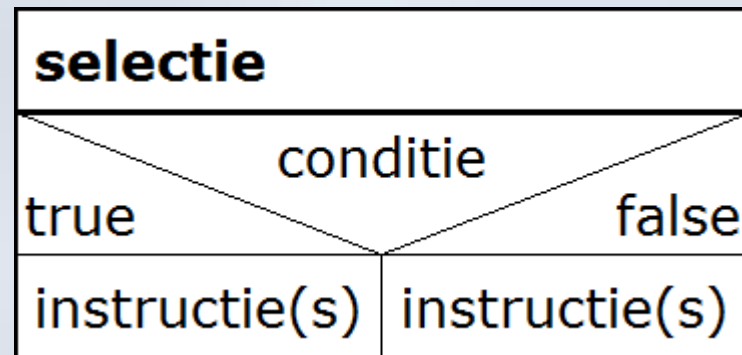
```
do instruction(s)  
while condition(s)
```

3 basisstructuren - PSD

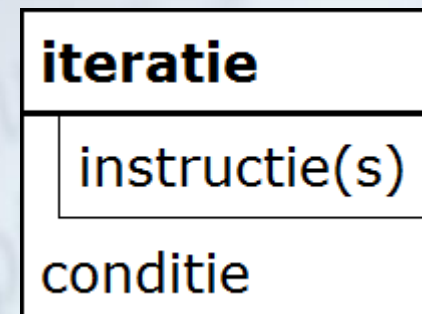
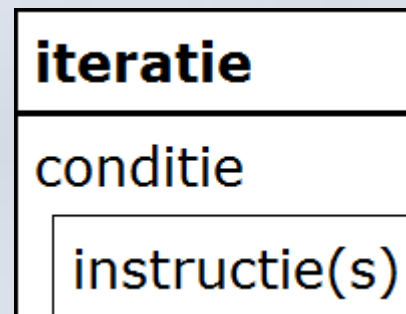
- Sequentie (*opeenvolging*)



- Selectie (*keuze*)
 - *als ... dan ... anders ...*



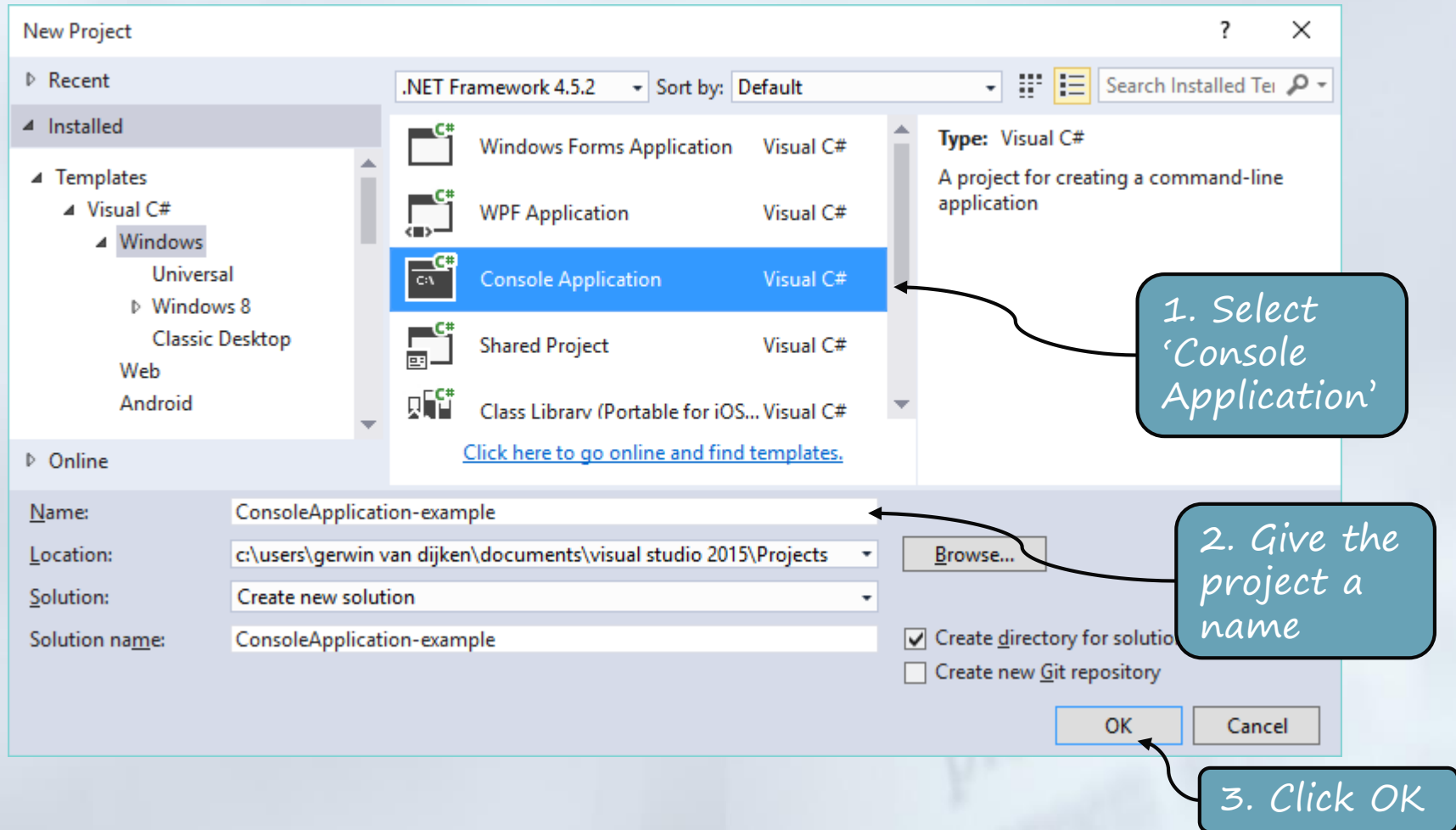
- Iteratie (*herhaling*)
 - *zolang ... doe*
 - *doe ... zolang*



Voorbeeld – Console applicatie

(1/6)

- Create a new project via (menu) File | New | Project...



Voorbeeld - Console applicatie

(2/6)

- *Bereken het bedrag voor getankte benzine.* (Console applicatie)

```
class Program
{
    static void Main(string[] args)
    {
        // 1. lees aantal liters getankt
        // ...

        // 2. bereken het bedrag
        // ...

        // 3. toon het bedrag
        // ...
    }
}
```

Voorbeeld - Console applicatie

(3/6)

■ Stap 1: invoer / lezen

```
class Program
{
    static void Main(string[] args)
    {
        // 1. lees aantal liters getankt
        Console.Write("Geef het aantal liters getankt: ");
        string invoer = Console.ReadLine();

        // 2. bereken het bedrag
        // ...

        // 3. toon het bedrag
        // ...
    }
}
```



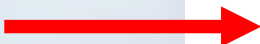
■ Datatype conversie van string naar double

```
static void Main(string[] args)
{
    // 1. lees aantal liters getankt
    Console.WriteLine("Geef het aantal liters getankt: ");
    string invoer = Console.ReadLine();

    // zet string om naar double (datatype conversie)
    double aantalLiters = double.Parse(invoer);
    // of: double aantalLiters = Convert.ToDouble(invoer);

    // 2. bereken het bedrag
    // ...

    // 3. toon het bedrag
    // ...
}
```



Voorbeeld - Console applicatie

(5/6)

- Stap 2:
bereken
bedrag


```
class Program
{
    const double LITERPRIJS = 1.78;

    static void Main(string[] args)
    {
        // 1. lees aantal liters getankt
        Console.Write("Geef het aantal liters getankt: ");
        string invoer = Console.ReadLine();

        // zet string om naar double (datatype conversie)
        double aantalLiters = double.Parse(invoer);

        // 2. bereken het bedrag
        double bedrag = aantalLiters * LITERPRIJS;

        // 3. toon het bedrag
        // ...
    }
}
```



Voorbeeld - Console applicatie

(6/6)

■ Stap 3: uitvoer

*Plaats (zinvol)
commentaar bij
de code.*


```
class Program
{
    const double LITERPRIJS = 1.78;

    static void Main(string[] args)
    {
        // 1. lees aantal liters getankt
        Console.Write("Geef het aantal liters getankt: ");
        string invoer = Console.ReadLine();

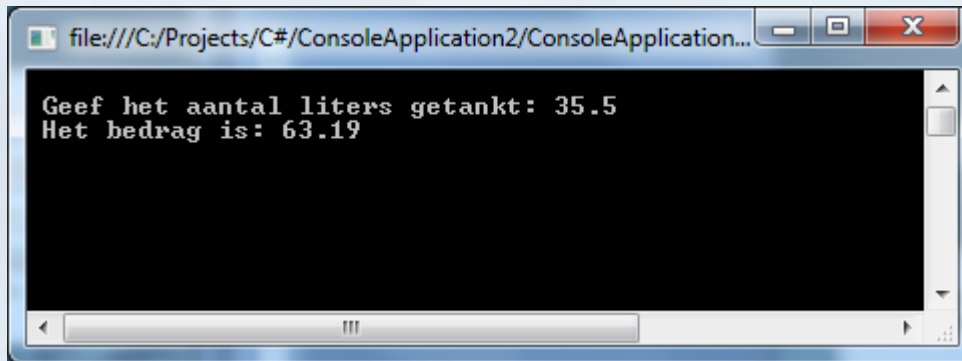
        // zet string om naar double (datatype conversie)
        double aantalLiters = double.Parse(invoer);

        // 2. bereken het bedrag
        double bedrag = aantalLiters * LITERPRIJS;

        // 3. toon het bedrag
        Console.WriteLine("Het bedrag is: " + bedrag);
    }
}
```



Voorbeeld - resultaat



- Wacht op invoer van gebruiker:

```
static void Main(string[] args)
{
    // ...

    // wacht totdat gebruiker een (willekeurige) toets indrukt
    Console.ReadKey();
}
```

Declaratie / initialisatie var/const

```
class Program
{
    // declaratie en initialisatie van constante
    → const double LITERPRIJS = 1.78;

    static void Main(string[] args)
    {
        // declaratie van variabelen
        → string invoer;
        double aantalLiters;

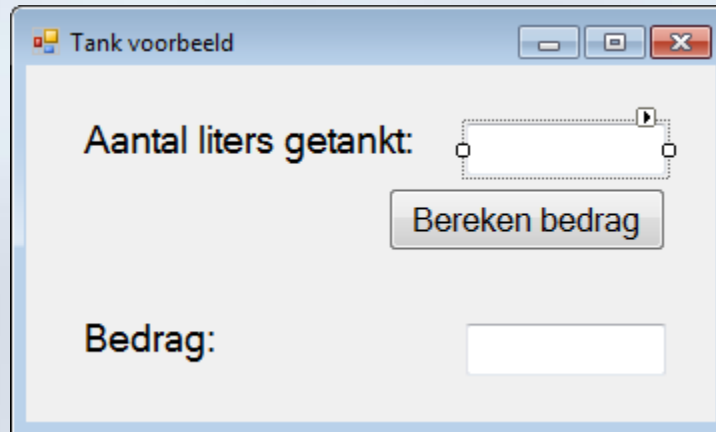
        // 1. lees aantal liters getankt
        Console.Write("Geef het aantal liters getankt: ");
        invoer = Console.ReadLine();

        // zet string om naar double (datatype conversie)
        aantalLiters = double.Parse(invoer);

        // ...
    }
}
```

Voorbeeld – Windows Forms Appl. (1/5)

- *Bereken het bedrag voor getankte benzine.* (Windows Forms)

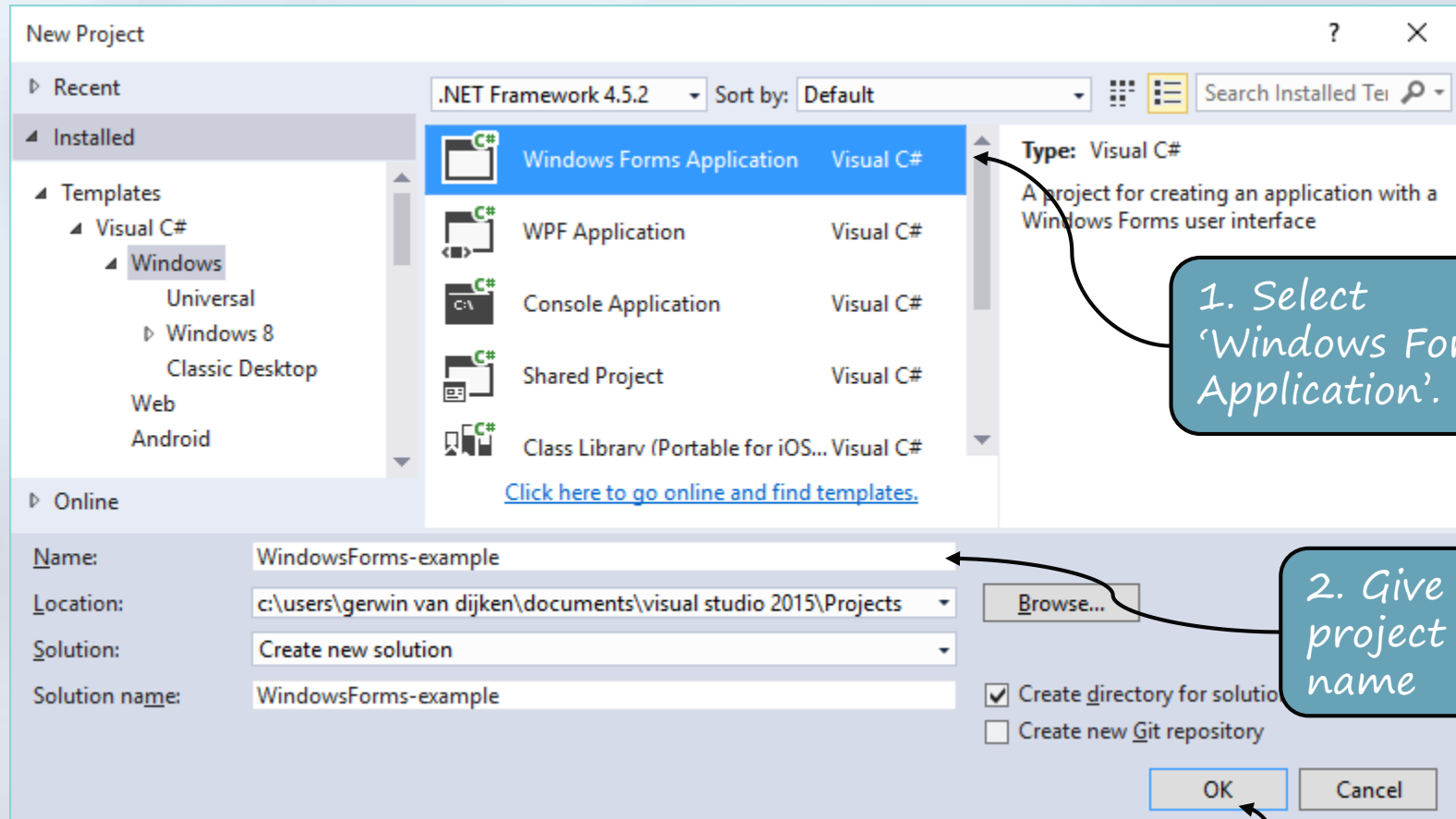


The screenshot shows a Windows Forms application window titled "Tank voorbeeld". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- A label "Aantal liters getankt:" followed by a text box with a dotted border and a small arrow button on the right.
- A button labeled "Bereken bedrag" positioned below the text box.
- A label "Bedrag:" followed by an empty text box.

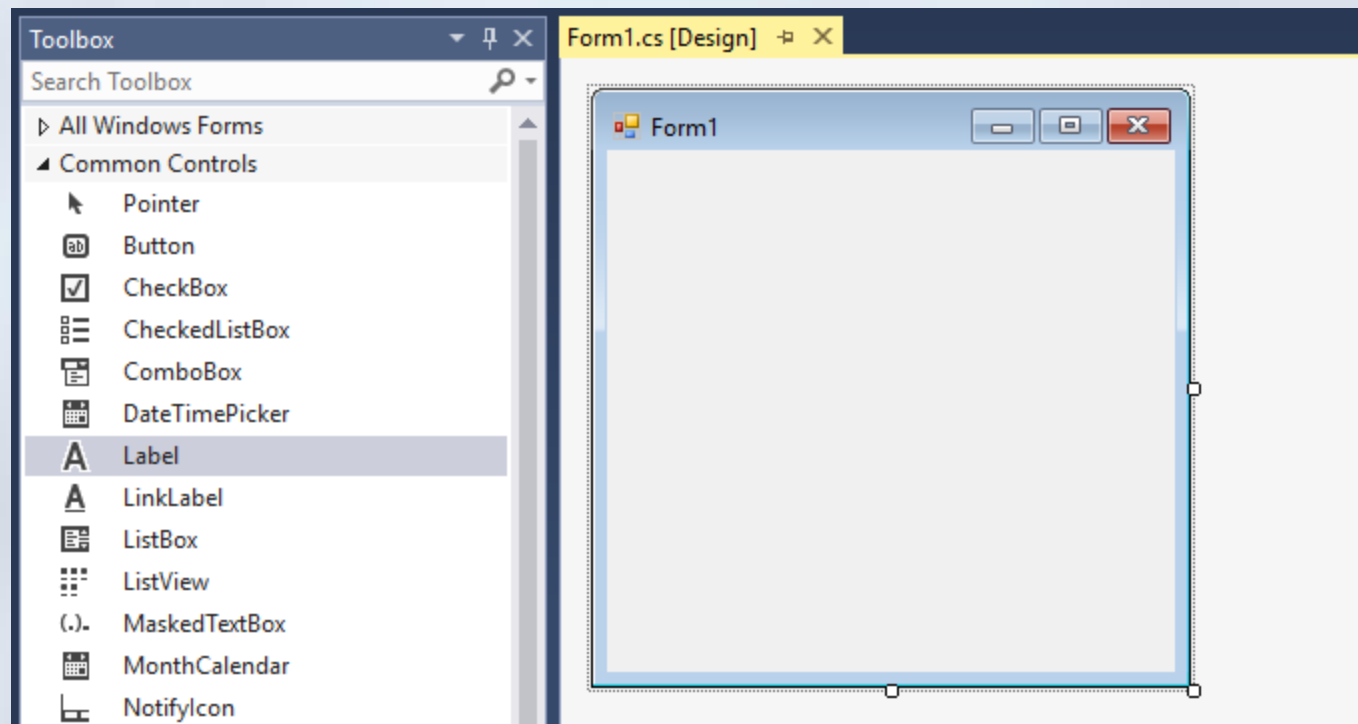
Voorbeeld – Windows Forms Appl. (2/5)

- Create a new project via (menu) File | New | Project...



Voorbeeld – Windows Forms Appl. (3/5)

- Een lege Form wordt aangemaakt, jouw taak om (userinterface) controls toe te voegen



Voorbeeld – Windows Forms Appl. (4/5)

```
public partial class Form1 : Form
{
    const double LITERPRIJS = 1.78;

    private void btnBerekenBedrag_Click(object sender, EventArgs e)
    {
        // 1. lees aantal liters getankt
        string invoer = txtAantalLiters.Text;

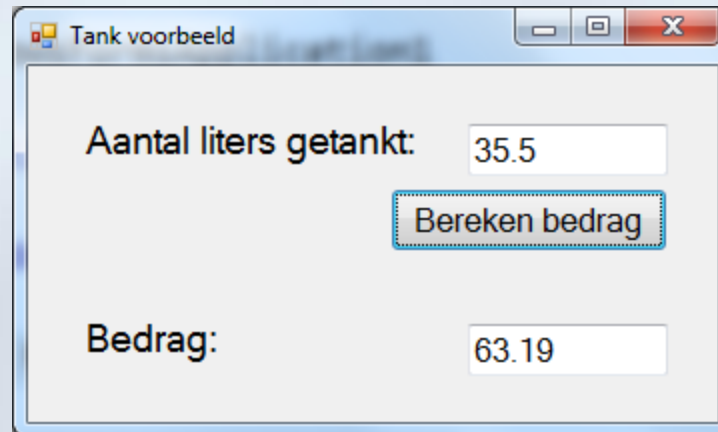
        // zet string om naar double (datatype conversie)
        double aantalLiters = double.Parse(invoer);

        // 2. bereken het bedrag
        double bedrag = aantalLiters * LITERPRIJS;

        // 3. toon het bedrag
        txtBedrag.Text = bedrag.ToString("0.00");
    }
}
```

Als we 2 cijfers
achter de
komma willen.

Voorbeeld – Windows Forms Appl. (5/5)



The screenshot shows a Windows Forms application window titled "Tank voorbeeld". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main content area is light beige and contains the following elements:

- A label "Aantal liters getankt:" followed by a text box containing the value "35.5".
- A button labeled "Bereken bedrag" with a dotted border, positioned below the first text box.
- A label "Bedrag:" followed by a text box containing the value "63.19".

Opdracht 'btw'

- Lees een prijs in. Van deze prijs wordt de btw (21%) bepaald. Afgedrukt wordt de prijs, de btw en de totaalprijs (prijs + btw).

Opdracht 'btw' – pseudocode

BTW = 0.21

lees prijs

btw = prijs * BTW

totaalPrijs = prijs + btw

toon prijs, btw, totaalPrijs

Opdracht 'btw' – PSD

btw

BTW \leftarrow 0.21
lees prijs
btw \leftarrow prijs * BTW
totaal_prijs \leftarrow prijs + btw
toon prijs, btw, totaal_prijs

Opdracht 'lekkage'

- *Een kraan lekt een bepaalde hoeveelheid cc water per uur. Hoeveel liter water gaat verloren in 1 week?*
- Analyseer het probleem en schrijf de pseudocode.

Opdracht 'lekkage' – pseudocode

UREN_IN_WEEK = 168

CC_IN_LITER = 1000

lees cc_per_uur

cc_per_week = cc_per_uur * UREN_IN_WEEK

liters_per_week = cc_per_week / CC_IN_LITER

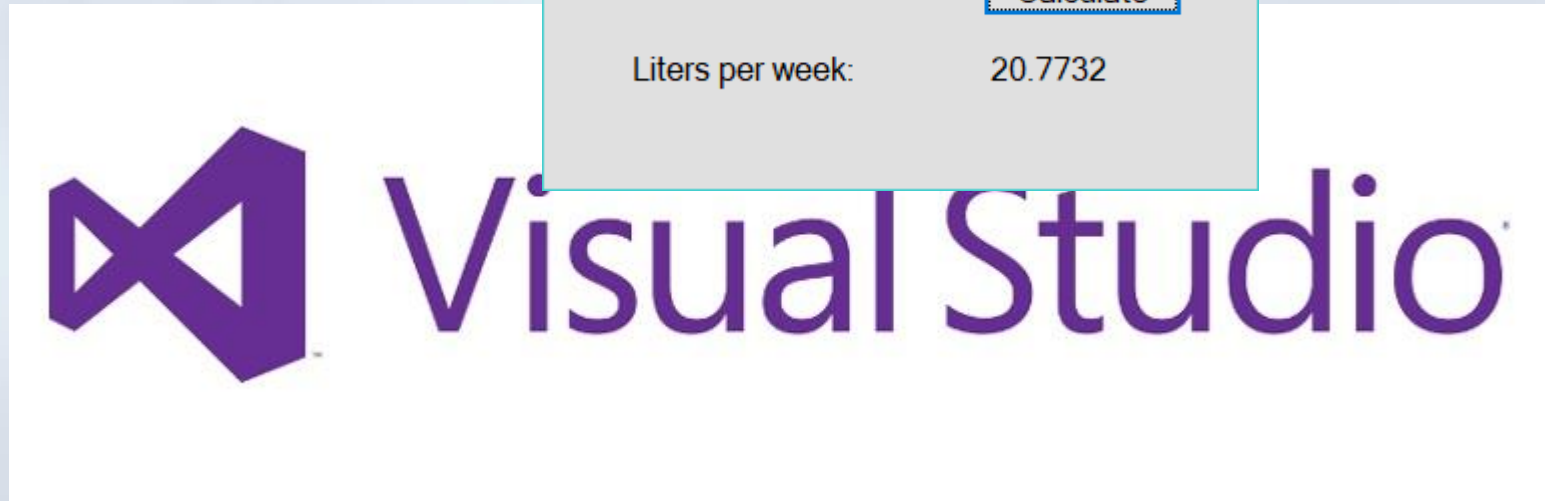
toon liters_per_week

Opdracht 'lekkage' - PSD

lekkage

UREN_IN_WEEK \leftarrow 168
CC_IN_LITER \leftarrow 1000
lees cc_per_uur
cc_per_week \leftarrow cc_per_uur * UREN_IN_WEEK
liters_per_week \leftarrow cc_per_week / CC_IN_LITER
toon liters_per_week

Visual Studio - demo



(maken van een Windows Forms applicatie)

Huiswerk voor volgende week

- **Lezen** *(voor volgende week)*
 - Yellow Book → zie Moodle
- **(praktijk) Programmeren 1** *(deze week)*
 - week 2 opdrachten