
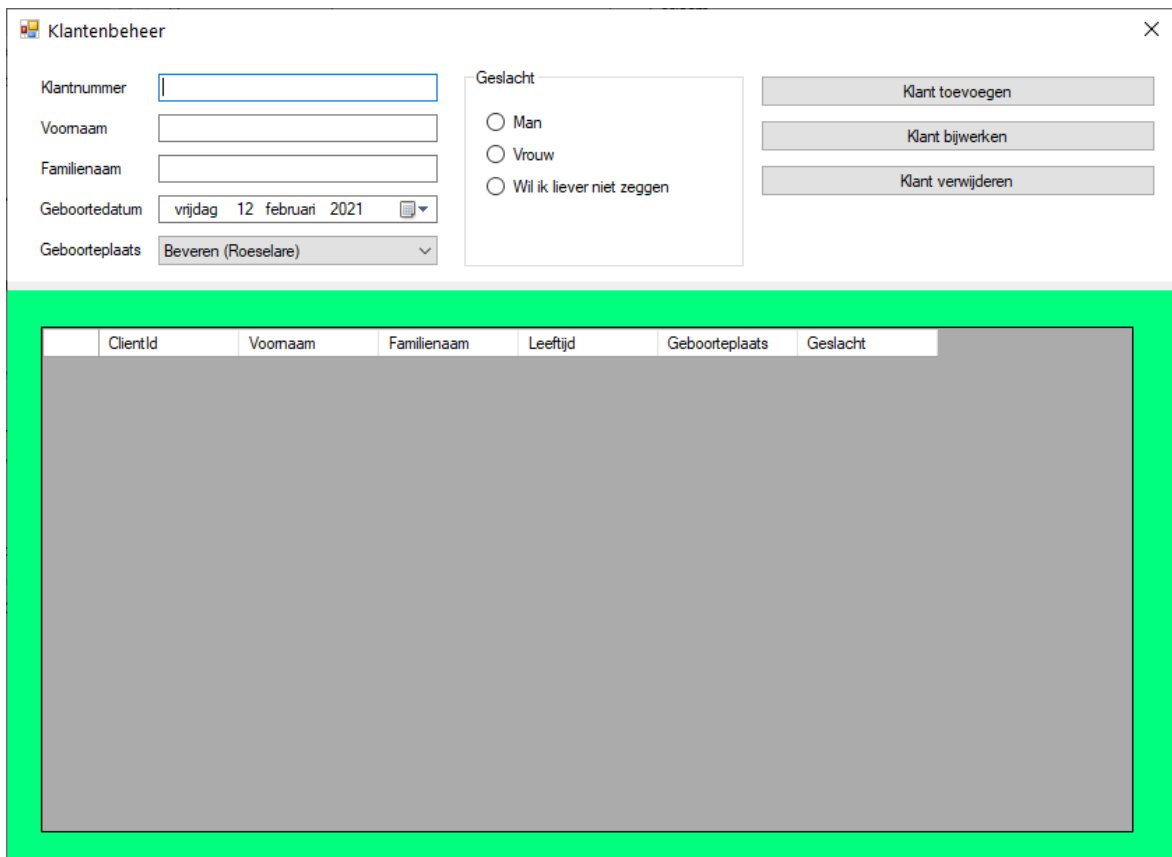










# Type Conversion Exercise \*

## 1 Basis

-  Maak een nieuwe Windows Forms applicatie (.NET Framework)
-  Wijzig de naam van je Form naar frmClients
-  Wijzig de tekst in de titelbalk van je Form naar Klantenbeheer
-  Bouw onderstaande lay-out na, kijk goed naar alle elementen. Elk van de elementen is te vinden in de Toolbox in Visual Studio
  - het 'kader' rond de radio buttons is een 'GroupBox'.
  - de groene achtergrondkleur onderaan is een 'Panel'
  - het element dat onderaan staat is een 'DataGridView' (de kolommen kun je al toevoegen via 'Add Column' en de Adding/Editing/Deleting opties moeten **uitgeschakeld** staan)
-  Geef ALLE elementen ook een goede naam (ja ook labels enz...namen ALTIJD in het Engels). Aan de lijst met geboorteplaatsen voeg je 10 steden en/of gemeenten toe (je mag zelf kiezen welke).



-  Verwijder de niet gebruikte using statements uit je frmClients class
-  Voeg aan de code van je Form, onder de constructor (public frmClients()), een methode **Init** toe. Deze methode gebruiken we om de standaardwaarden van onze Form in te stellen. Roep de methode op vanuit de constructor van de Form. De **Init** methode heeft geen return value en geen parameters.
-  In de **Init** methode gaan we een standaardwaarde instellen voor de keuzelijst van geboorteplaatsen: stel de SelectedIndex property van je keuzelijst in op 0. 0 is het eerste Item in de keuzelijst (zero-based). Stel eveneens in dat standaard de radiobutton “Wil ik liever niet zeggen” is geselecteerd.
-  Voeg een InputValidation methode toe. De return value is of alle input geldig is of niet (bool). De methode krijgt 6 parameters:
  - string clientId
  - string firstName
  - string lastName
  - DateTime birthDate
  - string birthPlace
  - bool isMale
  - bool isFemale
  - bool isNotSaying
-  Aangezien we in de InputValidation method een (bool) waarde terugsturen; maak je een bool variabele aan. Geef deze als naam **isValid**. Deze waarde zal bijhouden of alle velden correct werden ingevuld of niet. Standaard is de bool = true (valid dus). Deze variabele zal je ook moeten returnen anders krijg je een “not all code paths return a value” foutmelding.
-  Voeg aan de knop **Klanten toevoegen** een Click event toe. Vanuit dat Click event roep je de InputValidation methode op. Geef alle correcte velden als parameters door aan de InputValidation methode. Vang ook de return van de InputValidation methode op!
-  Alle velden zijn verplicht in te vullen. Schrijf een **if** statement die controleert of alle velden werden ingevuld. De velden die al een standaardwaarde hebben hoeft je niet te controleren aangezien deze nooit leeg kunnen zijn.  
Tip: string.IsNullOrEmpty()  
Als alle velden zijn ingevuld hoeft er voorlopig niets te gebeuren (dat doen we later). Als één van de velden niet is ingevuld dan plaats je isValid op false.
-  Voeg in je **if** statement (wanneer alle velden correct zijn ingevuld) volgende variabelen toe:
  - int id
  - int age
  - string gender

- Voeg een methode toe `ConvertClientId`. De return van de methode is een integer (aangezien we zopas een “int clientId” hebben gemaakt). Aan deze methode zullen we het clientId doorgeven als string, maak hiervoor een parameter.
- Roep de methode op door in je `InputValidation` `int id = ConvertClientId(clientId);` te plaatsen.
- Tijd om code toe te voegen aan de `ConvertClientId` methode. In deze methode zetten we het clientId om van een string naar een integer. We moeten wel eerst testen of dat wel kan: voeg een try...catch blok toe aan de methode. In de try converteer je het clientId van een string naar int en return je deze integer waarde.

In de catch kiezen we ervoor om als waarde -1 te returnen (indien omzetten faalt zullen we dus als clientId -1 krijgen).

- Voeg een nieuwe methode `CalculateAge` toe. Deze methode stuurt de leeftijd van de klant terug als cijfer. We moeten de geboortedatum doorgeven aan deze methode om dus de leeftijd terug te krijgen.

Voeg aan deze methode code toe die de leeftijd uitrekent en terugstuurt. Als de leeftijd niet kan uitgerekend worden stuur je -1 terug.

Om de leeftijd te berekenen kan je het geboortjaar aftrekken van het huidige jaar. Je moet dan wel nog via een if controleren of de huidige `DayOfYear` (1-366) kleiner is dan de geboortedatum `DayOfYear`...want dan is de persoon nog niet jarig geweest en moet de leeftijd nog met 1 worden verminderd

➔ `DateTime.Now.DayOfYear < birthDate.DayOfYear` ➔ age verlagen met 1

- Wijzig de `InputValidation` methode zodat de age variabele ingesteld is op het resultaat van de `CalculateAge` methode.
- Voeg een `GetGender` methode toe. Deze methode gebruikt de 3 waarden (`isMale`, `isFemale`, `isNotSaying`) om de tekstuele waarde terug te sturen. Dus als `isMale` is geselecteerd dan stuurt deze methode “Man” terug. (Man, Vrouw, Onbekend)
- Roep op de correcte plaats in de `InputValidation` methode de `GetGender` methode op en geef de 3 waarden door.
- Pas de methode `InputValidation` verder aan. Als `iClientId` geldig is (groter dan -1) en de age is geldig (groter dan 0) en de gender niet leeg is. Plaats dan `isValid` op true. Dan is alles geldig en kan de klant later worden opgeslagen. Zoniet, plaats `isValid` op false.
- Voeg een (void) methode `AddClient` toe. We geven volgende parameters door aan deze methode:
  - int clientId
  - string firstName
  - string lastName
  - int age

- string birthplace
- string gender

- ☞ Roep vanuit het Click event van de Add button de AddClient methode op. Deze methode mag enkel opgeroepen worden als het resultaat van de InputValidation true is (en dus alle velden geldig zijn ingevuld).
- ☞ Toon een MessageBox wanneer een klant succesvol werd toegevoegd. Doe dit vanuit het click event van de Add button. Deze messagebox bevat een OK knop en als icoon het "information" icon (zoek via google eens informatie op over "C# messagebox").
- ☞ Nu voegen we code toe aan de AddClient methode. Het enige wat deze methode moet doen is de waarden die binnenkomen via de parameters toevoegen als rij in de DataGridView van je form. Doe dit met een try catch...als iets fout gaat return je false.

Tip: `DataGridViewName.Rows.Add(allesWaardenHierMetKommaGescheiden);`  
(je moet geen conversies doen, gewoon de parameters van AddClient gebruiken)



- ☞ Voeg een Click event toe aan de knop "Klant verwijderen"
- ☞ Voeg een methode toe met als naam DeleteClients. Deze methode geeft als return of de klanten werden gewist of niet (bool). Ze bevat geen parameters.
- ☞ Roep de methode DeleteClients op vanuit het Click event
- ☞ In de DeleteClient wis je de door de gebruiker geselecteerde rij uit de DataGridView. Je kan hiervoor `DataGridViewName.SelectedRows` gebruiken. Merk op: `SelectedRows` ...

Tip: `DataGridViewName.Rows.RemoveAt(row.Index);`

Tip 2: het verwijderen zal pas lukken als de **volledige rij** geselecteerd staat (klikken aan de linkerzijde om de rij te selecteren)

## 2 Uitbreiding

- ☞ Pas je knop Klant Verwijderen aan zodat eerst een MessageBox wordt getoond waarin het verwijderen moet bevestigd worden. Als het resultaat "Ja" is dan zal pas gewist worden. Deze MessageBox verschijnt enkel als er rijen in de DataGridView geselecteerd zijn. (Tip: `DialogResult / MessageBoxButtons.YesNo`)
- ☞ Pas je InputValidation methode aan zodat geen 2 klanten met hetzelfde Id kunnen worden toegevoegd. Maak hiervoor een methode ValidClientId. (Tip: van een 'row' kun je ook de `.Cells[ ]` opvragen waarbij je tussen de vierkante haakjes vermeldt de hoeveelste cel uit de rij je wil raadplegen of gebruiken)
- ☞ Pas je DataGridView aan zodat je altijd automatisch de volledige rij selecteert (is één van de properties)

-  Zorg ervoor dat als je een bestaande klant selecteert (de rij aanklikt = SelectionChanged event), de gegevens van die klant in de invulvelden bovenaan komen te staan. Wijzig je iets aan die velden en druk je op bijwerken, dan wordt die specifieke klant aangepast.  
(Tip: SelectionChanged event)  
(Tip2: je voegt best een verborgen kolom toe met de geboortedatum, hiervoor zul je ook wel nog andere wijzigingen moeten aanbrengen in je code...)
-  Zorg dat je na het toevoegen van een nieuwe klant ook altijd de rij van deze klant actief (geselecteerd) zet (enkel deze rij)