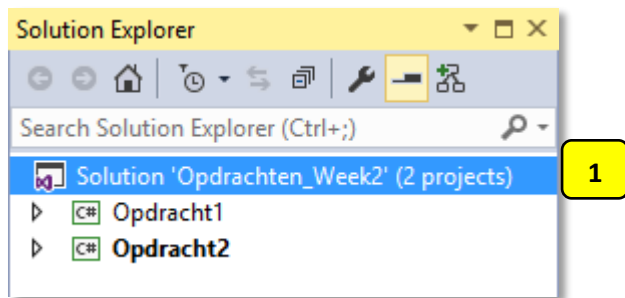


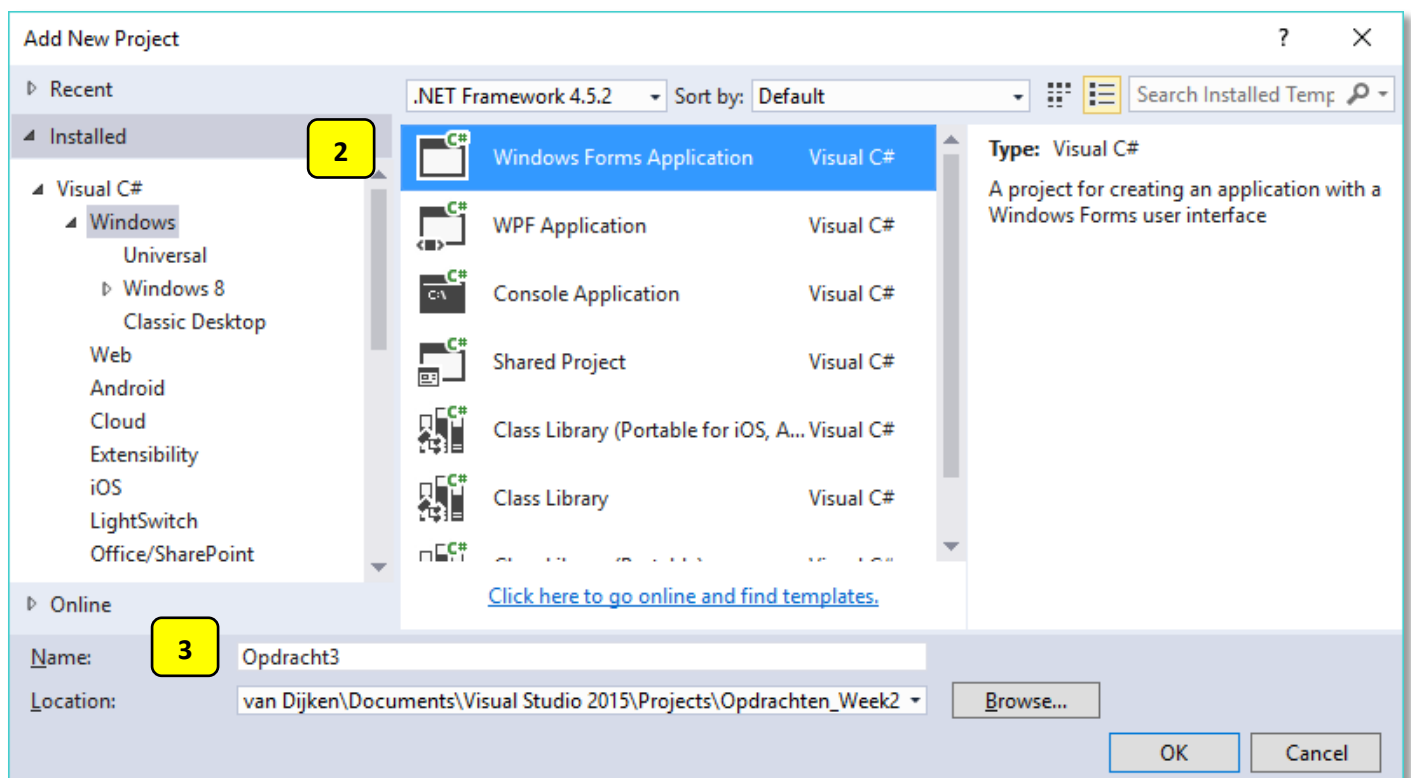
Windows Forms applicatie aanmaken

1. In de 'Solution Explorer' klik (met rechtermuisknop) op de solution en selecteer Add | New Project...;

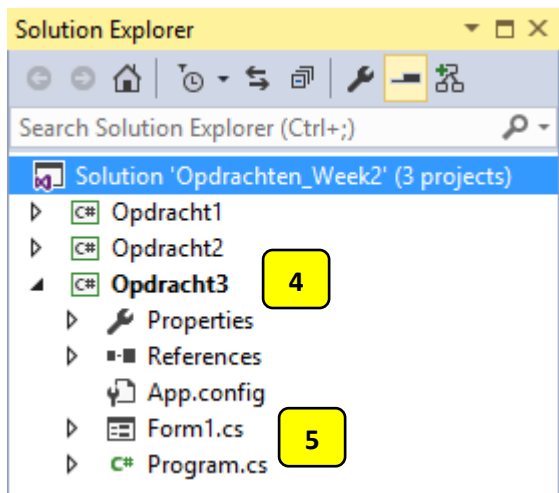


2. Selecteer in het menu Visual C# | Windows en vervolgens in het middenscherf "Windows Forms Application";

3. Geef het nieuwe project een naam (bv "Opdracht3");



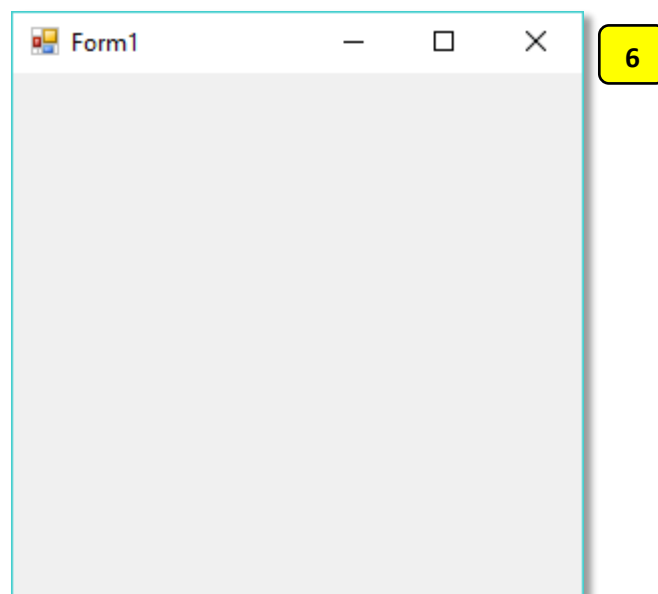
4. In de Solution is nu een nieuw project (Opdracht 3) toegevoegd;



5. Bestand “Form1.cs” bevat het hoofdformulier v/d applicatie. Het hoofdformulier is dus al aangemaakt, en kan worden aangepast door de ontwikkelaar, zowel het design (userinterface controls) als de achterliggende C#-code.

Bestand Program.cs bevat de code om het formulier (Form1) aan te maken en te runnen; dit bestand hoeft u normaliter niet aan te passen.

6. Als de Windows Forms applicatie wordt gestart (met F5 of via de Start-knop in het menu) dan krijg je het standaard formulier te zien, zie hieronder.

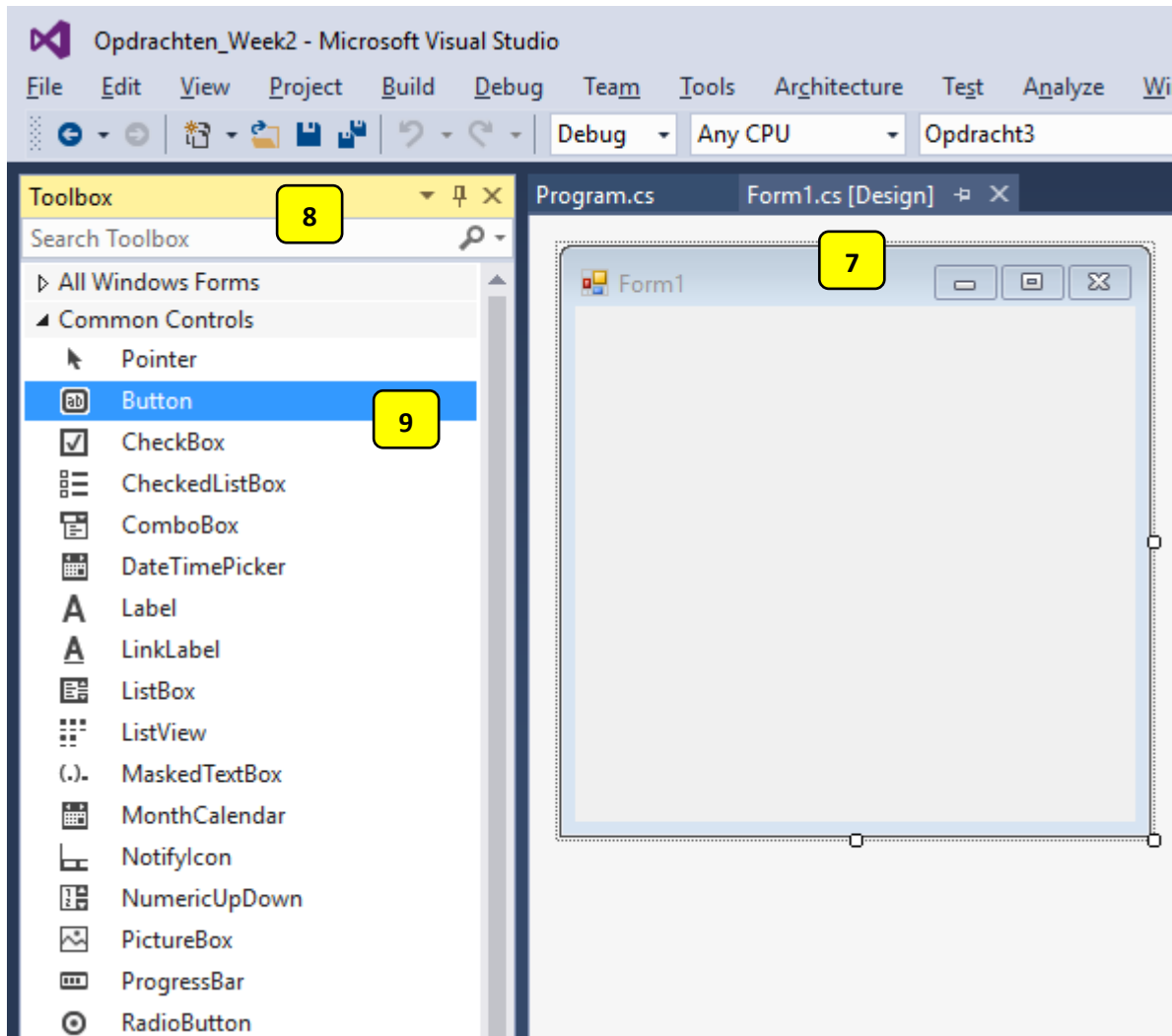


Design aanpassen van een Windows Forms applicatie

In Visual Studio kun je nu het standaard aangemaakte hoofdformulier aanpassen, door er userinterface controls (zoals een button of een label) op te plaatsen/slepen.

7. Dubbelklik bestand "Form1.cs" (in de Solution Explorer) zodat het hoofdformulier in de Designer wordt getoond.

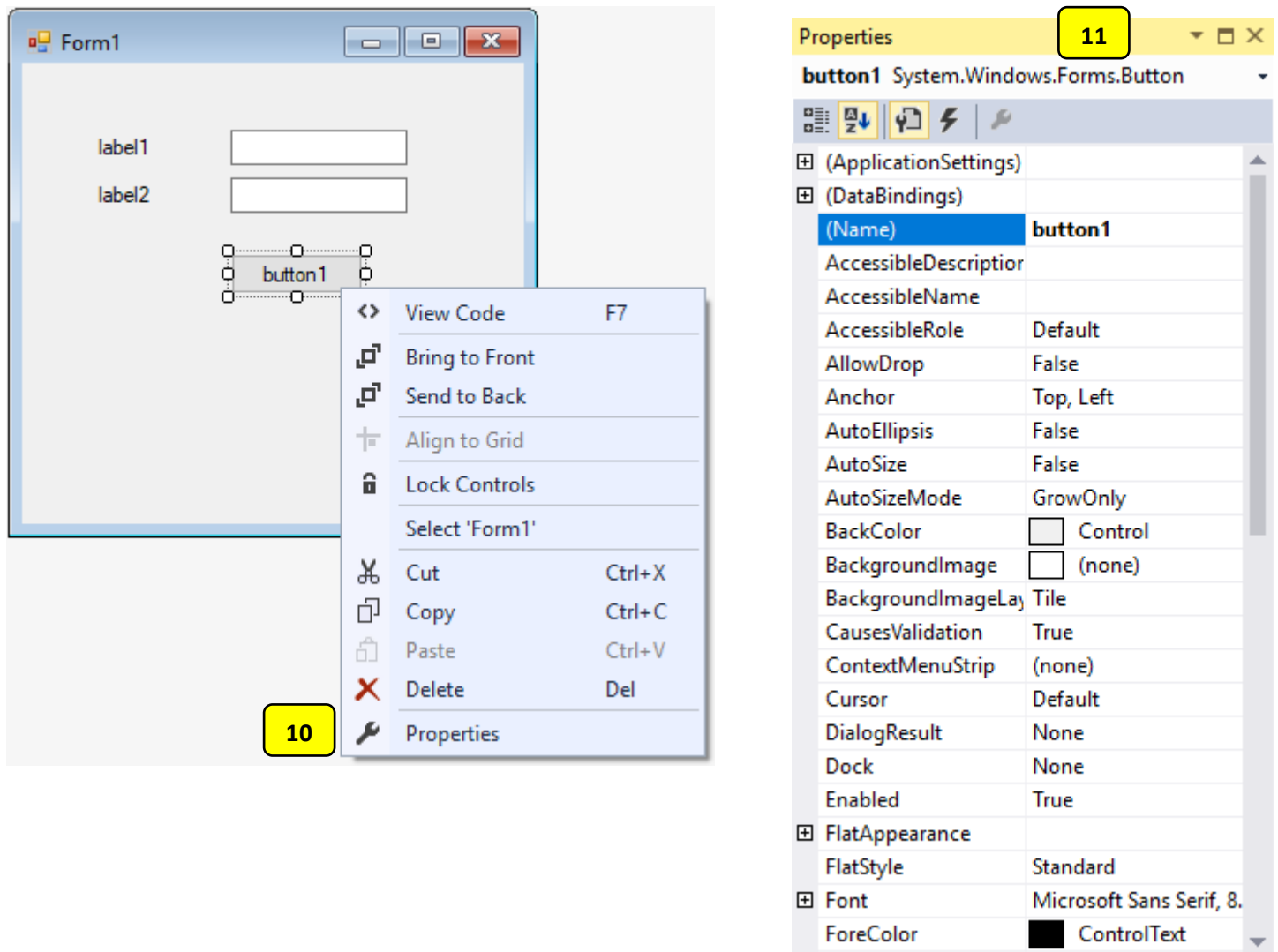
8. Zorg er vervolgens voor dat de Toolbox zichtbaar is (via menu View | Toolbox of via CTRL|ALT|X).



9. Je kunt nu bv een Button op het formulier slepen en andere controls. We zullen in de eerste periode voornamelijk werken met Button, Label en TextBox.

10. Zodra een userinterface control op het scherm is geplaatst, kun je de properties (eigenschappen) hiervan aanpassen. Dat doe je door de control (bv “button 1”, zie hieronder) aan te klikken, en met de rechtermuisknop item “Properties” te selecteren.

11. Aan de rechterkant (normaliter onder de Solution Explorer) zie je nu een Properties gedeelte.



Je kunt nu o.a. de naam van de (userinterface) control aanpassen via Property “Name”). Het is belangrijk dat elke control een betekenisvolle naam krijgt want deze naam wordt ook gebruikt in de C# code v/h formulier (zie later).

Het is een goede gewoonte om een vaste prefix te gebruiken bij de naam van een control; gebruik prefix “btn” voor buttons (bv “btnBereken”), “lbl” voor labels (bv “lblNaam”) en “txt” voor textboxes (bv “txtVoornaam”). Hierdoor maak je minder fouten bij het gebruiken van de userinterface controls in de achterliggende C# code (zie later).

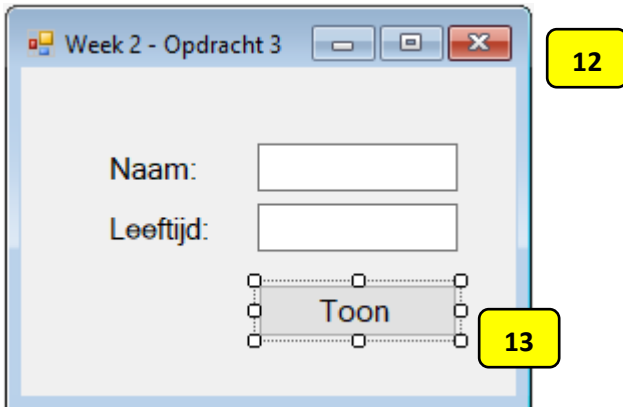
Je kunt uiteraard via de properties ook vele andere eigenschappen aanpassen, zoals de tekst die getoond moet worden aanpassen (property “Text”) en de fontgrootte (via property “Font”).

Het is verder een goede gewoonte om de titel van het formulier aan te passen; deze staat standaard op “Form1”, maar het is natuurlijk beter om daar een duidelijke titel voor te gebruiken (voor de wekelijkse opdrachten van Programmeren 1 kun je bv “Week 2 – Opdracht 3” gebruiken). Selecteer het formulier (door ergens op het formulier te klikken) en pas de property “Text” aan.

Achterliggende code van een Windows Forms applicatie

12. Nu de properties van het hoofdformulier zijn aangepast (o.a tekst van het hoofdformulier is ingesteld op “Week 2 - Opdracht 3” en de controls hebben zinvolle namen gekregen: lblNaam, txtNaam, lblLeeftijd, txtLeeftijd, btnToonInfo) kunnen we de achterliggende C# code schrijven. Deze achterliggende code wordt vaak “Code Behind” genoemd.

13. Er zijn verschillende manieren om naar de Code Behind te gaan, o.a. via F7 of door op een control te dubbelklikken. Dit laatste zorgt er voor dat er een zogenaamde *eventhandler* wordt aangemaakt voor de dubbelgeklikte control. Een eventhandler bevat (C#) code die uitgevoerd wordt zodra het *event* (gebeurtenis) optreedt, zoals het klikken op een button. Voor buttons wil je altijd een eventhandler.



14. In onderstaande figuur zie je Code Behind (C# code) van het formulier. Er is een eventhandler voor de button (“btnToonInfo_Click”) die uitgevoerd wordt zodra de gebruiker van de applicatie op de knop “Toon” klikt.

```
11 namespace Opdracht3
12 {
13     public partial class frmMainform : Form
14     {
15         public frmMainform()
16         {
17             InitializeComponent();
18         }
19
20         private void btnToonInfo_Click(object sender, EventArgs e)
21         {
22             // haal de gegevens op uit de userinterface controls
23             string naam = txtNaam.Text;
24             string invoer = txtLeeftijd.Text;
25             int leeftijd = int.Parse(invoer);
26
27             // toon informatie
28             MessageBox.Show("Hallo " + naam + ", je bent " + leeftijd + " jaar oud!", "Info");
29         }
30     }
31 }
```

De eventhandler zorgt er voor dat de ingetikte gegevens uit de userinterface controls wordt gehaald (let er op dat de property “Text” wordt gebruikt, bv txtNaam.Text) en dat er een *MessageBox* wordt getoond.

Week 2 - Opdracht 3

Naam:

Leeftijd:

Info

Hallo Kevin, je bent 34 jaar oud!