



GRADUAAT IN HET
PROGRAMMEREN

Visual C#.NET
Oefeningenbundel

Cursus

C# Advanced

Opleidingsonderdeel

Informatica | Programmeren

Afstudeerrichting

Paul Dox
Patricia Briers

Auteurs



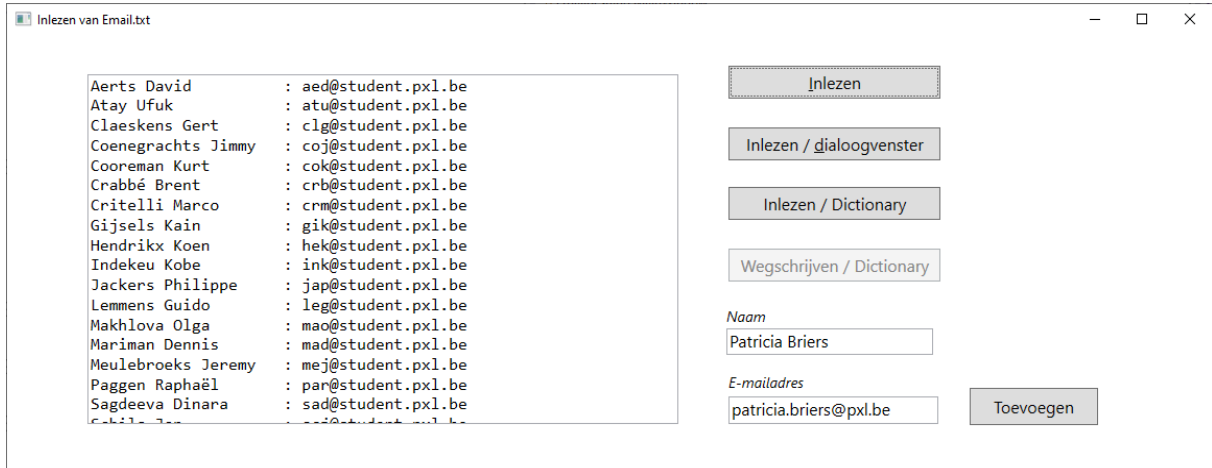
INHOUD

TEKSTBESTANDEN	3
TOEPASSING 1 : EMAILBESTAND	3
TOEPASSING 2 : STRINGZOEKENBESTAND	5
TOEPASSING 3 : PUNTENBESTAND	6
TOEPASSING 4 : ZONNEPANEELMETINGEN	8
MEERDERE VENSTERS	12
TOEPASSING 5 : WOORDENBOEK	12
TOEPASSING 6 : WOORDENBOEKTAB.....	15
KLASSE	17
TOEPASSING 7 : KLASSEBANKREKENING	17
TOEPASSING 8 : KLASSETELLER	18
TOEPASSING 9 : KLASSEPUNTEN	19
TOEPASSING 10 : KLASSEPERSONEEL	21
TOEPASSING 11 : KLASSEHUIS.....	24
TOEPASSING 12 : KLASSEZONNEPANEELMETINGEN.....	27
TOEPASSING 13 : KLASSESPORTKAMPLEDEN	28
TOEPASSING 14 : KLASSESPORTKAMPLEDEN2	32
TOEPASSING 15 : KLASSESTUDENTENVERWERKING.....	35
TOEPASSING 16 : KLASSEOVERERVINGPERSOON.....	41
TOEPASSING 17 : ABANKREKENING.....	47
TOEPASSING 18 : AWERKNEMER.....	49
TOEPASSING 19 : ILIST.....	52
TOEPASSING 20 : IOUDERDOM	54
TOEPASSING 21 : IBANKREKENING	56
TOEPASSING 22 : VALIDATIELIB (CLASS LIBRARY).....	58
DATABANKEN	60
TOEPASSING 23 : DATASET MEDEWERKERS	60
TOEPASSING 24 : DATASET AUTO	62
TOEPASSING 25 : DB SPIONSHOP1.....	63
TOEPASSING 26 : DB SPIONSHOP2.....	64
TOEPASSING 27 : DB SPIONSHOP 3	65
TOEPASSING 28 : DB MEDEWERKER	67
TOEPASSING 29 : DB MEDEWERKERBEHEER	69

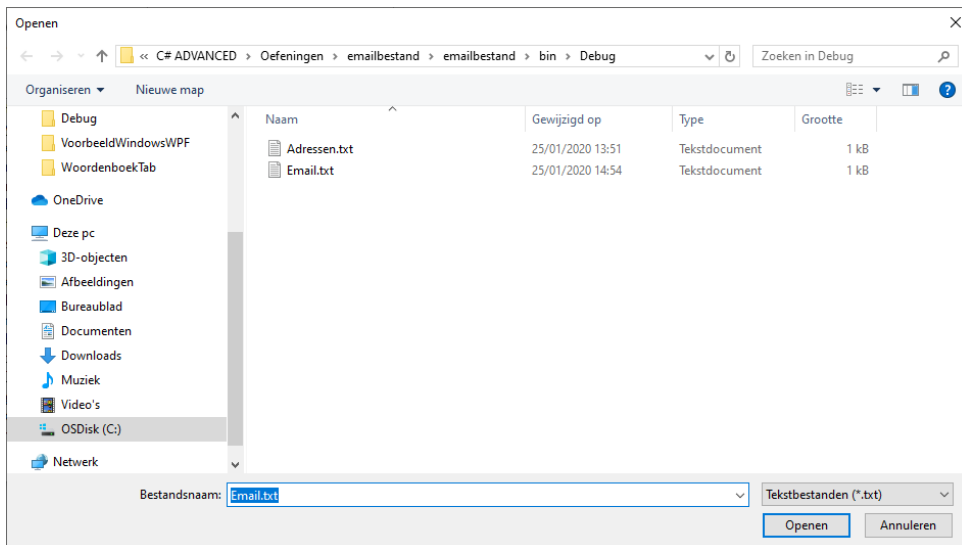
Tekstbestanden

Toepassing 1 : EmailBestand

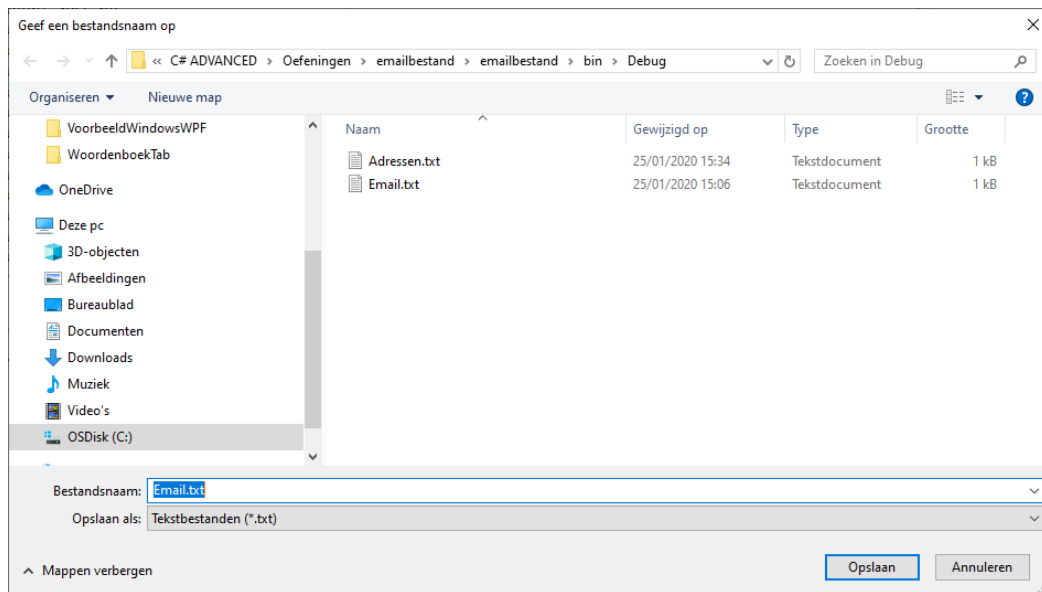
Lees uit het tekstbestand *Email.txt* de namen en de e-mailadressen en druk af in *txtResultaat*. Zorg voor een correcte lay-out. Voorzie een gestructureerde foutafhandeling wanneer het bestand niet gevonden kan worden en wanneer er zich een onverwachte fout kan voordoen.



- De opdrachtknop **Inlezen** leest het bestand in een *StringBuilder* en drukt het af in het linker tekstvak *TxtResultaat*. Gebruik hiervoor de functie *StringBuilder InlezenBestand(string bestandsnaam)*.
- De opdrachtknop **Inlezen/Dialoogvenster** maakt gebruik van *OpenFileDialog* om het juiste bestand te openen om in te lezen met de functie *InlezenBestand()*.



- Met de opdrachtknop **Inlezen/Dictionary** wordt het bestand ingelezen in een dictionary *Dictionary<string, string> dicGeg* en afgedrukt in *TxtResultaat*.
- De opdrachtknop **Wegschrijven / Dictionary** gebruikt de dictionary *dicGeg* om de e-mailadressen weg te schrijven naar een nieuw bestand *Adressen.txt*.
- De opdrachtknop **Toevoegen** zorgt ervoor dat de ingevoerde gegevens toegevoegd worden aan het bestand *Email.txt*. Gebruik hier een *SaveFileDialog* voor.

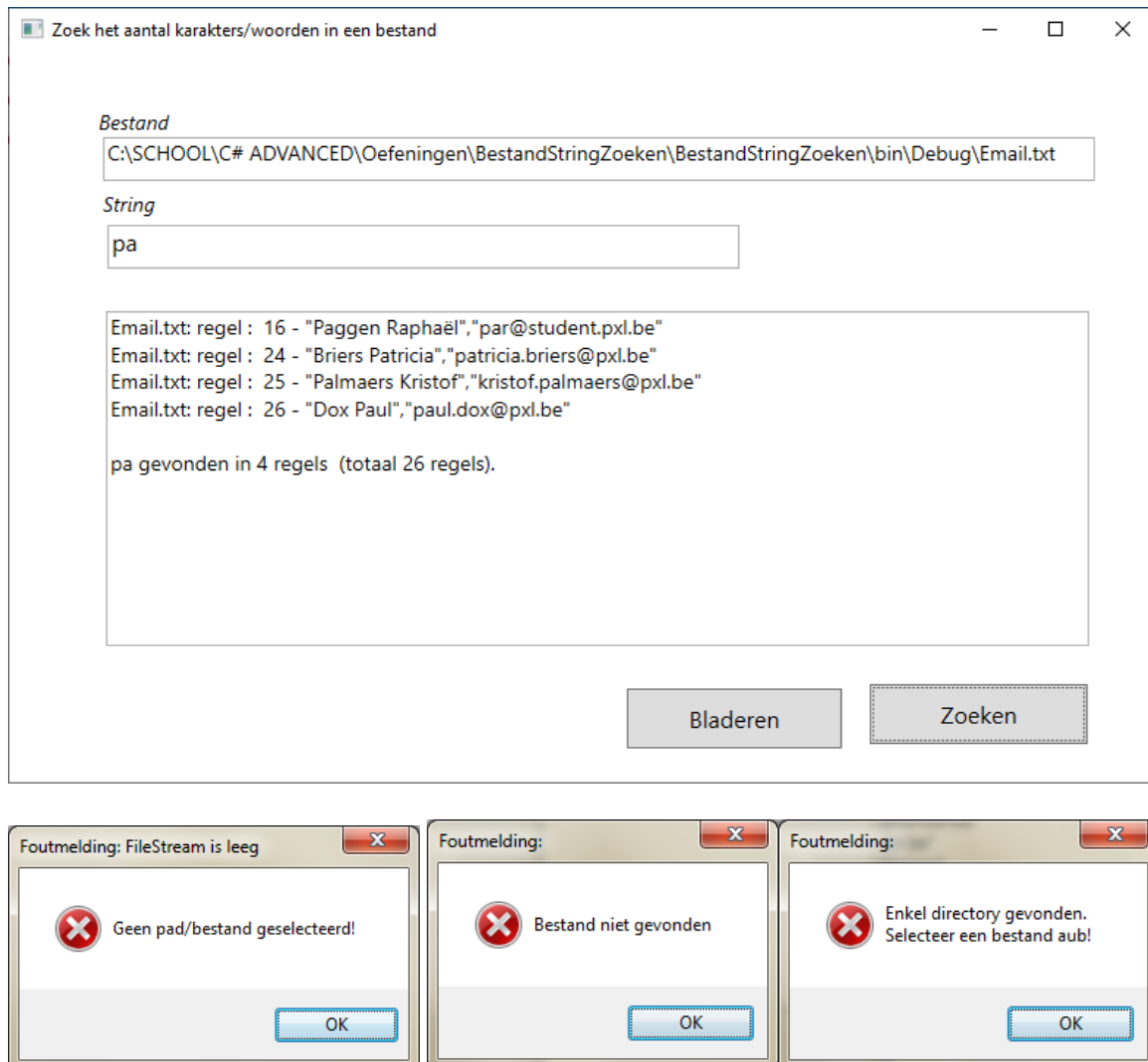


Toepassing 2 : StringZoekenBestand

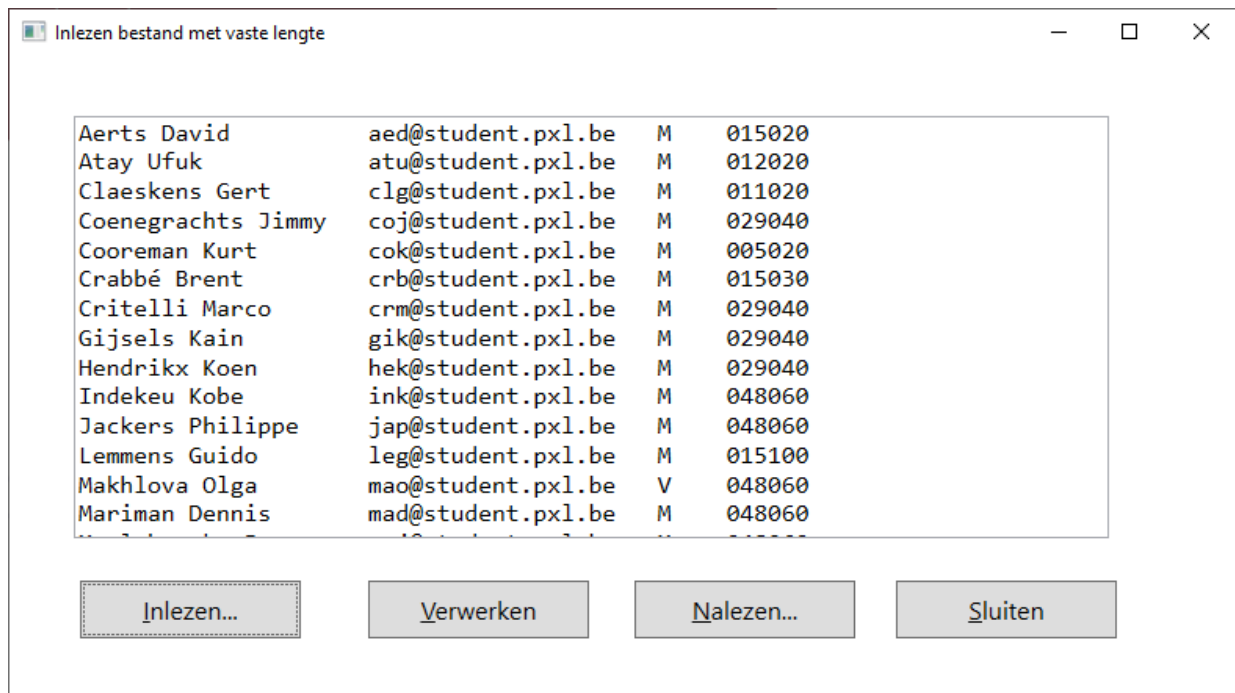
Open het tekstbestand *Email.txt* via het dialoogvenster *OpenFileDialog* en zoek in hoeveel regels een karakter(s) voorkomen in het tekstbestand. Voorzie een gestructureerde foutafhandeling:

- wanneer geen bestand geselecteerd is (FileStream leeg) ([ArgumentException](#))
- wanneer het bestand niet gevonden kan worden ([FileNotFoundException](#))
- enkel een map en geen bestand geselecteerd is ([UnauthorizedAccessException](#))
- wanneer er een onverwachte fout zich kan voordoen ([Exception](#)).

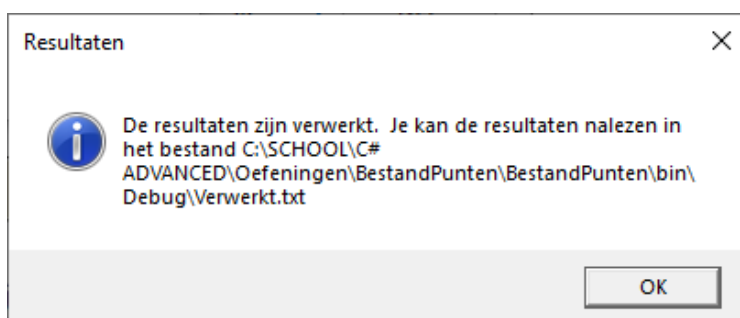
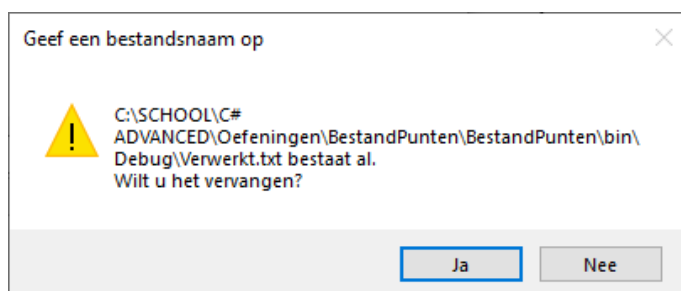
Geef bij elke soort fout een correct berichtenvenster.



Lees het tekstbestand *Punten.txt* met vaste lengte in en toon het in het tekstvak *txtResultaat*. Na het inlezen worden de opdrachtknoppen *Verwerken* en *Nalezen* zichtbaar.



Door middel van de opdrachtknop *Verwerken* moet een nieuw tekstbestand *PuntenVerwerkt.txt* met vaste lengte worden gemaakt. Het bestand bestaat uit de naam, het behaald resultaat (eerste 3 cijfers van 4^{de} kolom/laatste 3 cijfers van 4^{de} kolom) en de melding Geslaagd of Niet geslaagd. Een cursist is slechts geslaagd wanneer er minimum 85% van de punten behaald is. Bij het overschrijven van het bestand wordt de gebruiker gevraagd om dit te bevestigen. Wanneer het bestand is verwerkt en weggeschreven, verschijnt een berichtenvenster dat het bestand nagelezen kan worden.



De opdrachtknop *Nalezen* opent het dialoogvenster *OpenFileDialog* zodat het bestand *PuntenVerwerkt.txt* kan gelezen worden in het tekstvak *TxtResulaat*.



Maak een toepassing *Zonnepaneelmetingen* voor de verwerking van opbrengstmetingen van zonnepanelen. Een bezitter van zonnepanelen noteert meestal dagelijks de dagopbrengst (in Kw/u) van zijn zonnepanelen. De meting wordt in een tekstbestand *MetingenOpbrengstZonnepanelen.txt* bewaard. Dit bestand vind je op Blackboard en plaats je in de zelf toegevoegde map Bestanden. De eerste records zijn:

14/03/2019 - 11,8	23/03/2019 - 14,3
15/03/2019 - 02,5	24/03/2019 - 06,7
16/03/2019 - 09,2	25/03/2019 - 03,5
17/03/2019 - 04,6	27/03/2019 - 14,6
18/03/2019 - 09,3	28/03/2019 - 16,1
19/03/2019 - 08,4	29/03/2019 - 04,7
21/03/2019 - 02,4
22/03/2019 - 04,1	

Zoals je kan zien staat vooraan de datum van de gemeten dagopbrengst (dag/maand/jaar), daarna staat steeds “ - ”, tenslotte gevolgd door de dagopbrengst met 1 decimaal cijfer (in totaal steeds 4 posities, dus zo nodig met voorloophulpnullen).

Je mag veronderstellen dat er hierin geen fouten staan (bestaande datums en juiste metingen).

Het venster bevat volgende objecten:

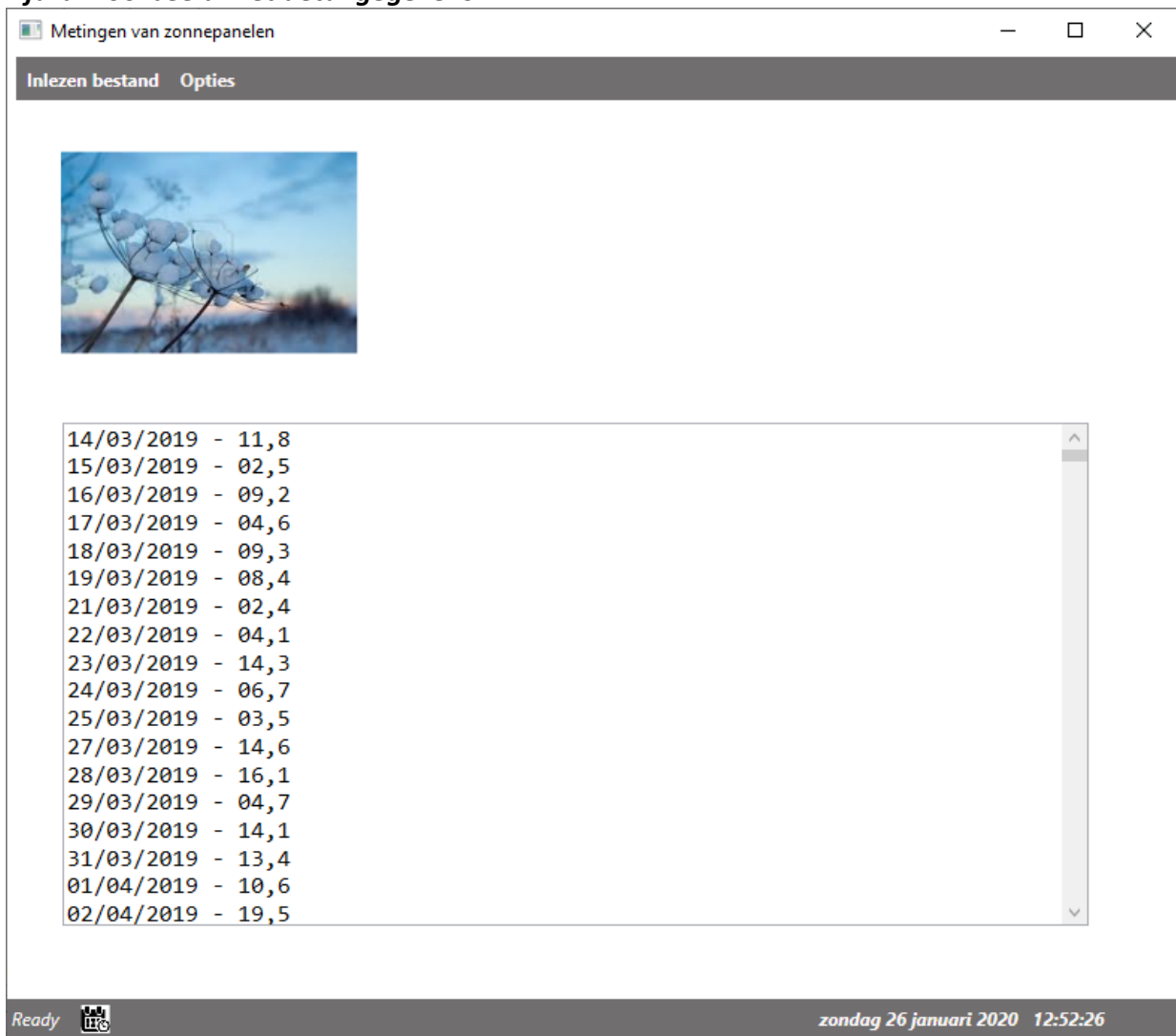
- Onderaan een statusbar met een *TextBlock* Ready, een *Image* DateTime en een *TextBlock* die de datum en tijd weergeeft (zie verder).
- Een tekstvak *TxtResultaat* om resultaten te tonen.
- Een *Image* om een afbeelding te tonen. Het getoonde beeld is afhankelijk van het seizoen waarin het programma gestart wordt (maand van de systeemdatum testen!). Voor december, januari en februari wordt “winter.jpg” getoond, voor maart, april en mei wordt “lente.jpg” getoond, voor juni, juli en augustus wordt “zomer.jpg” getoond en tenslotte wordt voor september, oktober en november “herfst.jpg” getoond. Deze 4 afbeeldingen worden je bezorgd en je plaatst ze in de map Bestanden. Je programma moet tevens testen of het betreffende bestand aanwezig is op de betreffende locatie. Indien dat niet het geval zou zijn, geef dan een passende melding: “Afbeelding winter niet aanwezig”, “Afbeelding lente niet aanwezig”, “Afbeelding zomer niet aanwezig” of “Afbeelding herfst niet aanwezig”.

De toepassing werkt menugestuurd. Voeg volgende menustructuur met 2 hoofdkeuzes toe aan je venster:

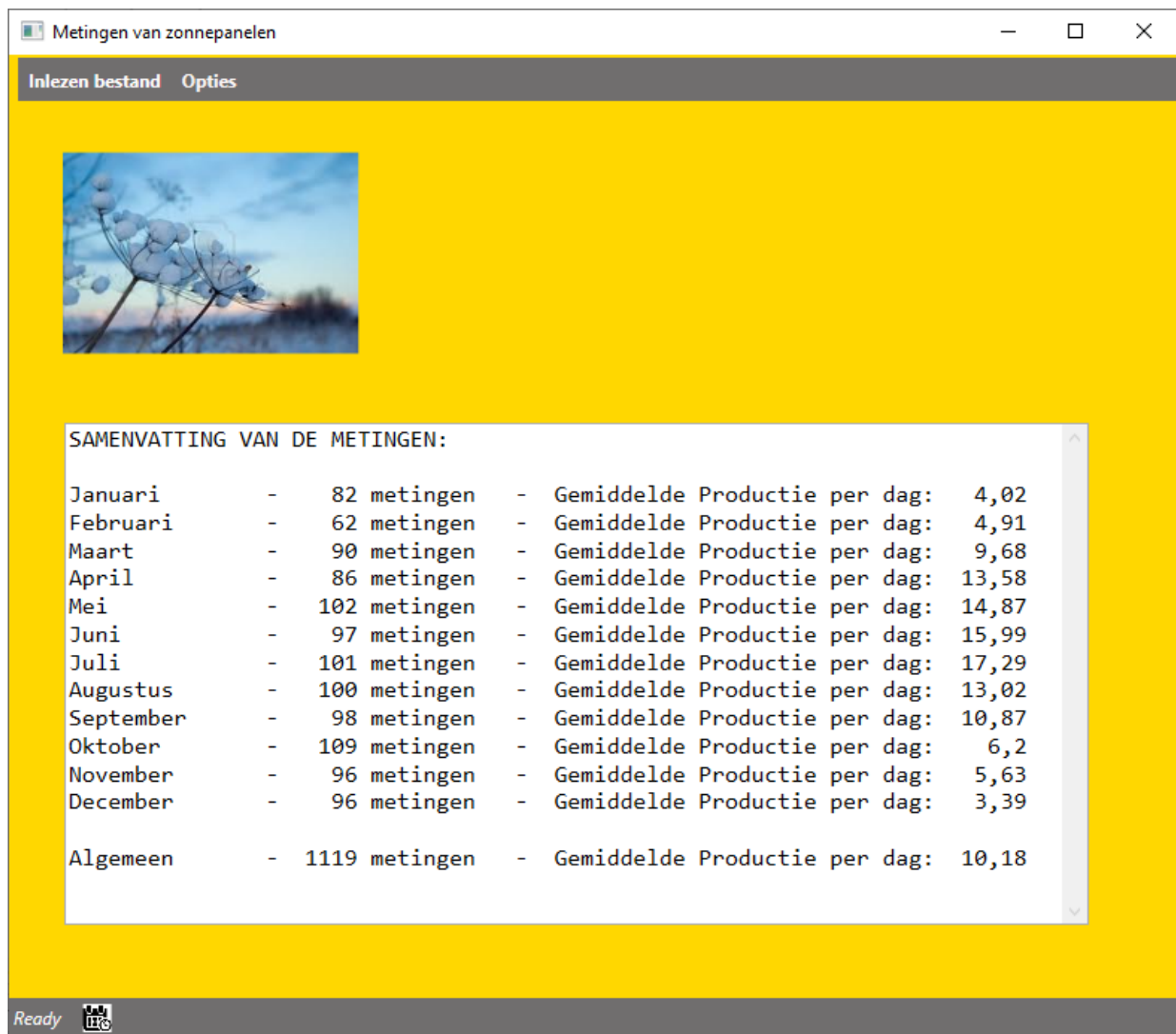
- **Inlezen.** Deze keuze bevat 2 subitems, nl. Details en Samenvatting:
 - ✓ Bij het klikken op de keuze *Details* worden de detailmetingen uit het tekstbestand *MetingenOpbrengstZonnepanelen.txt* ingelezen en regel per regel getoond in het tekstvak **txtResultaat**. Je mag veronderstellen dat het betreffende tekstbestand aanwezig is op de door u gekozen locatie. Zie verder voor een voorbeeld van een schermafdruck.
 - ✓ Bij het klikken op de keuze *Samenvatting* worden de detailmetingen uit het tekstbestand *MetingenOpbrengstZonnepanelen.txt* ingelezen. In dit geval worden de details niet getoond, maar wordt een samenvatting gegeven: per maand wordt het aantal metingen en het gemiddelde van die metingen weergegeven. Zie verder voor een voorbeeld van een schermafdruck.
- **Opties.** Deze keuze bevat eveneens 2 subitems, nl. Datum tonen/verbergen en **Achtergrondkleur instellen.**

- ✓ Bij het klikken op de keuze *Datum tonen/verbergen* wordt de de datum en tijd in de statusbalk getoond of verborgen.
- ✓ Bij het klikken op de keuze *Achtergrondkleur* instellen wordt de achtergrondkleur van het venster gewijzigd. Bij de start van het programma is de achtergrondkleur "geel". Telkens geklikt wordt op de keuze *Achtergrondkleur* instellen verandert willekeurig de achtergrondkleur in één van volgende kleuren: geel, wit, blauw, rood of groen. Zorg er tevens voor dat er nooit 2 keren na elkaar dezelfde kleur geselecteerd wordt. Bv. is de achtergrondkleur geel, dan zal de volgende achtergrondkleur nooit geel zijn, maar wit, blauw, rood of groen.

Afdrukvoorbeeld met detailgegevens:



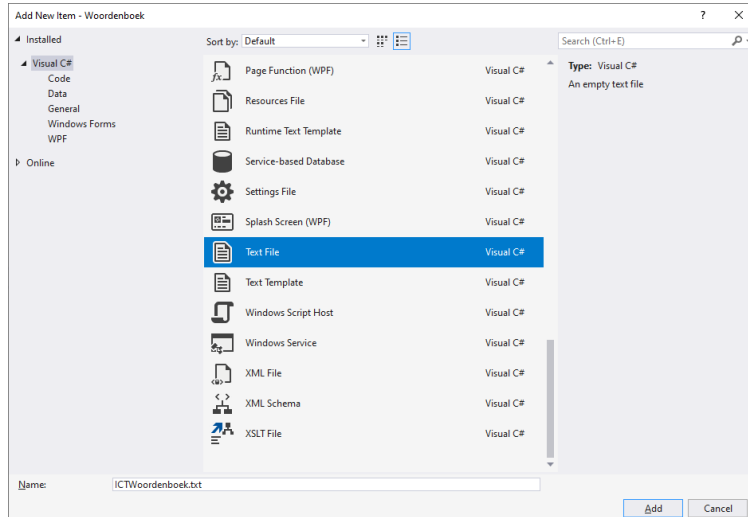
Afdrukvoorbeeld met samenvatting (per maand) van gegevens:



Meerdere vensters

Toepassing 5 : **Woordenboek**

Schrijf een toepassing om de Nederlandse vertaling van Engelse specifieke technische ICT-benamingen op te slaan en op te zoeken en eventueel zelf te vertalen. De vertaling wordt in een tekstbestand bijgehouden.



```
1 memory | geheugen
2 file | bestand
3 operating system | besturingssysteem
4 clipboard | klembord
5 hacker | cyberboef
6 application | toepassing
7 driver | stuurprogramma
8 font | lettertype
9 pixel | beeldscherm punt
10 spamming | e-mail reclame
11 RAM | werkgeheugen
12 router | data verkeerregelaar
13 URL | webadres
14 vulnerability | veiligingslek
15 selfie | ikkie
16 comment | commentaar
17 label | etiket
```

Voeg eventueel zelf een tekstfile toe in je toepassing.

De toepassing omvat 3 vensters en een statische klasse.

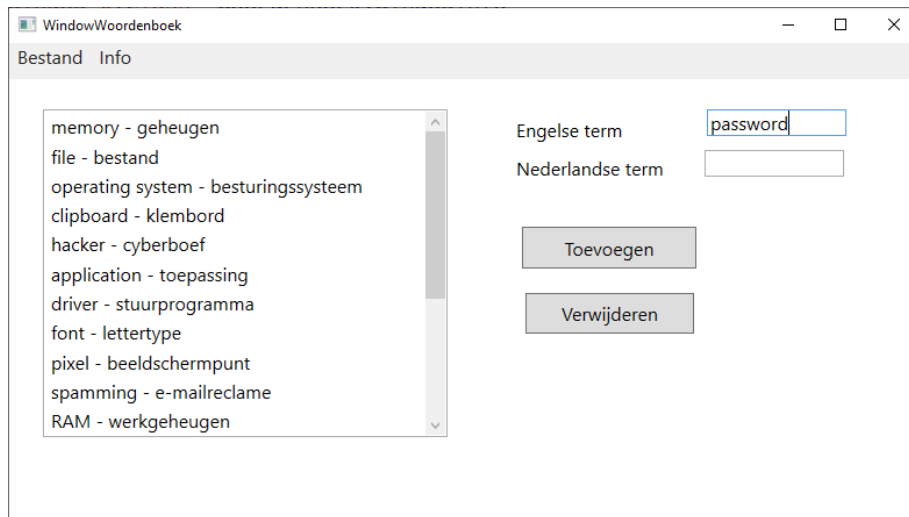
1 Klasse Wachtwoorden.cs

De statische klasse omvat 2 publieke List<> waar respectievelijk de Nederlandstalige termen en de Engelstalige termen in worden bijgehouden. Je kan ook kiezen voor een dictionary.

2 WindowsWoordenboek

Met de opdrachtknop *btnToevoegen* kan je een specifieke benaming in het Engels en de corresponderend Nederlandstalige term toevoegen. De lijst van toegevoegde woorden worden links in een listbox *LbxWoorden* toegevoegd (ook aan je List<>).

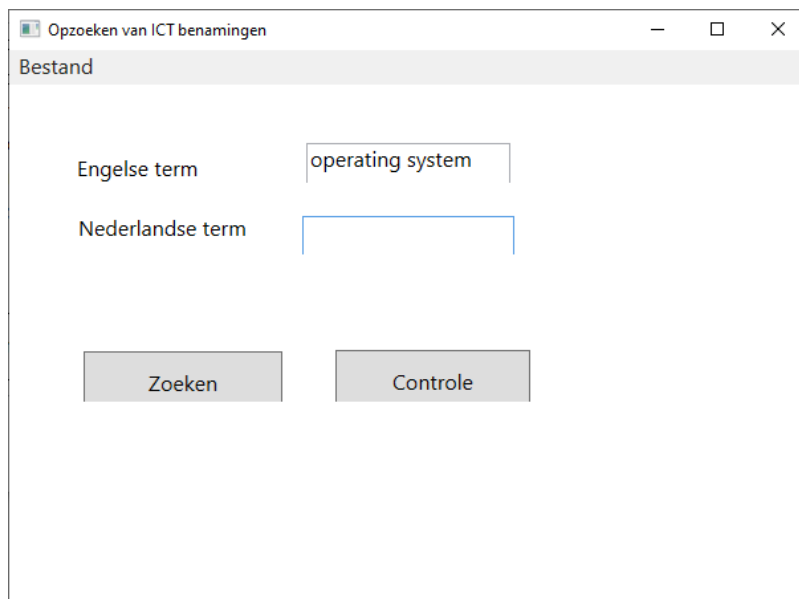
Met de opdrachtknop *btnVerwijderen* worden de termen uit de Listbox en de List<> verwijderend.



3 WindowsZoeken

Het menu bestaat uit *Bestand*: *Zoeken* (idem opdrachtknop Zoeken) - *Sluiten* (sluit het venster) en *Info*.

De opdrachtknop *BtnZoeken* geeft random een Engelse benaming. Geef de vertaling in het tekstvak *TxtNederlands* en met de opdrachtknop *BtnControle* wordt gecontroleerd of de Nederlandse vertaling correct is.



4 WindowsOver

Gebruik de informatie uit je bestand AssemblyInfo.cs (zie Project Properties) om je infovenster te voorzien van:

- Productname
- Version
- Copyright
- Companyname
- Description.
-



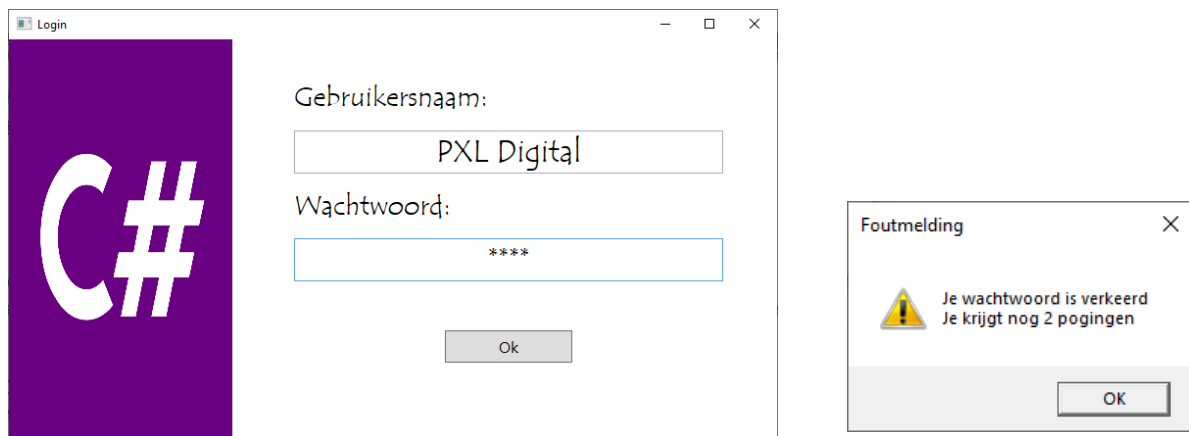
Gebruik de vensters van je vorige toepassing en hanteer nu een *TabControl* ipv meerdere vensters. Gebruik volgende onderdelen voor je toepassing.

1 Klasse Wachtwoorden.cs

De statische klasse omvat de publieke variabele *user* dat in de statusbalk wordt meegegeven en een 2-dimensionale array *wachtwoorden* waarin de gebruikers en haar/hun corresponderende wachtwoorden geregistreerd wordt.

2 Loginvenster

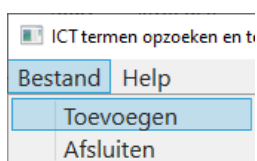
Het venster omvat de gebruikersnaam en het wachtwoord. Wanneer op de opdrachtbutton *BtnOk* geklikt wordt, moet er gecontroleerd worden of het wachtwoord overeenstemt met de mogelijke gebruikers uit de publieke array *wachtwoorden* en vervolgens wordt het hoofdvenster. Wanneer het wachtwoord ongeldig is, verschijnt een aangepast berichtenvenster en na 3 pogingen wordt de toepassing afgesloten.

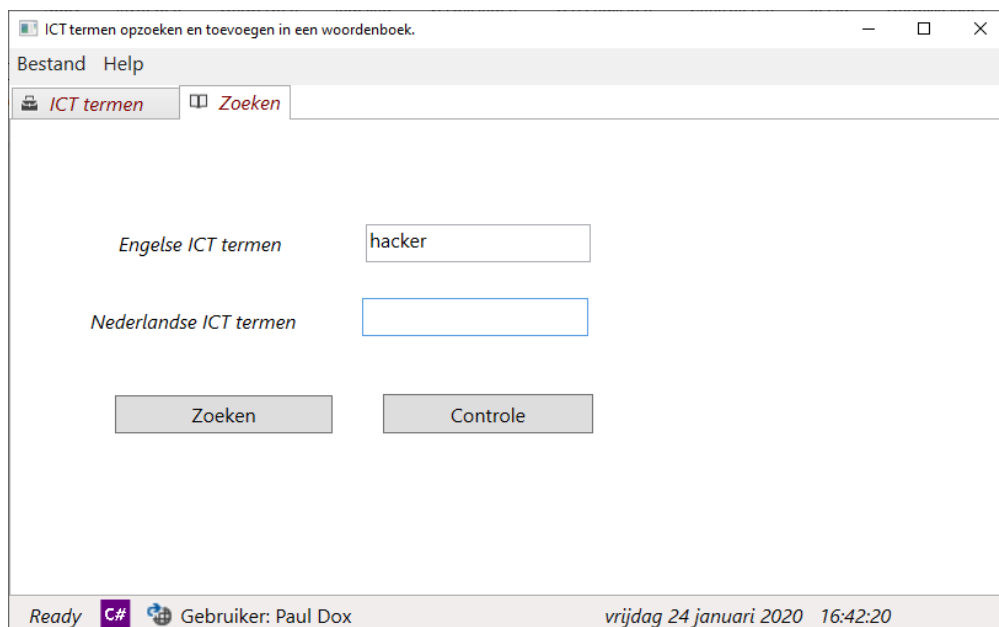
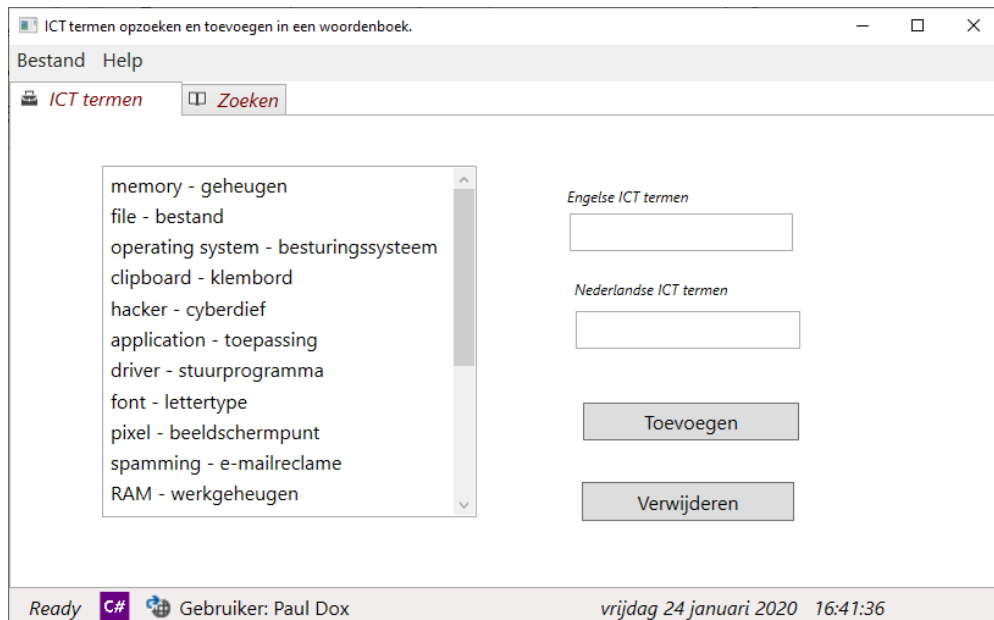


3 MainWindow

Zorg bovendien:

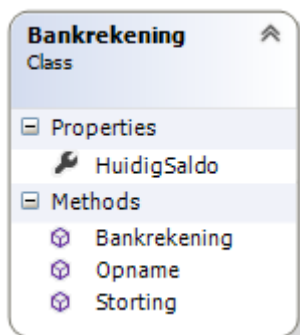
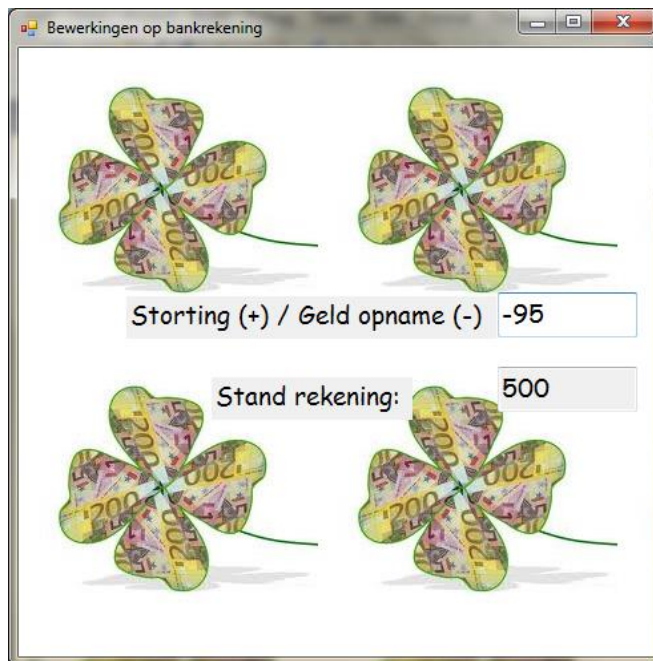
- Listbox om ICT termen toe te voegen of te verwijderen.
- Het bestand *ICTTermen* wordt ingeladen in de listbox en bij Afsluiten wordt het bestand terug bewaard.
- Gebruik Dockpanel voor het menu, tabcontrol en statusbalk
- Leg tekstvakken op het 2^{de} tabblad als het 2^{de} tabblad actief wordt.
- Tabblad 2 moet actief worden wanneer de menu-optie Toevoegen (onder Bestand) geselecteerd wordt.
- Onder het menuitem Help kan een dialoogvenster *Over...* openen met Assembly informatie.





Klasse

Toepassing 7 : KlasseBankrekening



De stortingen of geldopnames worden afgesloten door de return toets. Zorg er ook voor dat je enkel cijfers kan ingeven. Schrijf een voidprocedure *BedragVerhogen()* en een voidprocedure *BedragVerminderen()* die de klasse oproepen.

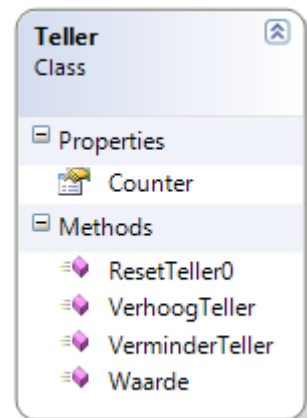
Schrijf een klasse Teller waarmee je een teller kan simuleren.

Eigenschappen:

- Counter: geeft en ontvangt een waarde.

Methods:

- ResetTeller: zet de counter op 0.
- VerhoogTeller: verhoogt counter met 1.
- VerminderTeller: vermindert counter met 1.
- Waarde: verhoogt counter met opgegeven waarde.



frmTeller



Toepassing 9 : **KlassePunten**

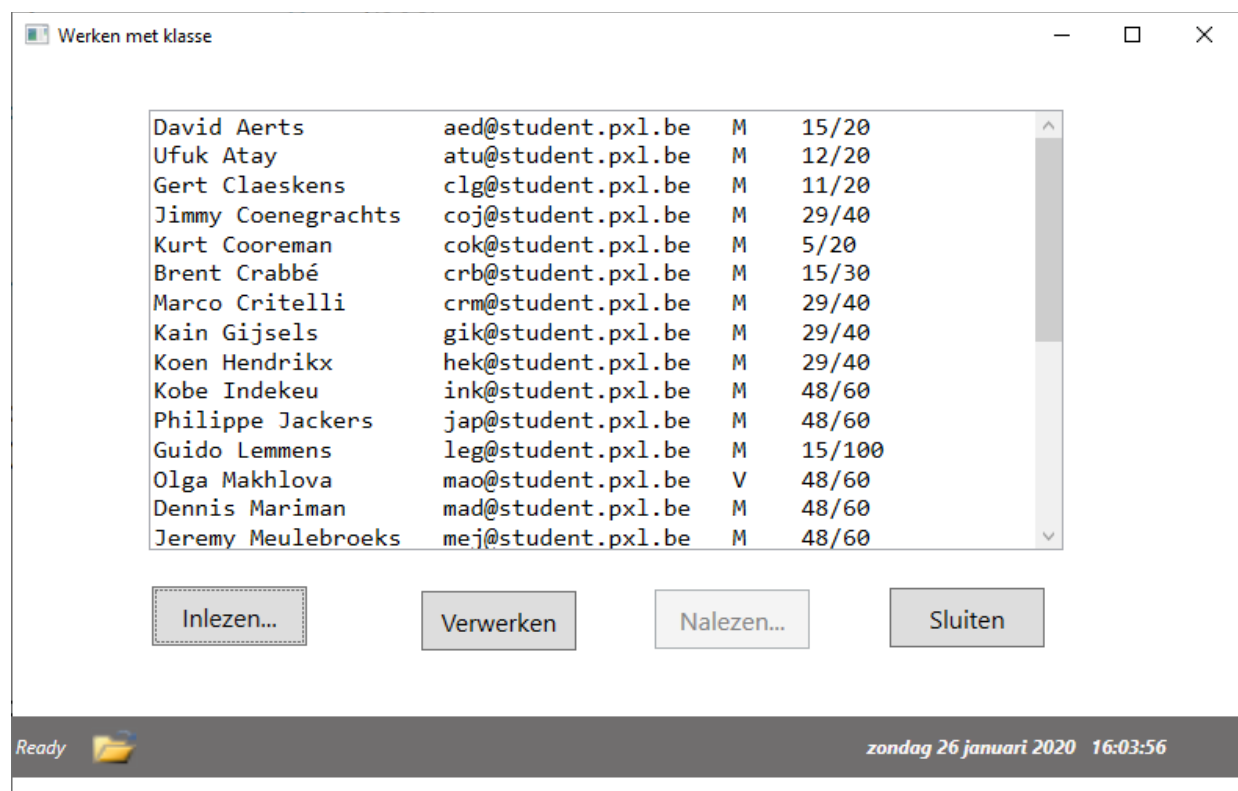
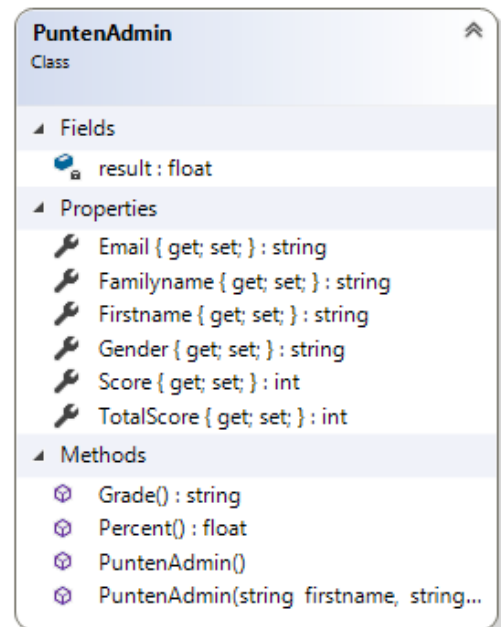
Maak voor toepassing 3 *Punten* een klasse aan de gegevens in te lezen, de punten te verwerken en de resultaten terug weg te schrijven.

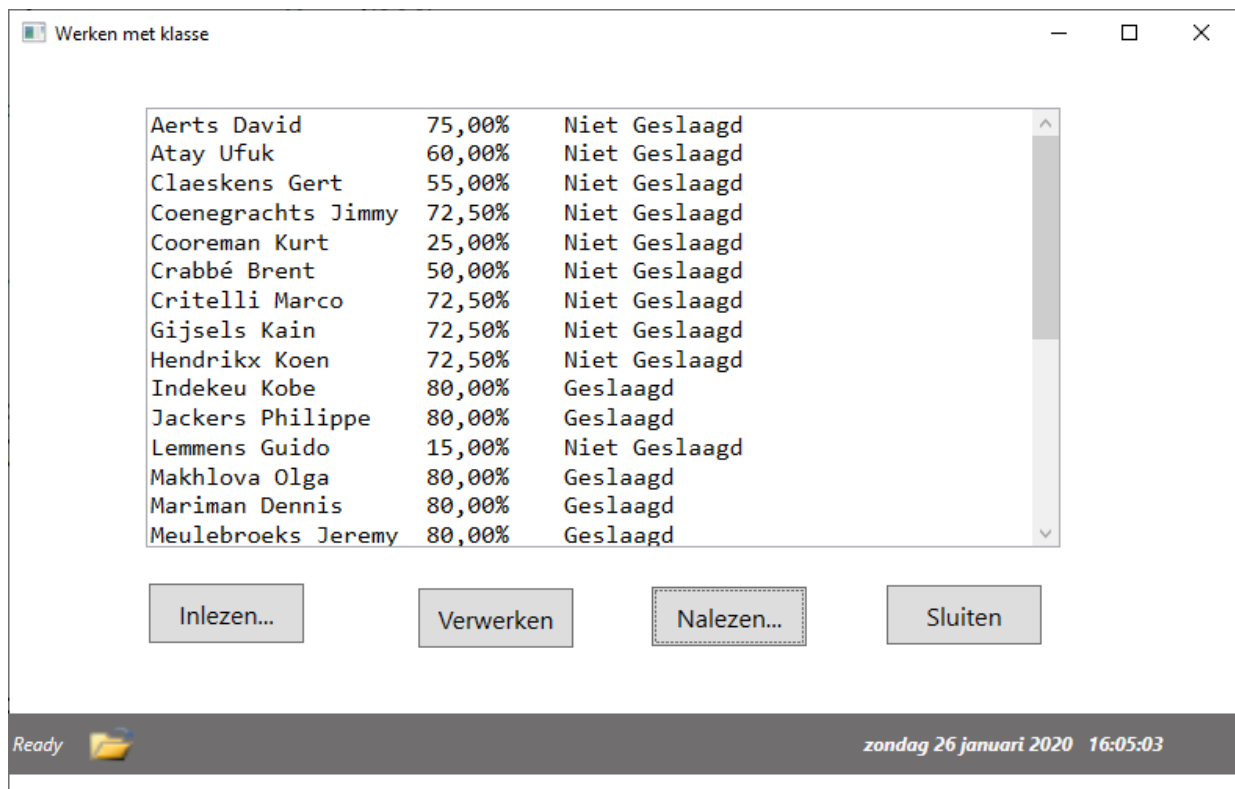
Je maakt voor deze toepassing eerst de (basis)klasse **PuntenAdmin** met volgende eigenschappen:

- **Familyname** - string.
- **Firstname** – string.
- **Gender** - string .
- **Email** – string.
- **Score** – integer (behaalde punten)
- **TotalScore** – integer (puntentotaal)

De klasse Punten bevat volgende methods:

- Constructor **PuntenAdmin** in 2 vormen: één zonder parameters waarbij een lege klasse gecreëerd wordt en één met 6 parameters (de 6 eerder beschreven eigenschappen).
- **Percent** – float (berekent het behaald percentage vb. 72,50%)
- **Grade** – string (geeft “Geslaagd” als je min. 80% hebt anders “Niet geslaagd”)





Maak een toepassing **KlassePersoneel** om een personeelsobject te creëren en de gegevens van dat personeelsobject te tonen.

Class diagram



Je maakt voor deze toepassing eerst de (basis)klasse **Personeel** met volgende eigenschappen:

- **Naam** - string.
- **Voornaam** -string.
- **Geslacht** - string (waarde “M” of “V”).
- **Beoordelingscijfer** – integer waarde van minimum 0 en maximum 10.
- **Startjaar** – integer die aangeeft in welk kalenderjaar het personeelslid in dienst getreden is.

Deze klasse bevat ook nog enkele “read-only” eigenschappen:

- **Aantdienstjaren** – integer die het aantal dienstjaren weergeeft: huidig jaar wordt verminderd met het startjaar.
- **Geslachttekst** – string met als waarde “Mannelijk” of “Vrouwelijk” (afhankelijk van de waarde van de eigenschap Geslacht)
- **Premie** – decimal. De premie wordt als volgt bepaald: eerst wordt het basisbedrag berekend: 500€, vermeerderd met 20€ per dienstjaar. Dit bedrag wordt gehalveerd voor personeelsleden die een Beoordelingscijfer hebben dat lager is dan 5. Voor diegenen die een beoordelingscijfer van 7 of 8 hebben, wordt het basisbedrag met 50% verhoogd. Voor wie een beoordelingscijfer van 9 of 10 heeft, wordt het basisbedrag verdubbeld.

De klasse **Personeel** bevat volgende methods:

- Constructor **Personeel** in 2 vormen: één zonder parameters waarbij een “leeg” personeelslid gecreëerd wordt en één met 5 parameters (de 5 eerder beschreven eigenschappen, in de volgorde zoals beschreven).
- **VerhoogBeoordeling** – deze methode zorgt ervoor dat het beoordelingscijfer met 1 verhoogd wordt.
- **VerlaagBeoordeling** – deze methode zorgt ervoor dat het beoordelingscijfer met 1 verlaagd wordt.

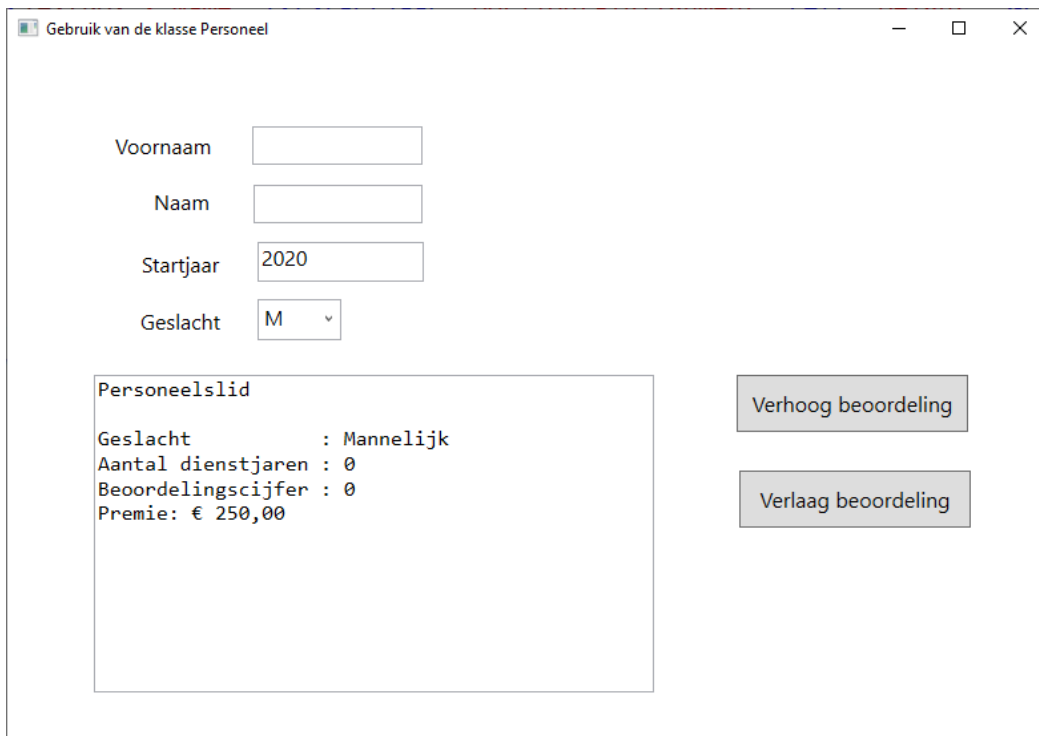
- **InformatieVolledig** – deze funciemethode geeft een string als resultaat met alle informatie van het personeelslid. (zie verder inhoud van het tekstvak txtResultaat).

Bij de start van het programma, krijgt de gebruiker de vraag of een “leeg” personeelsobject of een “ingevuld” personeelsobject gecreëerd moet worden. In het eerste geval is er geen naam en voornaam (leeg), is het geslacht “M” en het startjaar 2000. In het 2^{de} geval kies je zelf de “invulwaarden”.



Na die keuze verschijnt het venster Personeelsklasse.

Bij keuze van een “leeg” personeelsobject:

A window titled "Gebruik van de klasse Personeel" with standard window controls (minimize, maximize, close). It contains a form with four input fields: "Voornaam", "Naam", "Startjaar" (containing "2020"), and "Geslacht" (a dropdown menu with "M" selected). Below the form is a text area labeled "Personeelslid" containing the following text:

```
Geslacht      : Mannelijk
Aantal dienstjaren : 0
Beoordelingscijfer : 0
Premie: € 250,00
```

To the right of the text area are two buttons: "Verhoog beoordeling" and "Verlaag beoordeling".

Mogelijke inhoud bij keuze van een ingevuld personeelsobject:

Gebruik van de klasse Personeel

Voornaam

Naam

Startjaar

Geslacht

Personeelslid Paul Dox

Geslacht : Mannelijk
Aantal dienstjaren : 37
Beoordelingscijfer : 9
Premie: € 2 480,00

Verhoog beoordeling

Verlaag beoordeling

Het venster *Personeelsklasse* bevat volgende objecten (+ enkele labels):

- TxtVoornaam: toont de voornaam van het personeelsobject. Bij het verlaten van dit tekstvak wordt een eventuele wijziging in het personeelsobject uitgevoerd en ook al getoond in het tekstvak txtResultaat.
- TxtNaam: toont de naam van het personeelsobject. Bij het verlaten van dit tekstvak wordt een eventuele wijziging in het personeelsobject uitgevoerd en ook al getoond in het tekstvak txtResultaat.
- TxtStartjaar: toont het startjaar (jaar van indiensttreding) van het personeelsobject. Bij het verlaten van dit tekstvak wordt een eventuele wijziging in het personeelsobject uitgevoerd en ook al getoond in het tekstvak txtResultaat.
- CboGeslacht: toont in de listbox het geslacht (M of V) van het personeelsobject. Bij het verlaten van dit tekstvak wordt een eventuele wijziging in het personeelsobject uitgevoerd en ook al getoond in het tekstvak txtResultaat.
- TxtResultaat (niet ingeschakeld). In dit tekstvak wordt de informatie van het personeelsobject getoond door oproep van de methode InformatieVolledig (afdruk zoals in de schermvoorbeelden).
- BtnVerhoogBeoordeling: bij het drukken van deze knop wordt het beoordelingscijfer van het personeelsobject met één verhoogd en ook al getoond in het tekstvak txtResultaat.
- BtnVerlaagBeoordeling: bij het drukken van deze knop wordt het beoordelingscijfer van het personeelsobject met één verlaagd en ook al getoond in het tekstvak TxtResultaat.

Maak een venster **FrmHuis** om (maximum 2) huisobjecten, nl. huis1 en huis2 te creëren en de gegevens van die huisobjecten te tonen.

Je maakt voor deze toepassing eerst de (basis)klasse **Huis** met volgende eigenschappen:

- **Locatie** – string – bevat de gemeente van het huis.
- **Lengte** - double.
- **Breedte** - double.
- **AantalVerdieping** – integer waarde van minimum 1. (gelijkvloers wordt als een verdieping beschouwd). Geef foutmelding als getracht wordt om een negatieve of nulwaarde aan deze eigenschap toe te kennen.
- **Type** – string die het type van het huis aangeeft: “O” voor “open bebouwing”, “H” voor “half-open bebouwing” en “G” voor “Gesloten bebouwing”.

De klasse Huis bevat volgende methods:

- Constructor **Huis** in 2 vormen: één constructor zonder parameters waarbij een “leeg” huisobject gecreëerd wordt (lege Locatie - Lengte, Breedte en AantalVerdieping op nul, type op “O”) en één constructor met 5 parameters (de 5 eerder beschreven eigenschappen, in de volgorde zoals beschreven).
- **VerhoogAantalVerdiepingen** – deze methode zorgt ervoor dat het aantal verdiepingen verhoogd wordt met de waarde van de doorgegeven integer parameter (1 indien geen parameter wordt doorgegeven).
- **VerlaagAantalVerdiepingen** – deze methode zorgt ervoor dat het aantal verdiepingen verlaagd wordt met de waarde van de doorgegeven integer parameter (1 indien geen parameter wordt doorgegeven). OPM: de waarde mag niet tot gevolg hebben dat de eigenschap AantalVerdieping negatief zou worden. (zie eerder)
- **Oppervlakte** – deze functiemethode geeft een double als resultaat, nl. de vloeroppervlakte in vierkante meters van één verdieping (dus lengte, vermenigvuldigd met breedte). – zou ook met een read-only property kunnen.
- **Inhoud** - deze functiemethode geeft een double als resultaat, nl. de inhoud van het huis in kubieke meters. Je mag veronderstellen dat elke verdieping 2,5m hoog is en dat elke verdieping ook even groot is (dus vloeroppervlakte, vermenigvuldigd met het aantal verdiepingen en met 2,5). – zou ook met een read-only property kunnen.

Bij de start van het programma, wordt het venster FrmHuis getoond:.

Het venster bevat (+ enkele labels):

- TxtLocatie: bevat de Locatie van een te creëren huisobject.
- TxtLengte: bevat de Lengte van een te creëren huisobject.
- TxtBreedte: bevat de Breedte van een te creëren huisobject.
- TxtAantalVerdiepingen: bevat het AantalVerdiepingen van een te creëren huisobject.
- GrpType: een groepsvak met stackpanel om het type van een te creëren huisobject te bepalen door middel van de keuzerondjes radOpen, radHalfopen en radGesloten.
- TxtResultaat (niet ingeschakeld). In dit tekstvak wordt de informatie van een huisobject getoond (bij drukken van de knoppen btnTonen1 en btnTonen2).
- BtnCreatie1 en btnCreatie2: deze knoppen dienen voor de creatie van respectievelijk een object huis1 en huis2. Gebruik in het ene geval de constructor zonder argumenten en in het andere geval de constructor met argumenten.
- Merk op dat als het object huis1 al bestond, dat het bij het terug klikken op deze knop volledig “overschreven” wordt (eigenlijk wordt het “oude” object huis1 gewist en er wordt een nieuw object met diezelfde naam gemaakt).
- BtnVerhogen1 en btnVerhogen2: bij het drukken van deze knoppen wordt het aantal verdiepingen van respectievelijk het object huis1 en huis2 verhoogd met 1 verhoogd.
- BtnMeerdereVerhogingen1 en BtnMeerderVerhogingen2: bij het drukken van deze knoppen wordt het aantal verdiepingen van respectievelijk het object huis1 en huis2 verhoogd met een door de gebruiker ingegeven waarde verhoogd. (in te geven met een InputBox).
- BtnTonen1 en BtnTonen2: bij het drukken van deze knoppen worden de gegevens van respectievelijk de objecten huis1 en huis2 getoond in het tekstvak txtResultaat. (na een titel “gegevens van huis1” of “gegevens van huis2”).

- BtnLeegmaken: bij het drukken van deze knop worden de tekstvakken in het venster leeggemaakt en RadOpen wordt aangevinkt.
- Je mag bij de start van het programma al onmiddellijk een “leeg” object huis1 en huis2 creëren. Dit maakt de rest van de toepassing iets eenvoudiger. Je moet dan niet bij alle acties (bij verhoging van verdieping, verlaging van verdieping, tonen van informatie) testen of het betreffende object huis1 of huis2 bestaat in de klasse Huis.
- Creëer je de objecten niet onmiddellijk (alleen definitie), dan kan je programma eveneens werken, maar dan moet je de test of het object gecreëerd is wel telkens uitvoeren. Zo niet, dan krijg je fouten. (van een niet gecreëerd object kan je geen eigenschappen instellen of opvragen).

Toevoeging: definieer nu ook huis3 en huis4:

`private Huis huis3;`

`private Huis huis4;`

Na de creatie van huis1 voeg je instructie “huis3 = huis1;” toe. Na de creatie van huis2 voeg je instructie “huis4 = huis2;” toe.

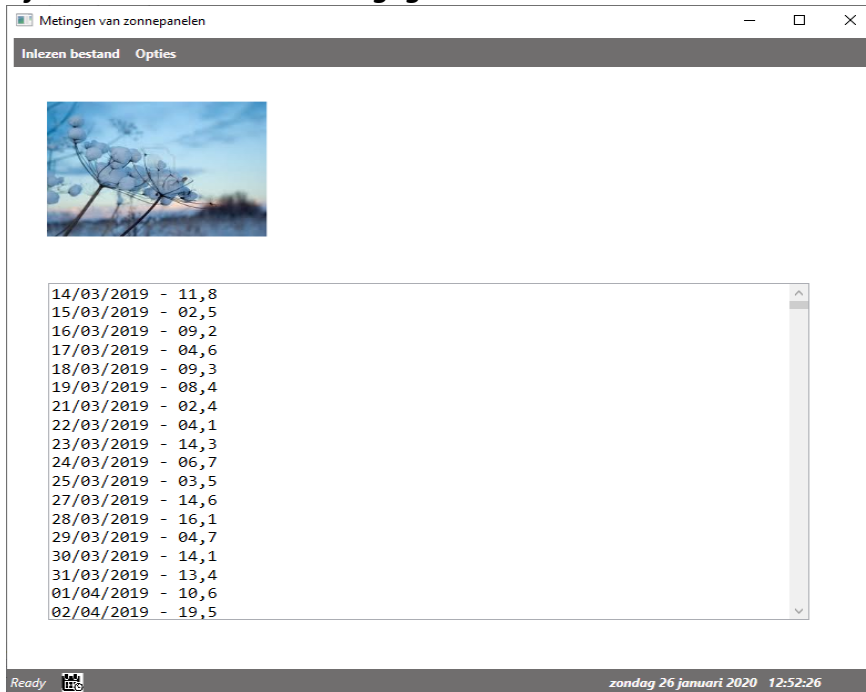
Voeg ten slotte ook de 2 opdrachtknoppen btnTonen3 en btnTonen4 toe. Test de werking uit. Wat stel je vast?

Toepassing 12 : **KlasseZonnepaneelmetingen**

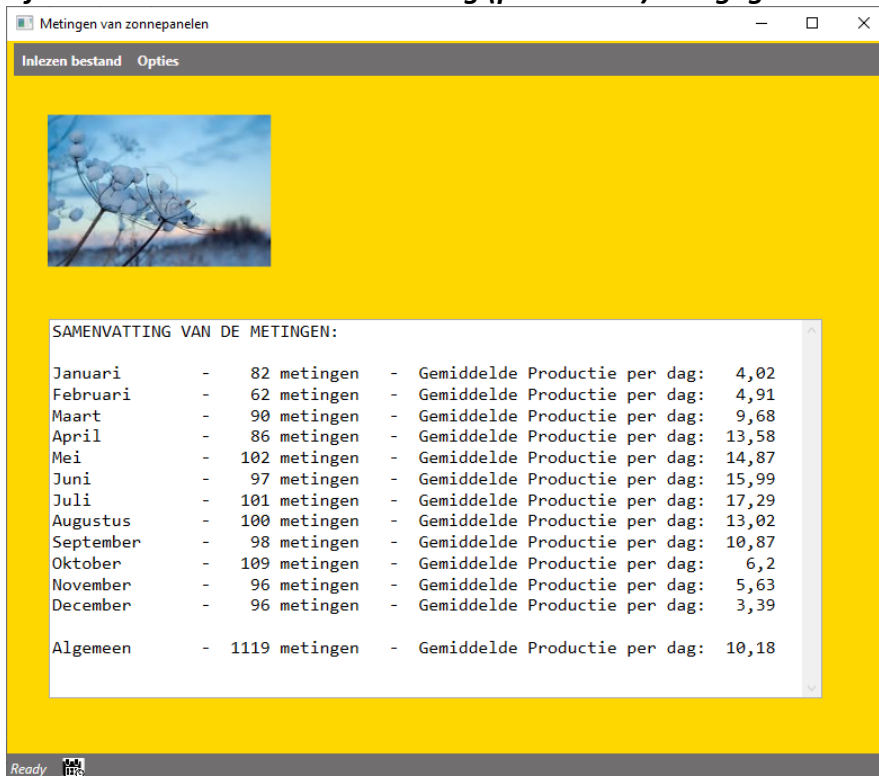
Maak voor toepassing 4 *Zonnepaneelmetingen* een klasse aan de gegevens in te lezen, de metingen te verwerken en de resultaten terug weg te schrijven.

Je maakt voor deze toepassing een eigen klasse **Metingen**. Formuleer zelf **en methodes**.

Afdrukvoorbeeld met detailgegevens:



Afdrukvoorbeeld met samenvatting (per maand) van gegevens:



Deze toepassing maakt gebruik van een tekstbestand “LedenSportkamp.txt” waarin de studentengegevens zijn opgenomen. Eerste records uit het bestand:

Abazar	Brecht	6VOL1
Abdellah	Nico	7TEN1
Abdelmalki	Wim	5TEN1
Abry	Thibaut	9ATL1
Acciani	Maarten	5PAA1
Achten	Joren	5VOL1
Adamczyk	Jody	9KAY1
Adriaenssens	Eveline	6ATL1
Aerts	Benjamin	1VOL2
Aerts	Lies	3ATL1
Aerts	Lotte	4PAA1
Aerts	Martijn	2VOL2
Aerts	Michael	9TEN2
Aerts	Roberto	4VOL2
Aertssen	Pieter	7KAY1
. . .		

In de eerste 30 posities staat de naam van de deelnemer, dan volgt (ook 30 posities) de voornaam van de deelnemer. Op de 61^{ste} positie staat een cijfer (maximum 9) dat aangeeft in welke vakantieweek de betreffende persoon aan een sportkamp deelneemt. (begin juli → 1, eind augustus → 9).

De 62^{ste} tot en met de 64^{ste} positie bevat de code van het gevolgde sportkamp. Volgende tabel geeft de mogelijke codes weer, met ook de omschrijving van het sportkamp én de prijs voor het sportkamp:

code	omschrijving	prijs
ATL	Atletiek	190
VOE	Voetbal	190
ZWE	Zwemmen	210
KAY	Kayak	250
TEN	Tennis	260
PAA	Paardrijden	310
VOL	Volleybal	190

Tenslotte staat op de 65^{ste} positie terug een cijfer (maximum 9) dat aangeeft de hoeveelste keer deze deelnemer aan een sportkamp deelneemt.

Je mag er van uitgaan dat bij deze structuur nooit fouten zijn en dat het bestand LedenSportkamp.txt aanwezig is op de door u bepaalde locatie. Het aantal records in het tekstbestand is niet vast.

Je maakt voor deze toepassing de (basisklasse) **Lid** met als eigenschappen **Naam**, **Voornaam**, **Sportcode**, **Weeknr** en **KampVolgnr**. De eerste 3 eigenschappen zijn van het type string. De 2 laatste eigenschappen zijn integers.

Tevens zijn er de (uitsluitend te lezen) eigenschappen **NaamVolledig**, **SportNaam**, **KampPrijs** en **TeBetalen**.

NaamVolledig geeft als resultaat een string met de achternaam, gevolgd door 1 blanco en de voornaam (zonder extra blanco's achteraan).

SportNaam geeft als resultaat een string met de omschrijving van de sporttak waarvoor een lid is ingeschreven (zie tabel op vorige pagina).

KampPrijs geeft als resultaat een double met de "normale" prijs van de sporttak waarvoor een lid is ingeschreven (zie tabel op vorige pagina).

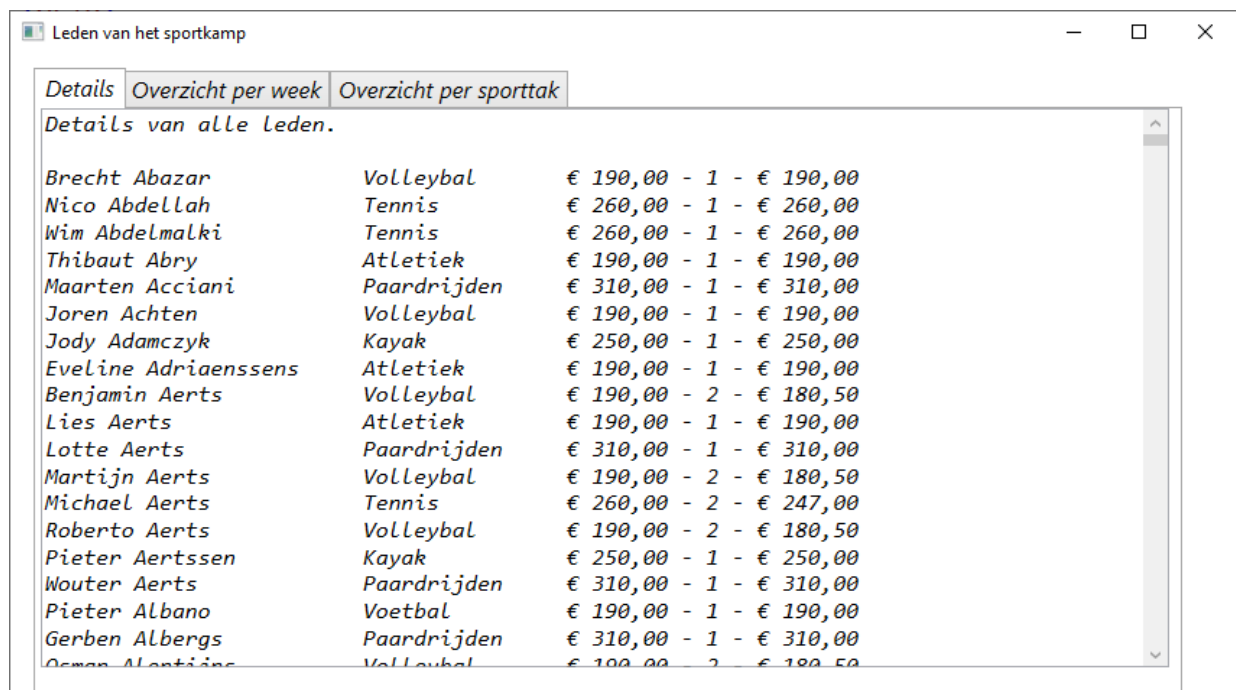
TeBetalen geeft eveneens als resultaat een double met de prijs die het lid moet betalen. Deze prijs is de Kampprijs, waarop eventueel een korting wordt toegepast: wie minstens voor de 5^{de} keer een sportkamp volgt (zie KampVolgnr) krijgt een korting van 10%, wie minstens voor de 2^{de} keer een sportkamp volgt (maar minder dan 5), krijgt een korting van 5%.

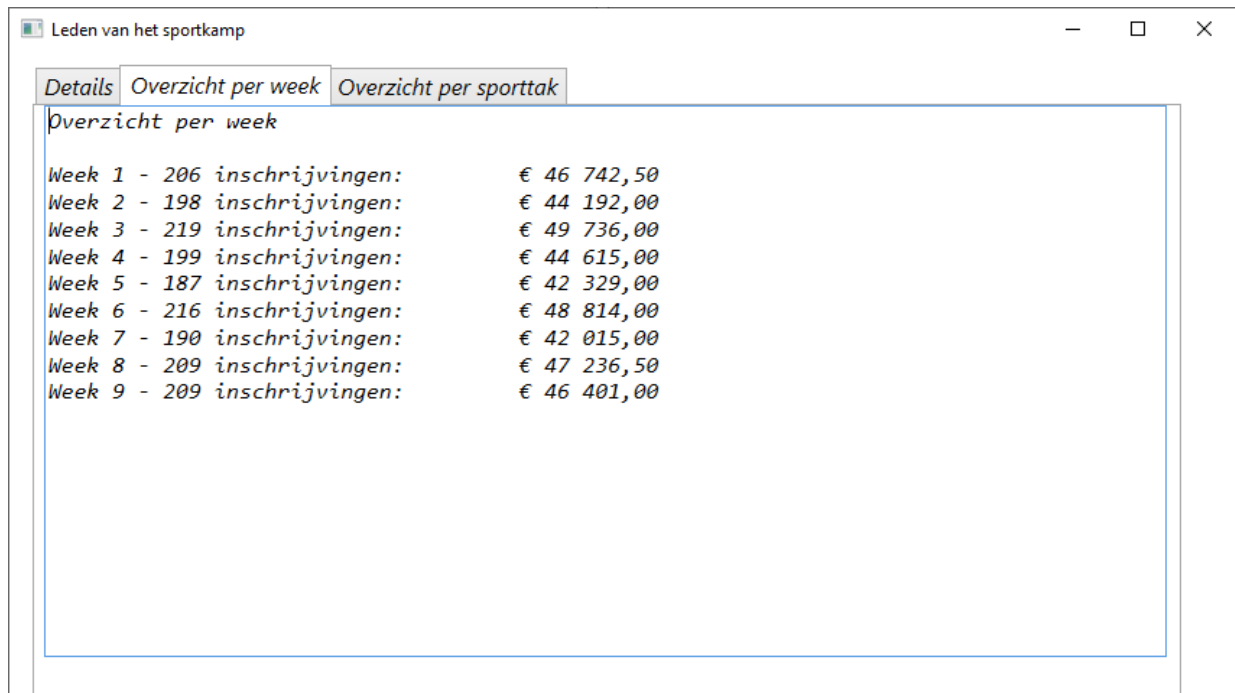
Er is ook nog de methode **InformatieVolledig()**. Deze methode geeft als resultaat een string met vooraan de inhoud van NaamVolledig (rechts aangevuld met blanco's tot een totaal van 35 posities), dan de SportNaam (aangevuld met blanco's tot een totaal van 15 posities), daarna de KampPrijs (normale prijs), dan de waarde van KampVolgnr en ten slotte de inhoud van TeBetalen.

Maak een venster SportkampLeden dat een tabControl bevat met 3 tabbladen (tabPages): Details, Overzicht per week en Overzicht per sporttak. Deze bevatten respectievelijk de tekstvakken TxtDetails, TxtOverzichtWeek en TxtOverzichtSport

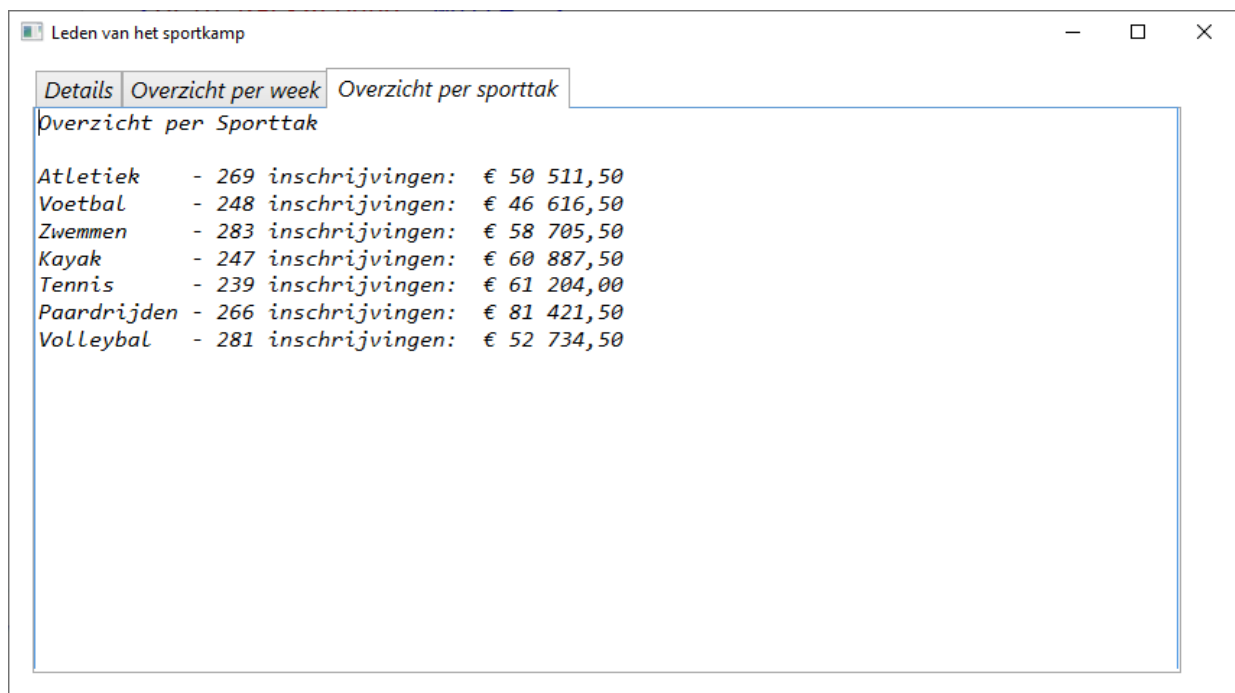
Bij de start van het programma worden onmiddellijk alle records uit het bestand "LedenSportkamp.txt" gelezen. Per record wordt een Lid-object gecreëerd.

Bij de start worden ook onmiddellijk de tekstvakken in de tabbladen gevuld, zoals weergegeven in volgende schermafdrucken:





Het bepalen van de inhoud van het tekstvak txtOverzichtSport (in het tabblad Overzicht per sporttak) is wel wat moeilijker. Een uitdaging voor de gevorderde student.



Je kan eventueel ook nog de omschrijvingen en prijzen van de sporttakken inlezen uit een bestand "Sporten.txt":

"ATL";"Atletiek";190

"VOE";"Voetbal";190

"ZWE";"Zwemmen";210

"KAY";"Kayak";250

"TEN";"Tennis";260

"PAA";"Paardrijden";310

"VOL";"Volleybal";190

Deze gegevens kan je inlezen in een tabel (array) of een list. Je kan hiervoor eventueel zelfs een nieuwe klasse Sport met de eigenschappen Code (sportcode), Omschrijving (omschrijving van de sporttak), Prijs (inschrijvingsbedrag voor de sporttak), Aantal (numerieke integer waarde om bij te houden hoeveel inschrijvingen er zijn voor de betreffende sporttak) en Totaalprijs (numerieke double waarde om bij te houden hoeveel het totale inschrijvingsbedrag is voor de betreffende sporttak)

Deze toepassing maakt eveneens gebruik van het tekstbestand “LedenSportkamp.txt” waarin de gegevens van leden van een sportkamp zijn opgenomen. Zie vorige oefening voor voorbeeldrecords – structuur... .

Je gebruikt voor deze toepassing eveneens de (basisklasse) **Lid** met dezelfde eigenschappen als in vorige oefening beschreven.

Je voegt nog één methode toe, nl. **recordVorm()**. Deze methode geeft als resultaat een string met vooraan de inhoud van Naam (aangevuld met blanco’s tot een totaal van 30 posities), dan de Voornaam (ook aangevuld met blanco’s tot een totaal van 30 posities), dan het cijfer van het Weeknr, de Sportcode en ten slotte het cijfer van het KampVolgnr. Je krijgt dus steeds een string van 65 posities (je mag er van uitgaan dat Weeknr en KampVolgnr steeds een waarde tussen “1” en “9” bevatten).

Het startvenster bevat een listbox *LstLeden*.

Bij de start van het programma worden onmiddellijk alle records uit het bestand “LedenSportkamp.txt” gelezen. Per record wordt een Lid-object gecreëerd. Dit object wordt in een list Ledenlijst toegevoegd en tevens wordt in de listbox *LstLeden* een item (een regel) toegevoegd met als waarde het resultaat van de methode InformatieVolledig(). Onmiddellijk na de start krijg je dus weergave zoals in volgende schermafdruck:



Naam	Sport	Weeknr	KampVolgnr	Totaal
Brecht Abazar	Volleybal	1	1	€ 190,00
Nico Abdellah	Tennis	1	1	€ 260,00
Wim Abdelmalki	Tennis	1	1	€ 260,00
Thibaut Abry	Atletiek	1	1	€ 190,00
Maarten Acciani	Paardrijden	1	1	€ 310,00
Joren Achten	Volleybal	1	1	€ 190,00
Jody Adamczyk	Kayak	1	1	€ 250,00
Eveline Adriaenssens	Atletiek	1	1	€ 190,00
Benjamin Aerts	Volleybal	2	1	€ 180,50
Lies Aerts	Atletiek	1	1	€ 190,00
Lotte Aerts	Paardrijden	1	1	€ 310,00
Martijn Aerts	Volleybal	2	1	€ 180,50
Michael Aerts	Tennis	2	1	€ 247,00
Roberto Aerts	Volleybal	2	1	€ 180,50
Pieter Aertssen	Kayak	1	1	€ 250,00
Wouter Aerts	Paardrijden	1	1	€ 310,00

Opm: je mag veronderstellen dat het bestand “LedenSportkamp.txt” aanwezig is op de door u gewenste locatie en dat er minstens een aantal records in voorkomen. De records zijn eveneens correct opgebouwd.

Als de gebruiker op een bepaalde regel van de listbox *LstLeden*, dan verschijnt een nieuw venster *Wijzigen* met de gegevens van de persoon die in de listbox *LstLeden* geselecteerd was.

Schermbvoorbeeld:

Gegevens van een lid van sportkamp.

Naam
Bolat

Voornaam
Sandra

Weeknr (1 to 9)
9

Kamp volgnr.
1

Sportkamp
ATL - Atletiek
VOE - Voetbal
ZWE - Zwemmen
KAY - Kayak
TEN - Tennis
PAA - Paardrijden
VOL - Volleybal

Bewaren Annuleren

Dit venster bevat volgende objecten (met bijhorende labels):

- **TxtNaam, TxtVoornaam, TxtWeeknr en TxtKampVolgnr**: tekstvakken voor de eventuele wijziging van de betreffende gegevens van het lid van een sportkamp.
- **LstSport**: een listbox met de mogelijke sporten (vooraan op de eerste 3 posities de code, daarna de omschrijving)
- **BtnBewaar**: button om het venster te sluiten en de gegevens te bewaren. In dit geval wordt in de listbox van het startvenster de eventuele wijziging onmiddellijk getoond.
- **BtnAnnuleer**: button om het venster te sluiten zonder de gegevens te bewaren.

Je moet geen controles op ingaven doen. Je mag dus veronderstellen dat de gebruiker juiste gegevens ingeeft. (dus bv. voor TxtWeeknr en TxtKampVolgnr een cijfer tussen “1” en “9”)

Schermbvoorbeelden bij bv. wijziging van de sport “Tennis” naar “Zwemmen” (en drukken van knop *BtnBewaar*):

Gegevens van een lid van sportkamp.

Naam
Bolat

Voornaam
Sandra

Weeknr (1 to 9)
9

Kamp volgnr.
1

Sportkamp
ATL - Atletiek
VOE - Voetbal
ZWE - Zwemmen
KAY - Kayak
TEN - Tennis
PAA - Paardrijden
VOL - Volleybal

Bewaren Annuleren

Christine Bloemen	Tennis	€ 260,00 - 1 - € 260,00
Michael Bloemen	Kayak	€ 250,00 - 1 - € 250,00
Jannes Bloem	Paardrijden	€ 310,00 - 2 - € 294,50
Martijn Bobbaerts	Tennis	€ 260,00 - 1 - € 260,00
Koen Bochfraati	Voetbal	€ 190,00 - 3 - € 180,50
Maxim Bodvin	Kayak	€ 250,00 - 1 - € 250,00
Laurens Boeckx	Tennis	€ 260,00 - 6 - € 234,00
Marvin Boelen	Kayak	€ 250,00 - 2 - € 237,50
Thibault Boelen	Tennis	€ 260,00 - 1 - € 260,00
Gunther Boeyink	Zwemmen	€ 210,00 - 1 - € 210,00
Robin Boffé	Kayak	€ 250,00 - 1 - € 250,00
Maarten Bogaerts	Kayak	€ 250,00 - 1 - € 250,00
Sarah Bogaerts	Paardrijden	€ 310,00 - 1 - € 310,00
Sandra Bolat	Zwemmen	€ 210,00 - 1 - € 210,00
Tine Boldbaatar	Zwemmen	€ 210,00 - 1 - € 210,00
Nathalie Bollaert	Kayak	€ 250,00 - 1 - € 250,00

Bij het afsluiten van de toepassing wordt ten slotte nog gevraagd of de (eventueel gewijzigde gegevens) bewaard moeten worden in het tekstbestand: melding

"c:\...\LedenSportkamp.txt overschrijven?". Bij het drukken van de Ja-knop wordt het bestand overschreven met de gewijzigde gegevens, waarna daarvan een melding gegeven wordt: "De gegevens zijn bewaard."

Deze toepassing maakt gebruik van een tekstbestand "studenten.txt" waarin de studentengegevens zijn opgenomen. Eerste records uit het bestand:

IVandenbossche	Els	Kroonweg 2	3500V36
IVan Summeren	Koen	Kerkstraat 7	3500M53
GLeys	Kristel	Lelielaan 4	3600V
GHox	Peter	Krekelstraat 11	3500M
IDAems	Greta	Sparrenhof 18	3290V66
GVaes	Marcel	Valleilaan 4	3290M
GJacobs	Jan	Luikse steenweg 124	3500M
IMaes	Jean	Sparstraat 12	3800M62
GWillems	Patrick	Kerkstraat 14	3290M
GKuppers	Joke	Putstraat 6	3600V
GLambrechts	Inge	Lange straat 365	3500V
GMichiels	Jaak	Putsteeg 5	3000M
GPauwels	Nico	Djef Antenstraat 56	3500M
ILaeremans	Claudia	Runkstersteenweg 12	3500V25
GBoonen	Tanja	Krekelstraat 4	3700V
GHeuts	Diederik	Grote baan 47	3500M
GPelemans	Jeanne	Leuvensesteenweg 144	3200V
GHonings	Linda	Aarschotse steenweg 456	3000V
GPeeters	Ellen	Djef Swennenstraat 89	3500V ...

In de eerste positie staat het type van de student (G → gewoon student of I → student met individueel traject, volgt geen standaardlespakket, maar individueel gekozen vakken). Dan volgt de naam van de student (20 posities), de voornaam van de student (20 posities), de straat van de student (30 posities), de postcode van de student (4 posities). Op de 76^{ste} positie het geslacht, dan volgt op de 77^{ste} en 78^{ste} positie het aantal studiepunten. Dit is alleen van toepassing voor studenten met een individueel traject en geeft de zwaarte van hun vakkenpakket aan (voor een gewone student bedraagt dit 60. Voor de IT-studenten wijkt dit hiervan af: kan minder zijn, kan zelf iets meer zijn)

Je mag er van uitgaan dat bij deze structuur nooit fouten zijn en dat het bestand "studenten.txt" aanwezig is op de door u bepaalde locatie. Het aantal records in het tekstbestand is niet vast.

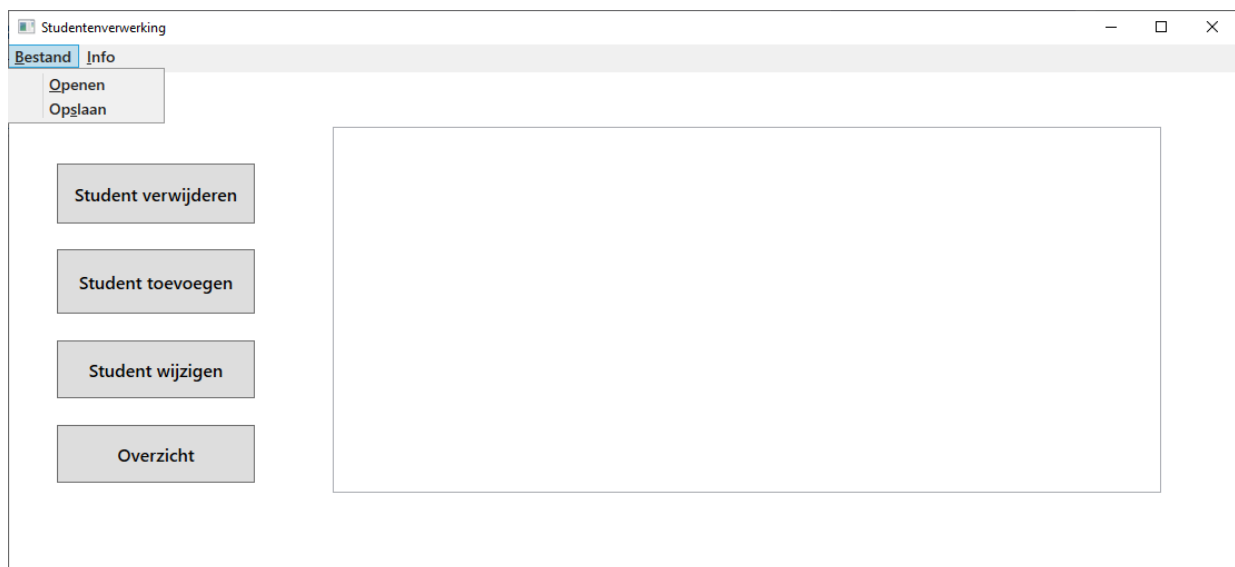
Je maakt voor deze toepassing de (basisklasse) **STUDENT** met als eigenschappen **TypeStudent**, **Naam**, **Voornaam**, **Straat**, **Postcode**, **Geslacht** en **Studiepunten**.

Deze eigenschappen zijn allemaal van het type string, behalve Studiepunten. Dat is een integer (getal van minimum nul tot maximum 99).

Er is bijkomend de (uitsluitend te lezen) eigenschap **Inschrijvingsbedrag**. Deze eigenschap geeft het door de student te betalen "standaard" inschrijvingsbedrag. Dit bedraagt 520€ voor de gewone studenten. Voor studenten met een individueel traject bedraagt het inschrijvingsbedrag 50€, vermeerderd met 8,66€ (520 / 60) per opgenomen studiepunt. Tevens is er de (uitsluitend te lezen) eigenschap **Financierbaarheid**. De waarde hiervan staat op 100 voor de normale studenten (die zijn 100% financierbaar). Voor de IndivTr-studenten is de financierbaarheid ook afhankelijk van het aantal opgenomen studiepunten: 100% financierbaarheid vanaf 45 studiepunten, tussen 30 en 44 studiepunten is de financierbaarheid 75%, tussen 15 en 29 studiepunten is de financierbaarheid 50%. Onder 15 studiepunten is de financierbaarheid 0%.

Ten slotte is er nog de functie **NaamVolledig**. Het resultaat is een string beginnend met de tekst "Standaardstudent " of "Student met individueel traject ", gevolgd door de voornaam en de naam.

Maak een venster STUDENTENVERWERKING_START dat dient als startvenster voor deze toepassing. Het venster ziet er als volgt uit:



Dit venster bevat:

Een **menustructuur** met 2 hoofdkeuzes:

1. Bestandsbewerkingen : bevat een keuze inlezen om studentengegevens in te lezen uit het tekstbestand "Studenten.txt" (zie eerder) en een keuze bewaren om gewijzigde studentengegevens te kunnen wegschrijven. (deze keuze wordt pas mogelijk nadat minstens één keer gegevens zijn ingelezen).

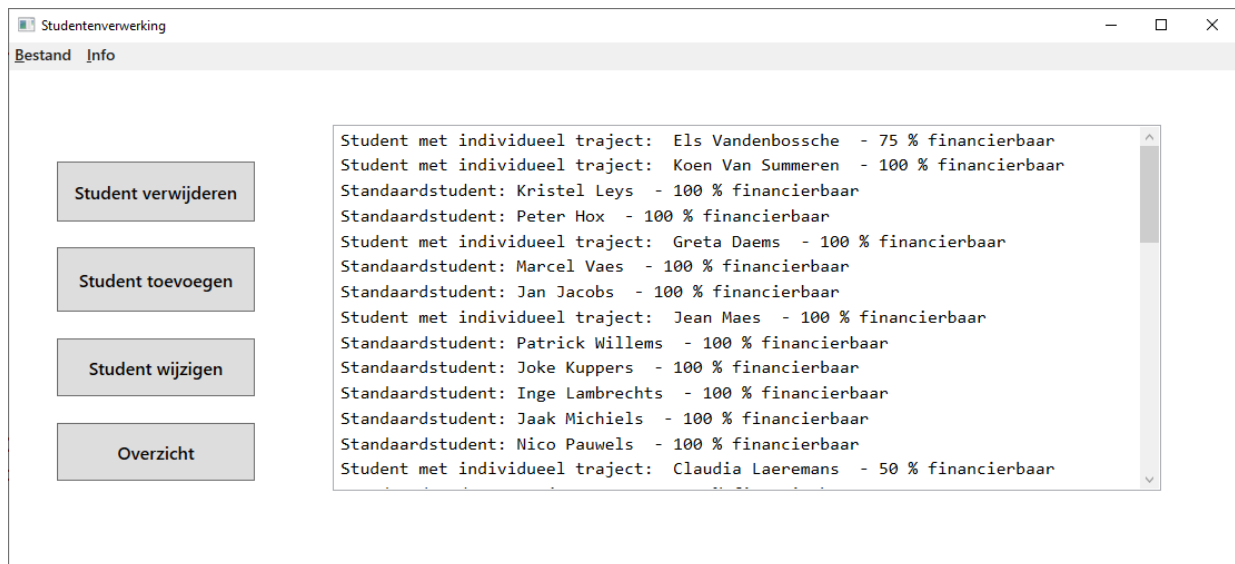
Bij het kiezen van bewaren worden de studentengegevens in "Studenten.txt" overschreven (dezelfde naam!). Pas dan bevat dit tekstbestand de eventueel gemaakte wijzigingen.

2. Info : roept een modaal venster op dat informatie weergeeft:



Een Listbox **LstStudentenlijst** die gevuld wordt bij het inlezen met studentengegevens van alle studenten uit het tekstbestand "studenten.txt".

Venstervoorbeeld:



Een opdrachtknop **BtnVerwijderen**. Bij het drukken op deze knop wordt eerst getest of de gebruiker een bepaalde student geselecteerd heeft. Zo nee, dan verschijnt de foutmelding "Geen student geselecteerd!", anders wordt met een messagebox een bevestiging gevraagd. Indien "Ja" geantwoord wordt, dan wordt de betreffende student verwijderd (uit het geheugen en uit de listbox, wijzigingen in het tekstbestand gebeuren alleen door de eerder beschreven menukeuze "Bestandsbewerkingen – Bewaren").

Een opdrachtknop **BtnWijzigen**. Bij het drukken op deze knop wordt eerst getest of de gebruiker een bepaalde student geselecteerd heeft. Zo nee, dan verschijnt de foutmelding "Geen student geselecteerd!", anders wordt een 2^{de} (modaal) venster **INGAVEFORM** geopend. Onmiddellijk worden de gegevens van de selecteerde student getoond. In geval van een IT-student kan de gebruiker het aantal studiepunten ingeven (anders, dus bij gewone studenten is dit niet beschikbaar). Merk op dat dit venster uitsluitend verlaten kan worden met de OK-knop of de CANCEL-knop. Bij het drukken op OK worden de wijzigingen aanvaard (in het geheugen en in de listbox LstStudentenlijst, wijzigingen in het tekstbestand gebeuren alleen door de eerder beschreven menukeuze "Bestandsbewerkingen – Bewaren").

Objecten in het venster (behalve de labels):

- **CboTypekeuze**: keuzelijst met 2 keuzes ("Gewoon student" en "Student met individueel traject")
- **TxtNaam, TxtVoornaam, TxtPostcode, TxtStraat, TxtGeslacht**: tekstvakken voor de ingave of wijziging van de betreffende gegevens van de student.
- **TxtStudiepunten**: tekstvak voor de ingave of wijziging van het aantal studiepunten. Dit kan alleen bij studenten met een individueel traject. Bij de andere (normale) studenten is dit tekstvak onzichtbaar (bijhorende label ook). OPM: de zichtbaarheid / onzichtbaarheid van dit object wijzigt ook als het type van de student veranderd wordt (wijziging keuzelijst cboTypekeuze).
- **BtnOK**: button om het venster te sluiten en de gegevens te bewaren.
- **BtnCancel**: button om het venster te sluiten zonder de gegevens te bewaren.

Je moet geen controles op ingaven doen. Je mag dus veronderstellen dat de gebruiker juiste gegevens ingeeft. (dus bv. voor studiepunten een getal van 1 tot maximum 99).

Dit venster zou er als volgt kunnen uitzien. (bv. voor wijziging van het huisnummer van de student Kristel Leys van 4 naar 41:

The screenshot shows a Windows-style window titled "Ingeven en wijzigen van studentengegevens". It contains a form with the following fields:

Label	Value
Type student	Gewoon student
Naam	Leys
Voornaam	Kristel
Postcode	3600
Straat	Lelielaan 4
Geslacht	V

Merk ook op dat bij wijziging van het type student van “Gewoon student” naar “Student met individueel traject” er dan een bijkomend tekstvak TxtStudiepunten zichtbaar wordt. (en omgekeerd onzichtbaar zou worden) Bv.:

The screenshot shows the same window as above, but with the "Type student" dropdown set to "Student met een individueel contract". An additional field, "Aantal studiepunten", is now visible at the bottom of the form.

Label	Value
Type student	Student met een individueel contract
Naam	Daems
Voornaam	Greta
Postcode	3290
Straat	Sparrenhof 28
Geslacht	V
Aantal studiepunten	66

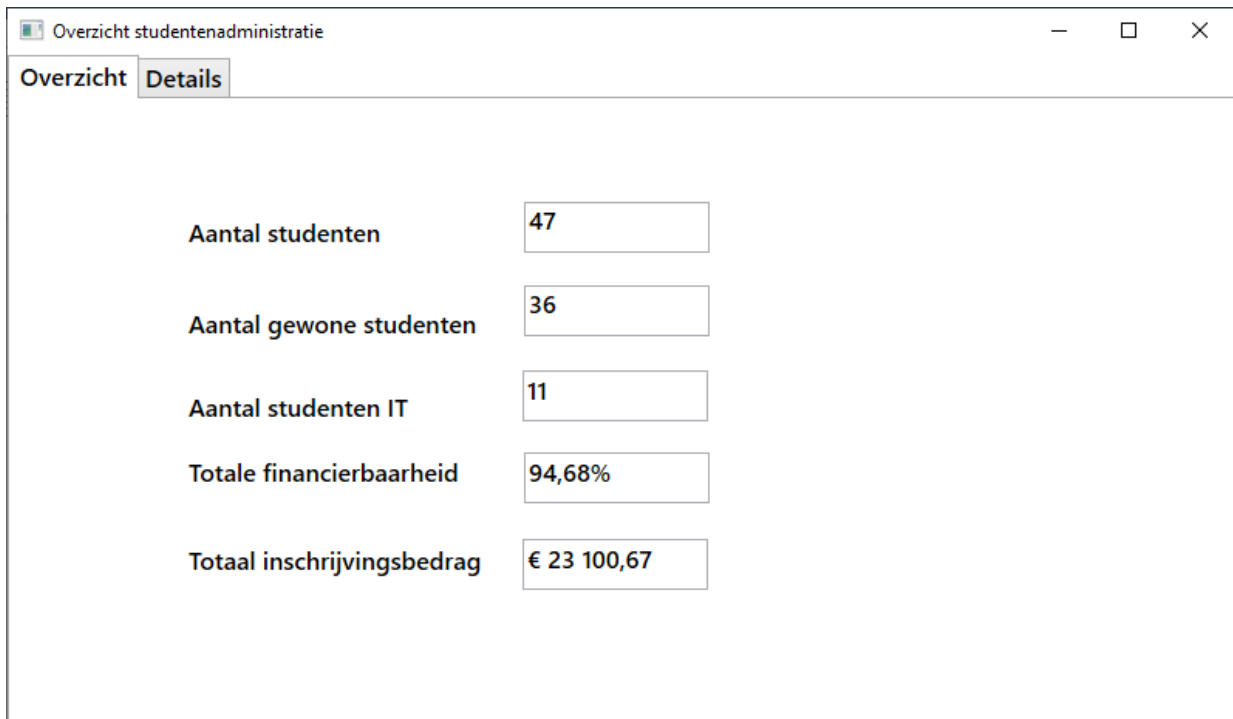
Er is ook een opdrachtknop **BtnToevoegen**. Bij het drukken op deze knop wordt eveneens het venster **INGAVEFORM** opgeroepen. (type student op “Gewoon student”, postcode op “3500”

en geslacht op "M"). De gebruiker kan nu de gegevens van de nieuwe student ingeven en indien gewenst bewaren (OK-knop) of vergeten (CANCEL-knop).

In geval van bewaring, worden de gegevens van de nieuwe student in het geheugen bewaard en toegevoegd in de lijst LstStudentenlijst. (wijzigingen in het tekstbestand gebeuren niet automatisch. Zie eerder)

Tenslotte is er in het venster Studentenverwerking_start een opdrachtknop **BtnOverzicht** om een 2^{de} (modaal) venster **Resultaten** op te roepen. Dit 2^{de} venster geeft een overzicht van de studenten in 2 tabbladen.

Tabblad overzicht:

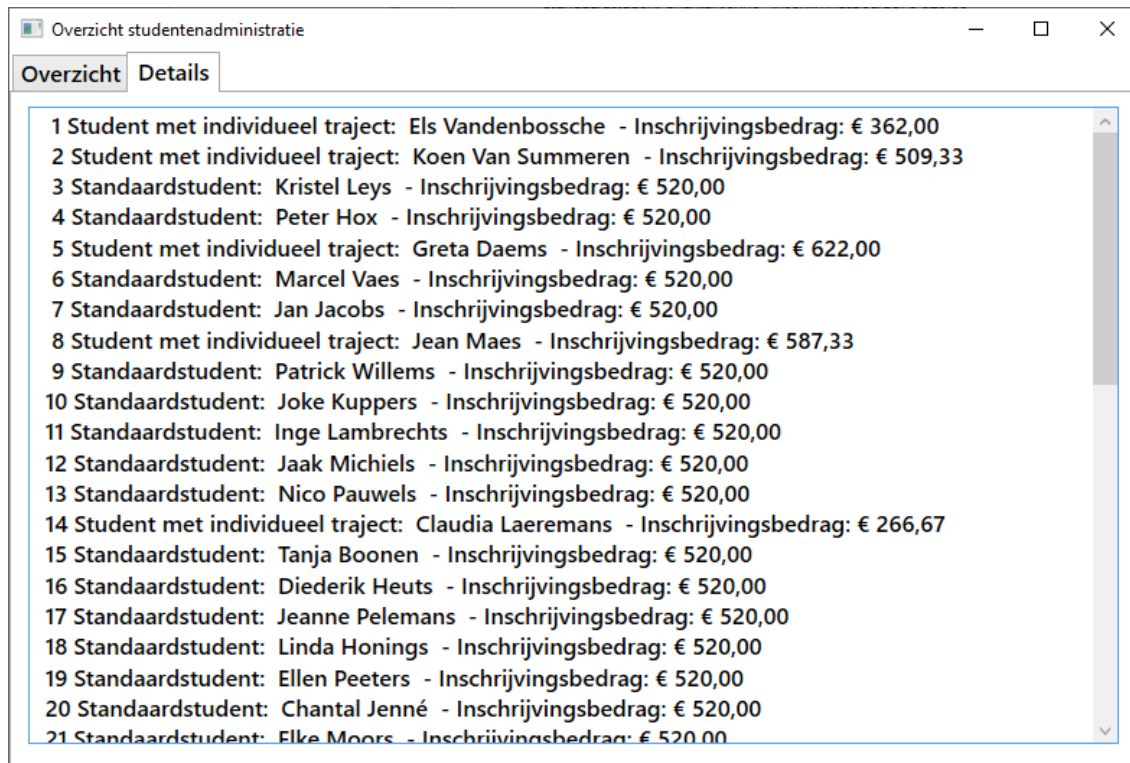


Overzicht studentenadministratie	
Overzicht	Details
Aantal studenten	47
Aantal gewone studenten	36
Aantal studenten IT	11
Totale financierbaarheid	94,68%
Totaal inschrijvingsbedrag	€ 23 100,67

Dit tabblad bevat 5 (niet wijzigbare) tekstvakken:

- **TxtAantStud:** geeft het aantal studenten weer.
- **TxtAantGewoon:** geeft het aantal gewone studenten weer.
- **TxtAantIT:** geeft het aantal IT-studenten weer.
- **TxtSomFinancierbaarheid:** geeft de som van percentages van financierbaarheid van alle studenten samen weer. Deze som wordt door 100 gedeeld. Het bedrag 44,5 in het schermvoorbeeld betekent dus dat de in werkelijkheid 47 ingeschreven studenten samen eigenlijk maar voor 44,5 studenten volledig financierbaar zijn. (omdat de IT-studenten dikwijls niet voor 100% tellen)
- **TxtSomInschrijvingsbedrag:** geeft de som van de inschrijvingsbedragen van alle studenten weer.

Tabblad details:



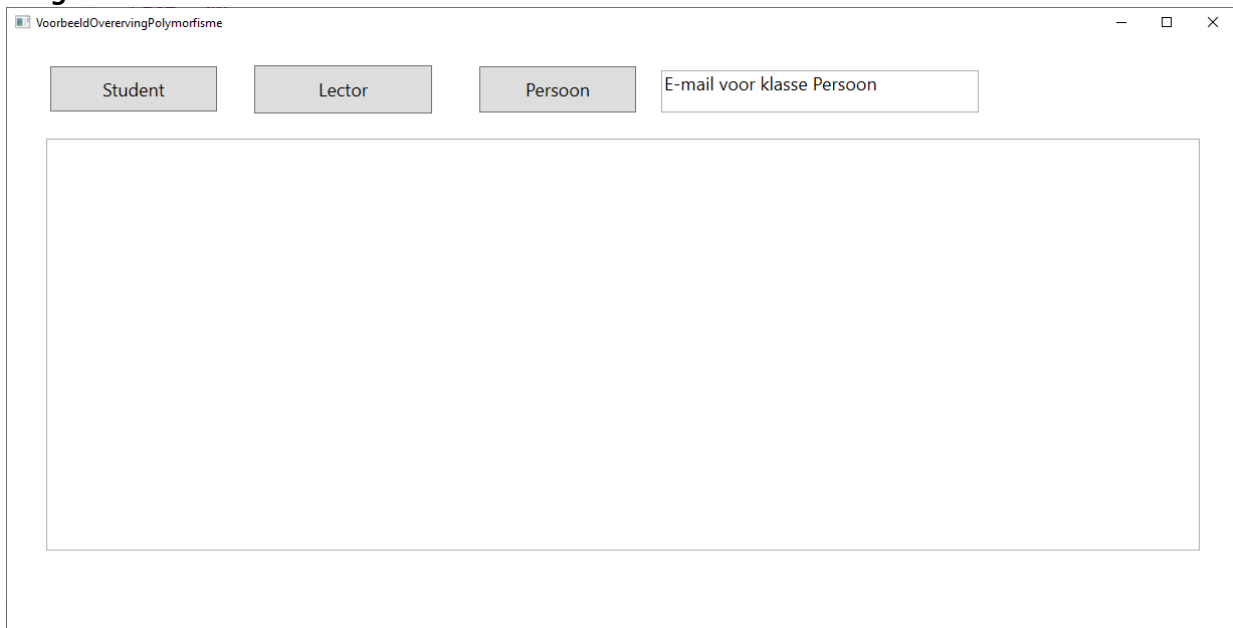
Dit tabblad bevat het (niet wijzigbare) tekstvak **TxtDetails**. Dit tekstvak geeft per student het resultaat van de functie `NaamVolledig`, gevolgd door het inschrijvingsbedrag van de betreffende student.

Opmerking:

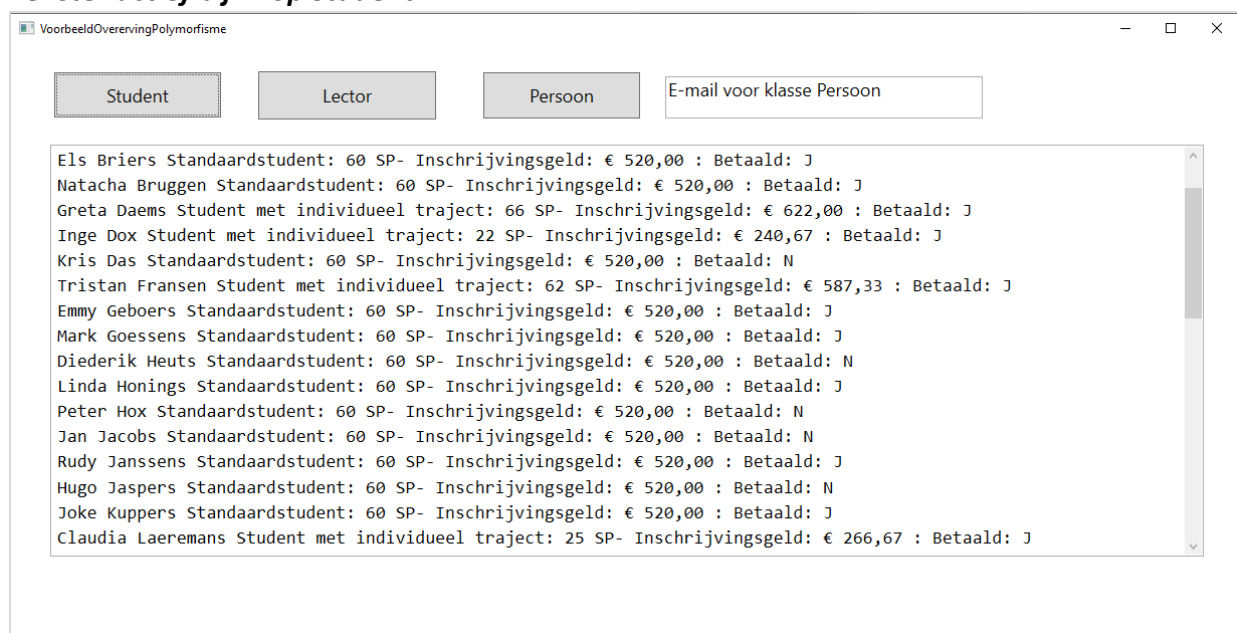
Gebruik een list met studentenobjecten om de gegevens die ingelezen zijn te bewaren!

Maak een toepassing om de gegevens van *studenten*, *lectoren* of je eigen *persoonlijke gegevens* op te vragen. We werken zoveel mogelijk met klassen.

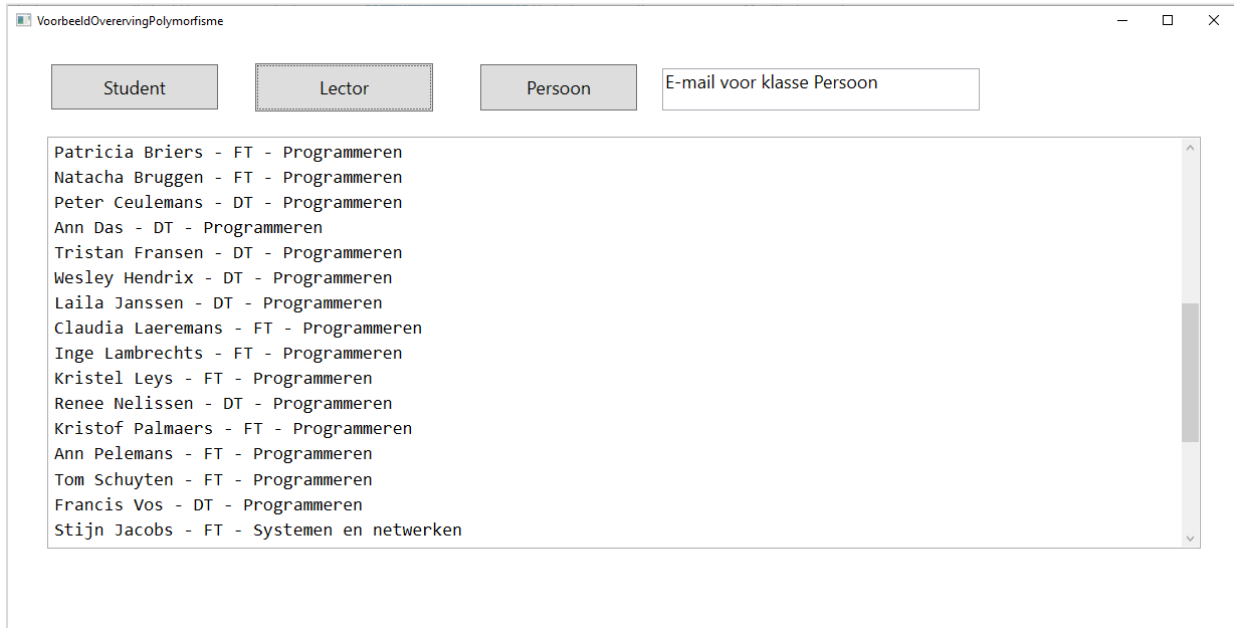
Leeg venster



Venster actief bij knop Student



Venster actief bij knop Lector



De toepassing maakt gebruik van een bestanden *studenten.csv* en *lectoren.csv* waarin de studentengegevens en de gegevens van de lectoren zijn opgenomen.

Hieronder zie je een stuk van **studenten.csv** waarin je de volgende kolommen zie:

- Naam
- Voornaam
- Straat
- Postcode
- Betaald: J of N
- Afdelingscode: SNE, PRO, DVG en IOT. De codes bevatten achtereenvolgens de afdelingen: Systemen en netwerken, Programmeren, Digitale vormgeving en Internet of things.
- TypeStudent: G of I waarbij G een gewone student is die een normaal traject volgt en I een student is met een individueel traject.
- Studiepunten die enkel voor studenten met een individueel traject voorkomen.

```
1 Appeltans;Maurice;Eikestraat 5;3800;N;SNE;G;  
2 Boonen;Tanja;Krekelstraat 4;3700;J;PRO;G;  
3 Brants;Marc;Rietjesweg 37;3700;N;IOT;G;  
4 Briers;Els;Lentestraat 56;3500;J;DVG;G;  
5 Bruggen;Natacha;Vissegatstraat 60;3800;J;SNE;G;  
6 Daems;Greta;Sparrenhof 28;3290;J;PRO;I;66  
7 Dox;Inge;Shopping 3;3600;J;DVG;I;22  
8 Das;Kris;Kroonwinningstraat 31;3500;N;DVG;G;  
9 Fransen;Tristan;Sparstraat 12;3800;J;PRO;I;62  
10 Geboers;Emmy;Leopoldlaan 12;3500;J;IOT;G;
```

Het bestand *lectoren.csv* bevat de volgende kolommen:

- Naam
- Voornaam

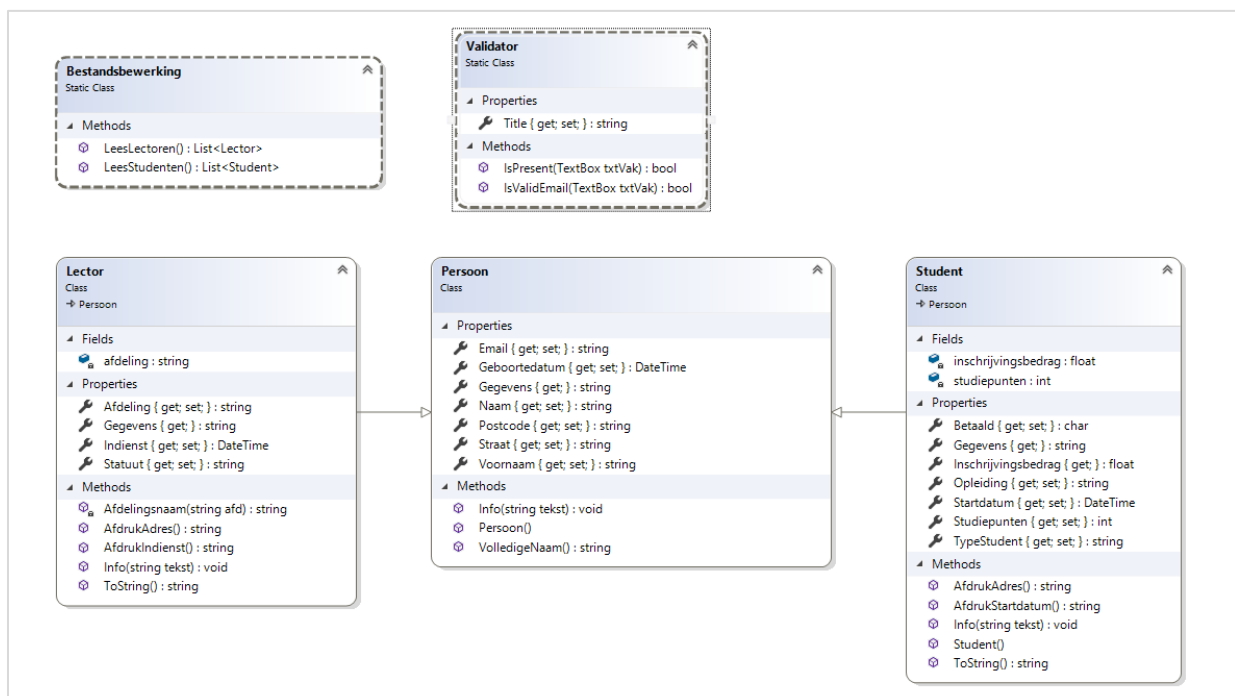
- Geboortedatum
- Straat
- Postcode
- Statuut: DT of FT (deeltijds of fulltime)
- Afdelingscode: SNE, PRO, DVG en IOT. De codes bevatten achtereenvolgens de afdelingen: Systemen en netwerken, Programmeren, Digitale vormgeving en Internet of things.
- Indienst: startdatum dat de lector begonnen is.

```

17 Beulen;Jasper;7/09/1988;Sparstraat 12;3800;FT;PRO;1/09/2013
18 Briers ;Patricia;25/08/1986;Krekelstraat 4;3700;FT;PRO;1/09/2011
19 Bruggen;Natacha;8/09/1991;Putstraat 6;3600;FT;PRO;1/09/2016
20 Ceulemans;Peter;24/08/1985;Djef Antenstraat 56;3500;DT;PRO;1/09/2010
21 Das ;Ann;7/09/1989;Grote baan 47;3500;DT;PRO;1/09/2014
22 Fransens;Tristan;24/08/1995;Putsteeg 5;3000;DT;PRO;1/09/2020
23 Hendrix;Wesley;8/09/1975;Kerkstraat 14;3290;DT;PRO;1/09/2000
24 Janssen ;Laila;22/08/1990;Krekelstraat 11;3500;DT;PRO;1/09/2015
25 Laeremans;Claudia;7/09/1992;Runkstersteenweg 12;3500;FT;PRO;1/09/2017
26 Lambrechts;Inge;21/08/1993;Lange straat 365;3500;FT;PRO;1/09/2018
27 Leys;Kristel;7/09/1994;Lelielaan 4;3600;FT;PRO;1/09/2019
28 Nelissen ;Renee;20/08/1973;Valleilaan 4;3290;DT;PRO;1/09/1998
29 Palmaers;Kristof;8/09/1987;Sparrenhof 28;3290;FT;PRO;1/09/2012

```

We vertrekken vanuit de klasse **PERSOON**. De klassen **LECTOR** en **STUDENT** erven de basiseigenschappen en methods over. Daarnaast maken we ook gebruik van 2 statische klassen **BESTANDSBEWERKING** en **VALIDATOR**.



Klasse Persoon

In bovenstaande klasse diagram zie je de eigenschappen terug.

De eigenschap **Gegevens** moet overgeërfd kunnen worden en geeft de VolledigeNaam() en geboortedatum (lange datumnotatie).

De functiemethode **VolledigNaam()** geeft voornaam en naam terug. Deze moet men ook kunnen overerven.

De void-methode **Info(string tekst)** geeft een berichtenvenster terug met de persoonlijke gegevens.



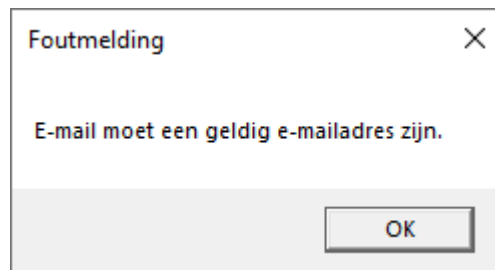
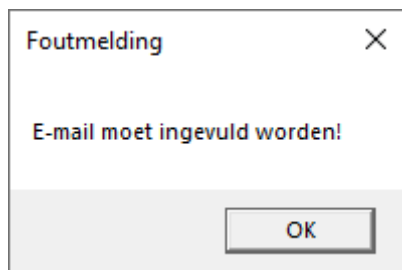
In de lege constructor **Persoon()** kan je je persoonlijke gegevens toekennen. Het e-mailadres wordt wel ingelezen uit het tekstvak **TxtEmail**. Hierbij wordt gebruik gemaakt van de klasse **Validator** om het tekstvak te testen.

Klasse Validator

Deze statische klasse bevat 2 methods om te testen of het tekstvak TxtEmail niet leeg is/correct vooraleer je het e-mailadres doorgeeft aan de eigenschap Email uit de klasse Persoon.

De booleaanse functiemethode **IsPresent(tekstvak)** controleert of het tekstvak niet leeg is. Geeft *False* wanneer het tekstvak leeg is en *True* wanneer er gegevens zijn.

De booleaanse functiemethode **IsValidEmail(tekstvak)** controleert of er in de gegevens een '@' en '.' aanwezig. Geeft *False* wanneer het e-mailadres foutief is en *True* wanneer het e-mailadres correct is.



Klasse Bestandsbewerking

Deze statische klasse bevat 2 functiemethoden die beiden een List teruggeven.

LeesLectoren() leest het bestand *lectoren.csv* in en geeft een List<Lector> terug.

LeesStudenten() leest het bestand *studenten.csv* in en geeft een List<Student> terug.

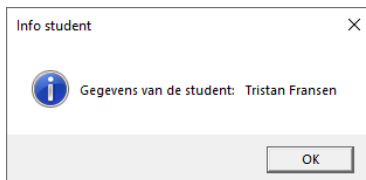
Klasse Student

De klasse bevat bijkomende eigenschappen:

- Startdatum: huidige dag
- Betaald: J of N
- TypeStudent: I of G
- Opleiding: SNE, PRO, DVG of IOT
- Studiepunten: moeten tussen 0 en 99 liggen
- Inschrijvingbedrag: voor de gewone student is dit € 520 en voor de individuele trajecten bedraagt het $50 + (8.66667 * \text{studiepunten})$
- Overgeërfde eigenschap Gegevens die de eigenschap VolledigeNaam() uit klasse Persoon + TypeStudent + Betaald + Inschrijvingsbedrag geeft.

De klassen bevat eveneens een aantal methods:

- Overgeërfde void-method **Info(string tekst)** dat onderstaande info teruggeeft.



- **AfdrukStartdatum():** geeft eigenschap VolledigeNaam() uit klasse Persoon + Startdatum
- **AfdrukAdres():** geeft VolledigeNaam() uit klasse Persoon + Straat + Postcode
- Over te erven **ToString():** Geeft TypeStudent = I onderstaande boodschap
Student met individueel traject: 62 SP- Inschrijvingsgeld: € 587,33 : Betaald: J
en bij TypeStudent = G de volgende boodschap:
Standaardstudent: 60 SP- Inschrijvingsgeld: € 520,00 : Betaald: N
- Constructor Student() dat de huidige datum aan de Startdatum toekent.

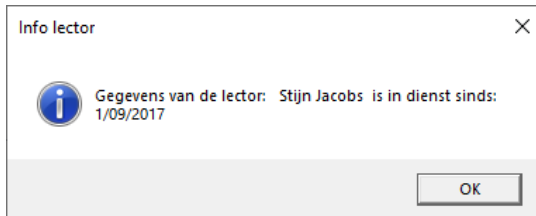
Klasse Lector

De klasse bevat bijkomende eigenschappen:

- Statuut: DT of FT (deeltijds of fulltime)
- Afdeling: ontvangt SNE, PRO, DVG of IOT maar geeft de volledige afdelingsnaam terug (Systemen en netwerken, Programmeren, Digitale vormgeving en Internet of things).
- Indienst: startdatum dat de lector begonnen is.
- Overgeërfde eigenschap Gegevens die de eigenschap VolledigeNaam() uit klasse Persoon + Statuut + Afdeling geeft.

De klassen bevat eveneens een aantal methods:

- Overgeërfde void-method **Info(string tekst)** dat onderstaande info teruggeeft.



- **AfdrukIndienst():** geeft eigenschap VolledigeNaam() uit klasse Persoon + "is in dienst sinds: " + Indienst (kort datumnotatie)
- **AfdrukAdres():** geeft VolledigeNaam() uit klasse Persoon + Straat + Postcode
- Over te erven **ToString():** Geeft Statuut = FT onderstaande boodschap

Patricia Briers Lector is voltijds actief

en bij Statuut = DT de volgende boodschap:

Renee Nelissen Lector is deeltijds actief

Toepassing 17 : **ABankrekening**

Een bank biedt de klanten 2 typen rekeningen aan: een gewone zichtrekening en een spaarrekening. Deze twee typen rekeningen bieden voor een deel dezelfde faciliteiten, maar er zijn ook verschillen. De gemeenschappelijke faciliteiten zijn:

- naam en adres vastleggen;
- een rekening met een beginsaldo openen;
- een overzicht van het huidige saldo bijhouden en laten zien;
- methoden om een geldbedrag te storten en op te nemen.

De zichtrekening kan niet in het rood staan in tegenstelling tot de spaarrekening. De rente voor een zichtrekening bedraagt 0,5% en voor een spaarrekening 1,5%. Voor een spaarrekening wordt jaarlijks beheerskosten van €100,00 aangerekend.

Bankrekening

Overzicht bankrekening

Storting(+)/Opname(-)

500

Zichtrekening

PXL Digital | Campus PXL Hasselt, Elfde-Liniestraat 26, 3500 Hasselt

Huidig saldo: € 2.000,00

Rente: € 10,00

Spaarrekening

PXL Digital | Campus PXL Hasselt, Elfde-Liniestraat 26, 3500 Hasselt

Huidig saldo: € 9.500,00

Rente: € 42,50

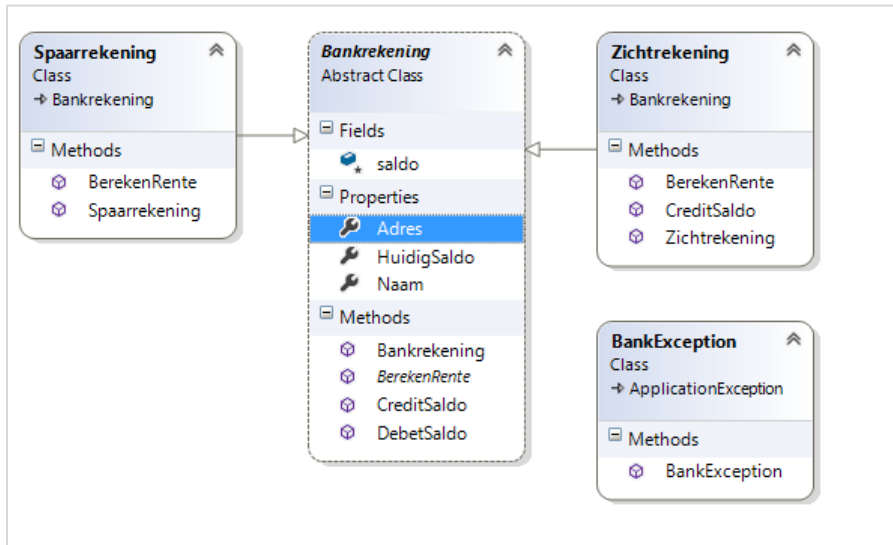
Saldo

Rente

Afsluiten

- *TxtBedrag*: bij een positief bedrag wordt het bedrag op beide rekeningen gestort, bij een negatief bedrag wordt het bedrag bij beide rekeningen afgenomen.
- *BtnSaldo*: geeft het saldo weer van de zichtrekening en spaarrekening.
- *BtnRente*: geeft de rente weer van de zichtrekening en spaarrekening.
- *BtnSluiten*: sluit de toepassing.

Maak voor deze toepassing een **abstracte klasse Bankrekening** en 2 afgeleide klassen: **Spaarrekening** en **Zichtrekening**.



- De klasse **Bankrekening** bevat de volgende methoden:
 - ✓ Constructor **Bankrekening(double opening, string name, string address)**.
 - ✓ **BerekenRente()** dat het over te erven method omschrijft.
 - ✓ **CreditSaldo** bepaalt het saldo over nul en geeft het aan **HuidigSaldo**.
 - ✓ **DebetSaldo** bepaalt het positief saldo en geeft het aan **HuidigSaldo**.
- De klasse **Spaarrekening** bevat de methoden:
 - ✓ **BerekenRente()** berekent de rente van de spaarrekening.
 - ✓ Constructor **Spaarrekening**.
- De klasse **Zichtrekening** bevat de methoden:
 - ✓ **BerekenRente()** berekent de rente van de spaarrekening.
 - ✓ Constructor **Zichtrekening**.
 - ✓ **CreditSaldo()** test of het saldo ontoereikend is en genereert een exception van de klasse **BankException**.
- De klasse **BankException** bevat de constructor.

Toepassing 18 : **AWerknemer**

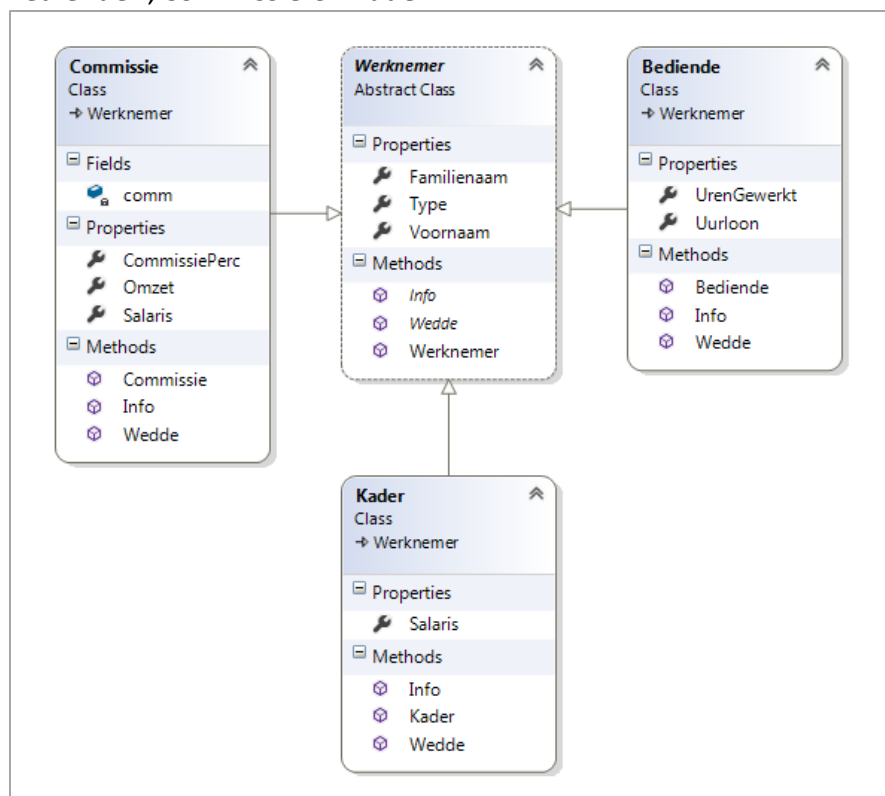
Deze toepassing maakt gebruik van een tekstbestand *Werknemers.txt* waarin de gegevens van het kader, bedienden en het personeel dat op commissie werkt, zijn opgenomen.

1	Robin	Alleweireldt	K	005500,00		
2	Steve	Appeltants	B		021,00	0161,50
3	Rose-Marie	Arnaults	K	005550,50		
4	Anse	Cuppens	C	001500,00	05,00	050000,00
5	Yves	Caelen	C	001500,00	05,00	002500,00
6	Veerle	Deleeuw	B		022,50	0185,00
7	Sven	Kelchtermans	B		018,00	0162,00
8	Laura	Lopez	B		018,50	0163,50
9	Jochem	Pijls	C	001450,00	04,50	007500,00
10	Marianne	Thys	K	005500,00		
11	Paula	Vandeput	B		017,50	0160,00

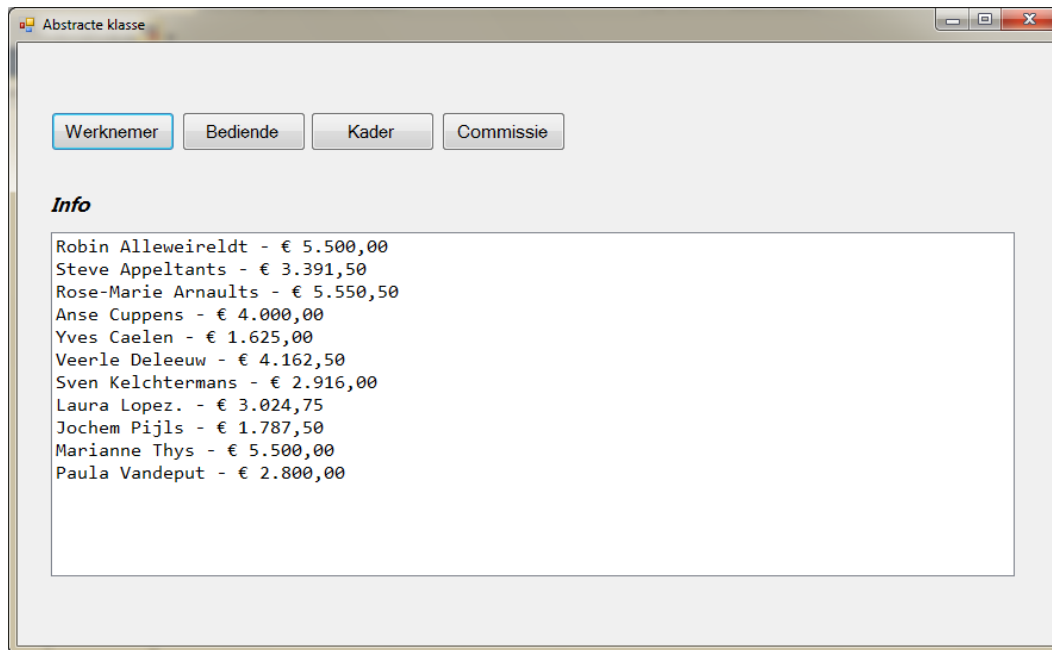
De verschillende kolommen hebben respectievelijk een vaste lengte van: 15, 20, 2, 9, 7, 9, 6 en 9 karakters lang.

Je mag er van uitgaan dat bij deze structuur nooit fouten zijn en dat het bestand aanwezig is op de door u bepaalde locatie. Het aantal records in het tekstbestand is niet vast.

Maak voor deze toepassing een **abstracte klasse *Werknemer*** en 3 afgeleide klassen: ***Bedienden***, ***Commissie*** en ***Kader***.



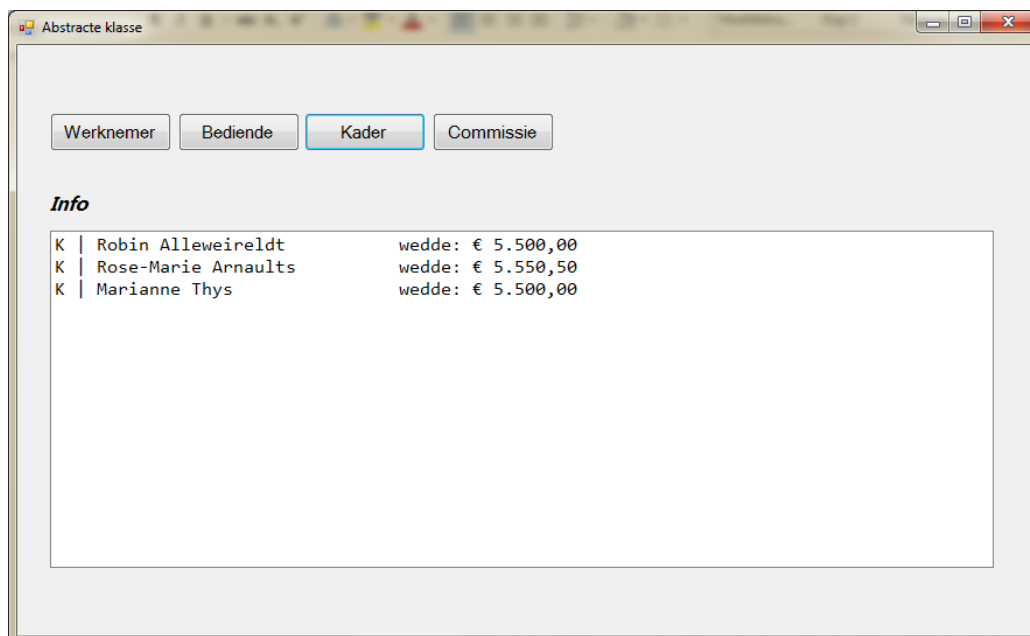
Bij de knop *Werknemer* worden alle records uit het bestand *Werknemers.txt* gelezen en bewaard in *List<Werknemer>*.



De klasse **Werknemer** omvat 2 abstracte methoden die in de afgeleide klassen overgeërfd en voor de betreffende werknemers moet uitgewerkt worden.

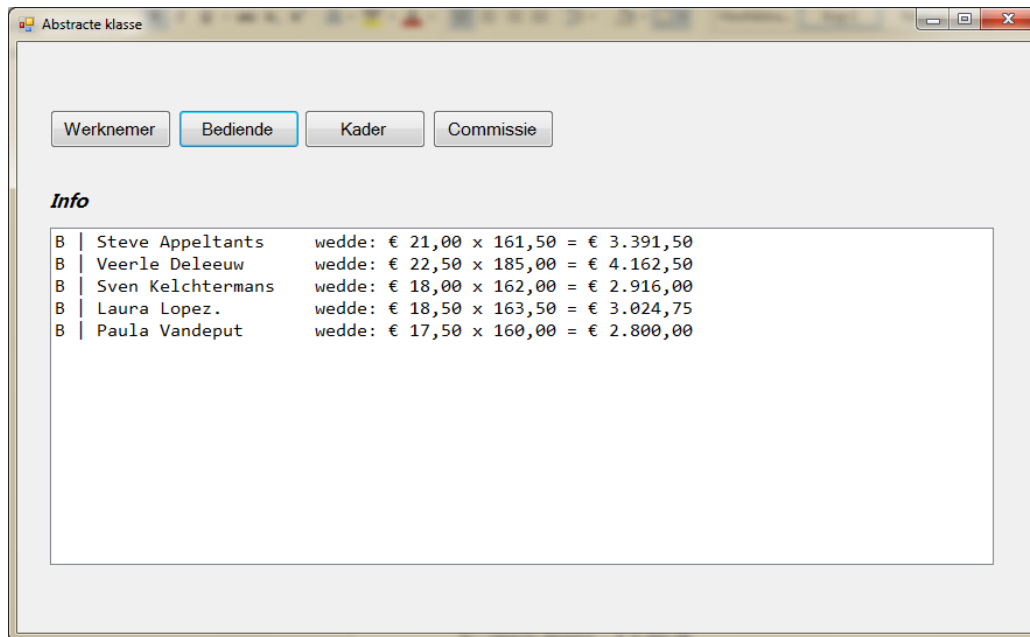
Klasse **Kader** omvat:

- *Wedde()* geeft salaris uit tekstbestand.
- *Info()* geeft onderstaande afdruk.



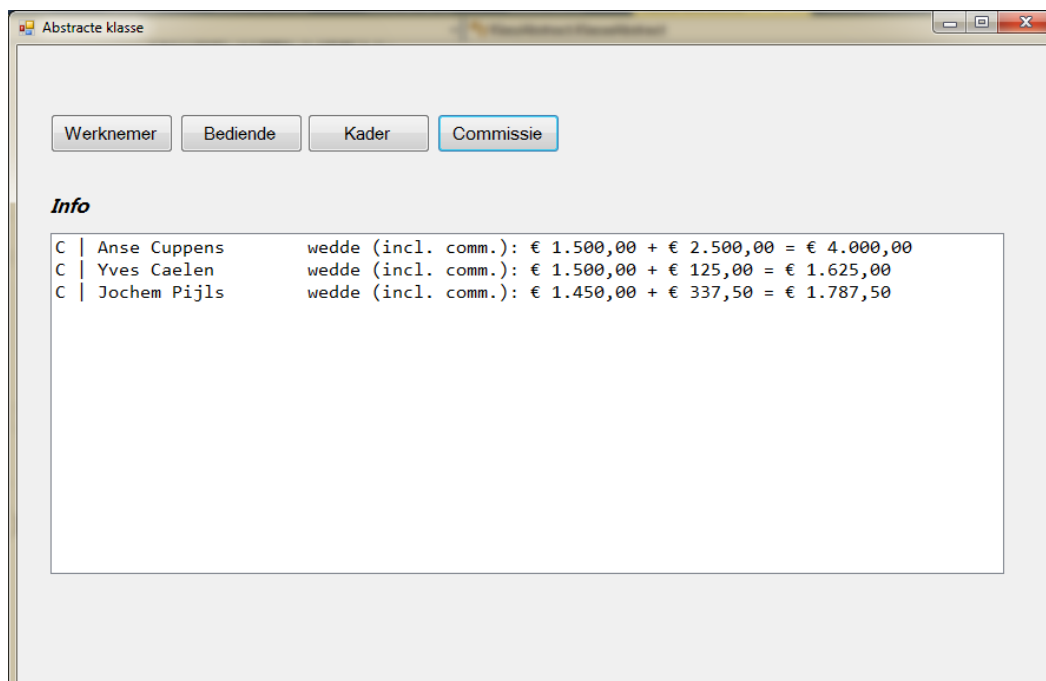
Klasse **Bedienden** omvat:

- *Wedde()* berekent uurloon * aantal gewerkte uren.
- *Info()* geeft onderstaande afdruk.

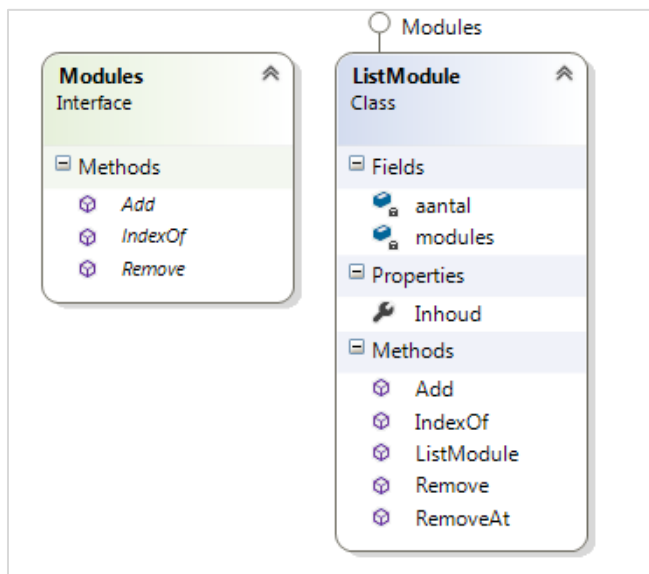


Klasse **Commissie** omvat:

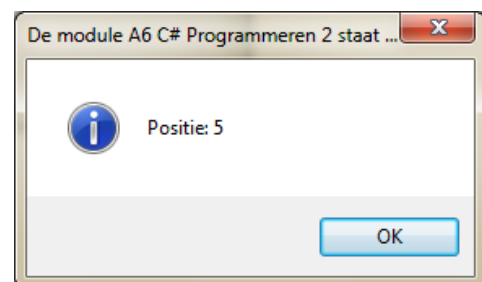
- *Wedde()* berekent salaris + commissie.
- *Info()* geeft onderstaande afdruk.



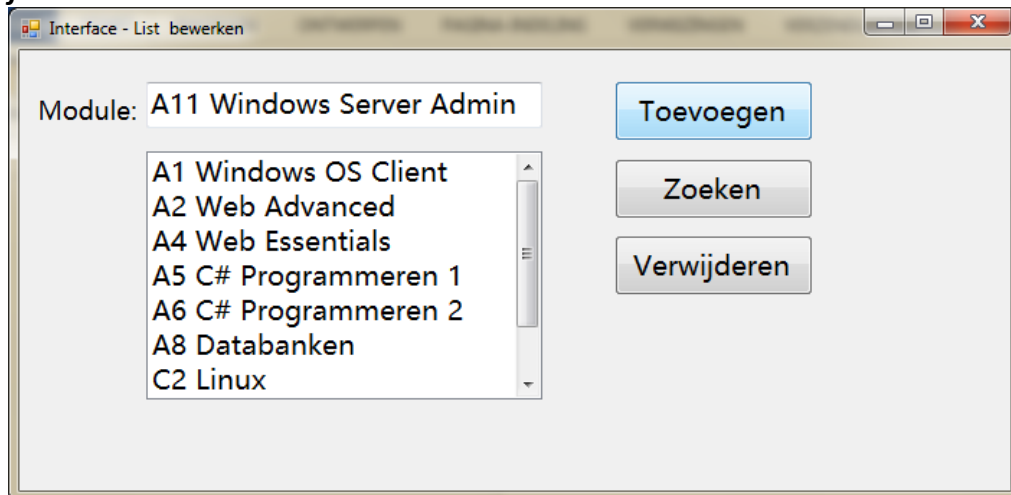
Maak voor deze toepassing een Interface **ListModules** en een klasse **Modules** waarin de eigenschappen van ListModules opgenomen zijn.



- De interface **Modules** omschrijft de methode *Add()*, *Indexof()* en *Remove()*.
- Klasse **ListModule** omvat een *List<string>* dat door de methoden bewerkt kan worden.
 - ✓ De methode *Add()* voegt de tekst uit *TxtModule* toe aan de *List<>* en aan de listbox *LstModules*.
 - ✓ De methode *RemoveAt()* verwijdert de tekst uit de *LstModules* en uit de *List<>* dat opgegeven is in *TxtModule*.
 - ✓ De methode *Remove()* roept *RemoveAt()* op om de tekst te verwijderen.
 - ✓ De methode *IndexOf()* bepaalt de positie (zie berichtenvenster) van de module in de *LstModules* dat opgegeven is in *TxtModule*.
 - ✓ De eigenschap *Inhoud* geeft de *List<>* terug.



frmList



- *btnToevoegen* maakt gebruik van `ListModules.Add()`.
- *btnZoeken* gebruikt `ListModules.IndexOf()`.
- *btnVerwijderen* gebruikt `ListModules.Remove()`.

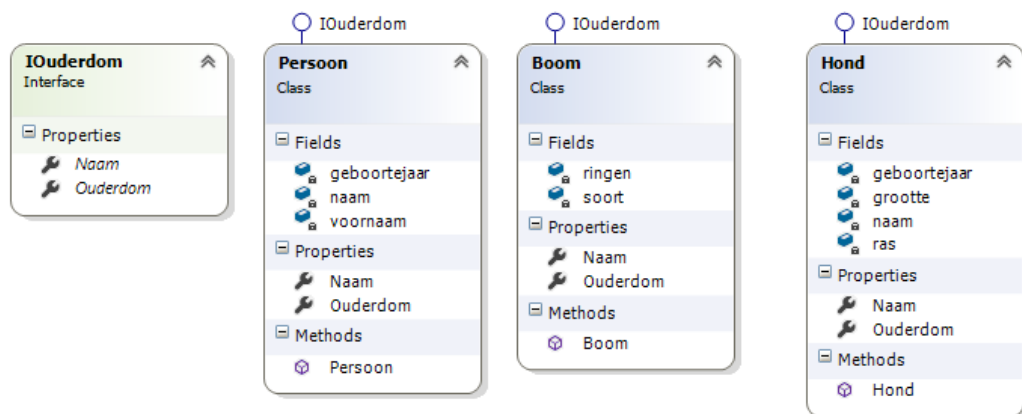
Toepassing 20 : IOuderdom

Deze toepassing maakt gebruik van een tekstbestand *Hondenras.txt* waarin de gegevens van verschillende hondenrassen, de grootte en het te vermenigvuldigen equivalent om de mensenleeftijd te berekenen, zijn opgenomen.

```
1 Beagle;K;12,5
2 Border Terrier;K;12,5
3 Boxer;G;9
4 Bulldog;MG;10,5
5 Cavalier King Charles;K;12,5
6 Chihuahua;K;12,5
7 Cocker Spaniel;K;12,5
8 Duise Herder;G;9
9 Dwerg Schnauzer;K;12,5
10 Franse Bulldog;K;12,5
11 Golden Retriever;MG;10,5
12 Labrador Retriever;MG;10,5
13 Lhasa Apso;K;12,5
14 Mops;K;12,5
15 Shih Tzu;K;12,5
16 Spaniel;MG;10,5
17 Staffordshire Bull Terrier;MG;10,5
18 West Highland White Terrier;K;12,5
19 Whippet;K;12,5
```

De verschillende gegevens zijn gescheiden door middel van een puntkomma. Je mag er van uitgaan dat bij deze structuur nooit fouten zijn en dat het bestand aanwezig is op de door u bepaalde locatie. Het aantal records in het tekstbestand is niet vast.

Maak voor deze toepassing een Interface **IOuderdom** en 3 klassen: **Hond**, **Persoon** en **Boom** waarin de eigenschappen van IOuderdom opgenomen zijn.



The screenshot shows a window titled 'Interface' with three distinct input sections:

- Leeftijd persoon:**
 - Voornaam:
 - Familienaam:
 - Geboortejaar:
 -
- Leeftijd Hondenras:**
 - Ras:
 - Grootte:
 - Naam:
 - Geboortejaar:
 -
- Leeftijd boom:**
 - Soort:
 - Plantjaar:
 -

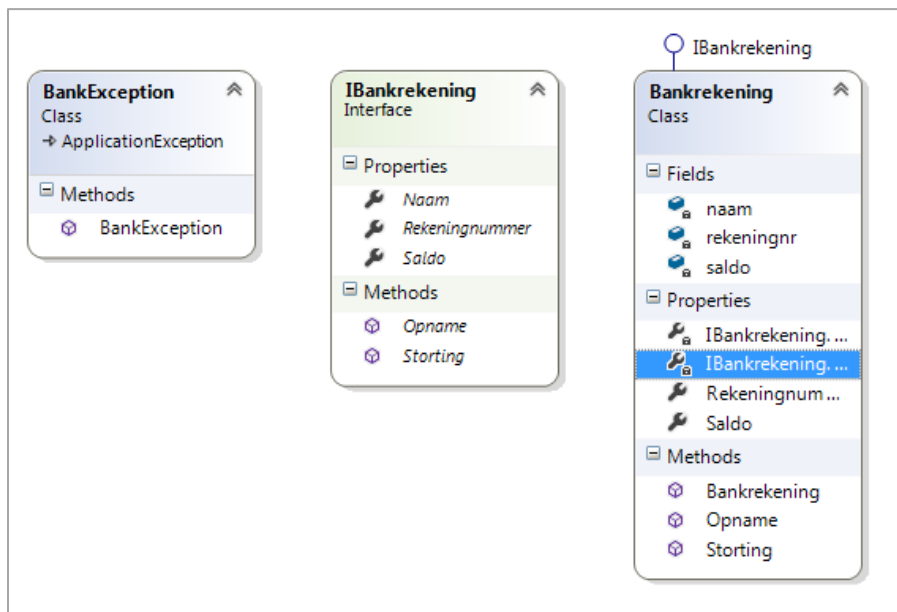
Bij het klikken op de verschillende opdrachtknoppen Ouderdom moet de leeftijd in een berichtvenster worden afgedrukt. De eigenschap **Naam** geeft de titel in het berichtvenster weer en de eigenschap Ouderdom wordt afgedrukt zoals hieronder getoond.



Toepassing 21 : **IBankrekening**

Maak voor deze toepassing een Interface **IBankrekening** en de klasse **Bankrekening** waarin de eigenschappen en de methoden van IBankrekening zijn opgenomen worden uitgewerkt voor de opname (-50 euro) en storting (+100 euro) van geld op de rekening.

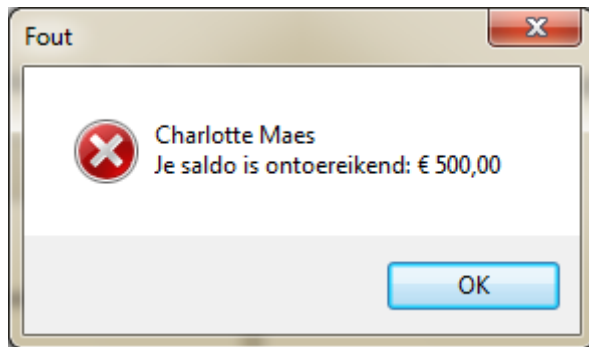
De klasse **BankException** geeft een foutmelding wanneer er meer geld van de rekening wordt afgehaald dan het saldo toelaat.



The screenshot shows a Windows application window titled "Interface bankrekening". It contains a form with the following elements:

- Naam:** A text box containing "Charlotte Maes".
- Rekeningnummer:** A text box containing "001-234567-89".
- Storting(+)/Opname(-):** A text box containing "-750".
- Saldo:** A text box containing "500".
- Bereken saldo:** A button with the text "Bereken saldo".

Fout gegeneert door de klasse BankException wanneer de opname groter is dan het saldo.



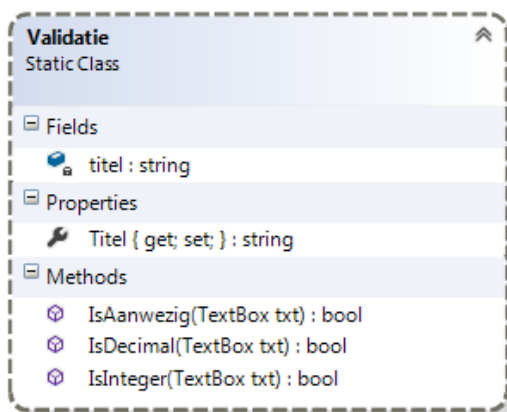
Maak een eigen Class Library **ValidatieLib** dat test of een tekstvak leeg is of het een integer bevat of het een geldig decimaal getal bevat.

De statische klasse heeft:

- een eigenschap **Titel** dat een gepaste titel kan meegeven bij een foutmelding.

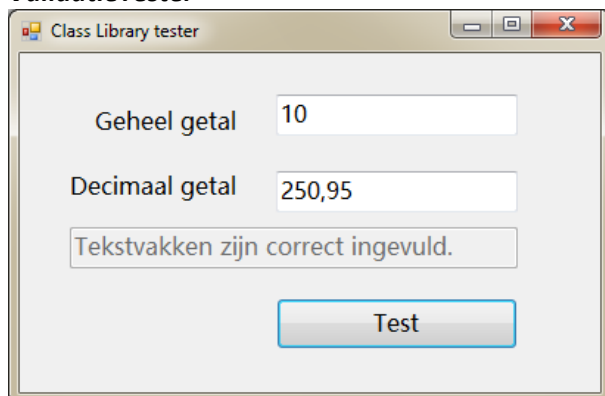
Methods:

- **IsAanwezig**: geeft een booleaanse waarde 'True' als het tekstvak is ingevuld en 'False' indien het tekstvak leeg is. Bij een lege waarde moet bovendien de foutmelding 'is vereist gegeven' komen.
- **IsInteger**: geeft een booleaanse waarde 'True' als het tekstvak een geheel getal heeft en bij 'False' moet de foutmelding 'het getal moet een geheel getal zijn.' komen.
- **IsDecimaal**: geeft een booleaanse waarde 'True' als het tekstvak een decimaal getal heeft en bij 'False' moet de foutmelding 'het getal moet een decimaal getal zijn.' komen.



Voeg je DDL toe in een nieuw project en test uit.

ValidatieTester



Class Library tester

Geheel getal

Decimaal getal

Test

Foutmelding

Geheel getal is een vereist gegeven.

OK

Class Library tester

Geheel getal

Decimaal getal

Test

Foutmelding

Decimaal getal moet een decimaal getal zijn.

OK

Databanken

Toepassing 23 : Dataset Medewerkers

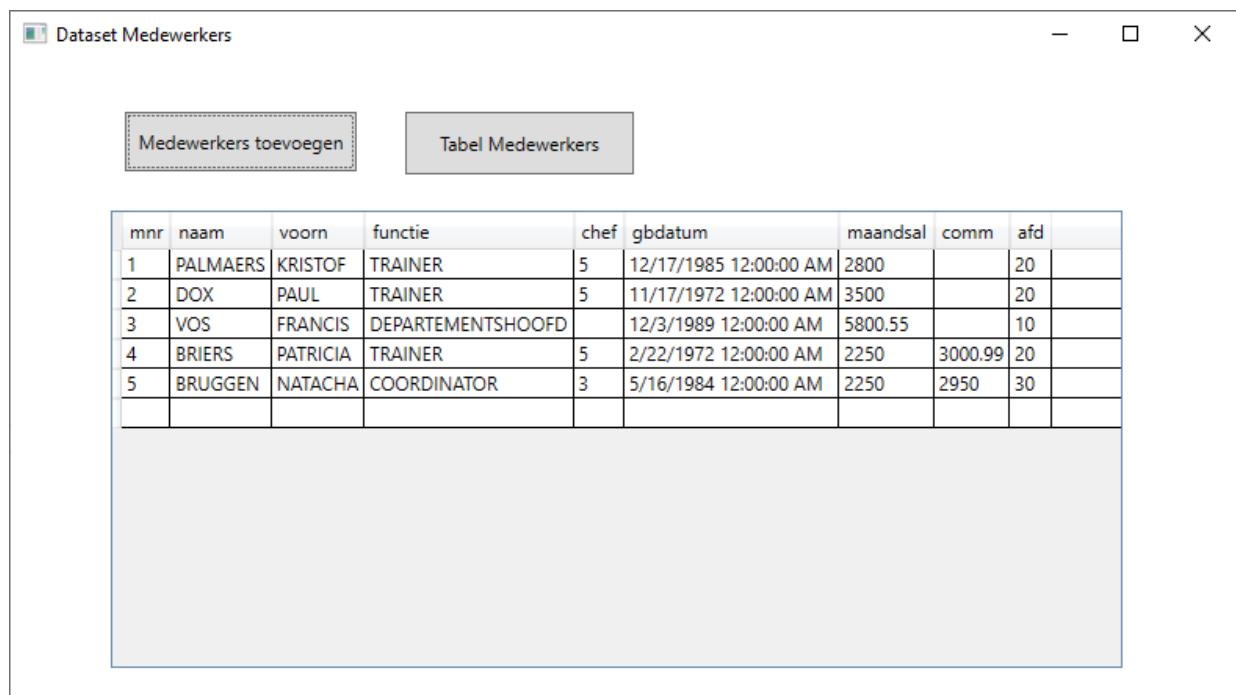
Maak een toepassing waarbij je een volledige dataset maakt dat de datatabel Medewerker bevat.

De datatabel Medewerkers bevat de volgende kolommen:

- **Mnr**: datatype (int)
- **Naam**: datatype (string)
- **Voorn**: datatype (string)
- **Functie**: datatype (string)
- **Chef**: datatype (string)
- **Gbdatum**: datatype (DateTime)
- **Maandsal**: datatype (float)
- **Comm**: datatype (float)
- **Afd**: datatype (int)

Voorzie voor de kolommen *Chef* en *Comm* dat deze kolommen een null-waarden kunnen hebben. De kolom *Mnr* is de primaire sleutel van de datatabel Medewerker.

De knop **BtnToevoegen** voegt onderstaande datatabel toe aan de DataGrid. Gebruik hiervoor de void-methods *MaakTabellen()* en *VulTabellen()*.



mnr	naam	voorn	functie	chef	gbdatum	maandsal	comm	afd	
1	PALMAERS	KRISTOF	TRAINER	5	12/17/1985 12:00:00 AM	2800		20	
2	DOX	PAUL	TRAINER	5	11/17/1972 12:00:00 AM	3500		20	
3	VOS	FRANCIS	DEPARTEMENTSHOOFD		12/3/1989 12:00:00 AM	5800.55		10	
4	BRIERS	PATRICIA	TRAINER	5	2/22/1972 12:00:00 AM	2250	3000.99	20	
5	BRUGGEN	NATACHA	COORDINATOR	3	5/16/1984 12:00:00 AM	2250	2950	30	

De knop **BtnMedTabel** leest met behulp van een OpenFileDialog de csv file *DatMedewerkers* en voegt DataRows toe aan de DataTable Medewerkers.

```
7369;CASPER;JANA;TRAINER;7902;17-12-1985;1800;;20
7499;ALLARD;NELE;VERKOPER;7698;20-02-1981;1600;3000;30
7521;DEFOUR;THOMAS;VERKOPER;7698;22-02-1982;2250;5000;30
7566;JACOBS;EMMA;MANAGER;7839;02-04-1987;4975;;20
7654;MARTENS;RAF;VERKOPER;7698;28-09-1976;2250;3400;30
7698;BRIERS;ANDREA;MANAGER;7839;01-11-1983;5850;;30
7782;CLERCKX;AN;MANAGER;7839;09-06-1985;3450;;10
7788;SWINNEN;CHRIS;TRAINER;7566;26-11-1979;4000;;20
7839;DE KONING;LIEVE;DIRECTEUR;;17-11-1972;7000;;10
7844;DEN RUYTER;JOACHIM;VERKOPER;7698; 28-09-1988;2500;0;30
7876;SLECHTEN;TOM;TRAINER;7788;30-12-1986;2700;;20
7900;JACOBS;SIMON;BOEKHOUDER;7698;03-12-1989;2800;;30
7902;DE COOMAN;DORIEN;TRAINER;7566;13-02-1979;4000;;20
7934;WOUTERS;SVEN;BOEKHOUDER;7782;23-01-1982;2300;;10
```

Dataset Medewerkers

Medewerkers toevoegen Tabel Medewerkers

mnr	naam	voorn	functie	chef	gbdatum	maandsal	comm	afd	
7369	CASPER	JANA	TRAINER	7902	12/17/1985 12:00:00 AM	1800	0	20	
7499	ALLARD	NELE	VERKOPER	7698	2/20/1981 12:00:00 AM	1600	3000	30	
7521	DEFOUR	THOMAS	VERKOPER	7698	2/22/1982 12:00:00 AM	2250	5000	30	
7566	JACOBS	EMMA	MANAGER	7839	4/2/1987 12:00:00 AM	4975	0	20	
7654	MARTENS	RAF	VERKOPER	7698	9/28/1976 12:00:00 AM	2250	3400	30	
7698	BRIERS	ANDREA	MANAGER	7839	11/1/1983 12:00:00 AM	5850	0	30	
7782	CLERCKX	AN	MANAGER	7839	6/9/1985 12:00:00 AM	3450	0	10	
7788	SWINNEN	CHRIS	TRAINER	7566	11/26/1979 12:00:00 AM	4000	0	20	
7839	DE KONING	LIEVE	DIRECTEUR		11/17/1972 12:00:00 AM	7000	0	10	
7844	DEN RUYTER	JOACHIM	VERKOPER	7698	9/28/1988 12:00:00 AM	2500	0	30	
7876	SLECHTEN	TOM	TRAINER	7788	12/30/1986 12:00:00 AM	2700	0	20	
7900	JACOBS	SIMON	BOEKHOUDER	7698	12/3/1989 12:00:00 AM	2800	0	30	
7902	DE COOMAN	DORIEN	TRAINER	7566	2/13/1979 12:00:00 AM	4000	0	20	
7934	WOUTERS	SVEN	BOEKHOUDER	7782	1/23/1982 12:00:00 AM	2300	0	10	

Maak een toepassing waarbij je een volledige DataSet *dsAuto* maakt dat een DataTable *Verkoper* en een DataTable *Artikel* bevat.

De tabel *Verkoper* bevat de volgende kolommen:

- **VerkoperId**: datatype (int) – primary key
- **Naam**: datatype (string)
- **Straat**: datatype (string)
- **Nr**: datatype (string)
- **Chef**: datatype (string)
- **Postcode**: datatype (string)
- **Woonplaats**: datatype (string)
- **Land**: datatype (string)
- **Afd**: datatype (string)

De tabel *Artikel* bevat de volgende kolommen:

- **ArtId**: datatype (int) – primary key
- **VerkoperId**: datatype (int) – vreemde sleutel naar VerkoperId in tabel Verkoper
- **Code**: datatype (string)
- **Beschrijving**: datatype (string)
- **Chef**: datatype (string)
- **Aankoopprijs**: datatype (decimal)
- **Verkoopprijs**: datatype (decimal)

Afhankelijk van de keuzerondjes moet er in het tekstvak *TxtResultaat* een overzicht gegeven worden van het verschillende modellen en hun verkoopprijs van het gekozen merk.

De knop *Listing* geeft een lijst van alle verkoper en alle auto's.

The application window 'Dataset Verkoper' contains the following elements:

- Verkoper Table:**

VerkoperId	Naam	Straat	Nr	Postcode	Woonplaats	Land
10	Elon Musk	Herkenrodesingel	77	3500	Hasselt	België
20	Clotilde Delbos	Gouverneur Roppesingel	2	3500	Hasselt	België
- Artikel Table:**

ArtId	VerkoperId	Code	Beschrijving	Aankoopprijs	Verkoopprijs
1	10	Tesla	Tesla Model S	65000	75500
2	10	Tesla	Tesla Model S 85kWh (Dual Motor)	88400	96250
3	20	Renault	Renault Kadjar	14000	18650
- Buttons:** A 'Listing' button is located at the top right.
- Lists:**
 - LIJST VAN VERKOPERS:** Contains 'Elon Musk' and 'Clotilde Delbos'.
 - LIJST VAN AUTO'S:** Contains 'Tesla Model S', 'Tesla Model S 85kWh (Dual Motor)', and 'Renault Kadjar'.
- Selection Interface:**
 - Radio buttons for 'Tesla' (selected) and 'Renault'.
 - A text area showing the result for the selected brand:

Tesla	Tesla Model S	VP: € 75.500,00
Tesla	Tesla Model S 85kWh (Dual Motor)	VP: € 96.250,00

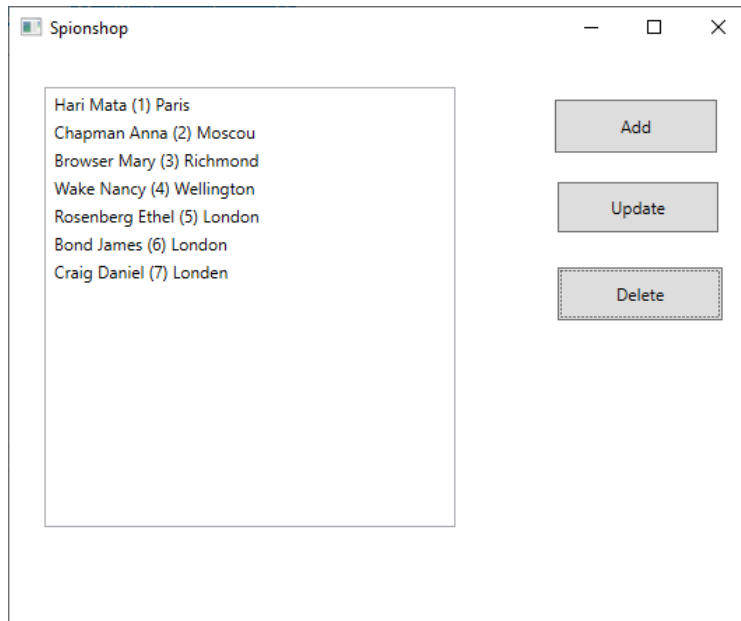
Maak een toepassing om de gegevens in een listbox weer te geven. De database Spionshop bevindt zich op de SQL Server.

Wanneer je de toepassing opstart, wordt onderstaande gegevens getoond.

Met de knop **BntAdd** voeg je een rij toe aan de database Spionshop.

Met de knop **BtnUpdate** worden de gegevens in de toegevoegde rij gewijzigd.

Met de knop **BtnDelete** kan je de laatste toegevoegde rij wissen.



Toepassing 26 : dB Spionshop2

Maak een toepassing om de gegevens in een listbox weer te geven. De database Spionshop bevindt zich op de SQL Server.

Wanneer je de toepassing opstart, wordt onderstaande gegevens getoond.

Bij het opstarten van de toepassing dB Spionshop 2 wordt de gegevens terug ingelezen vanuit SQL Server in de listbox.

The screenshot shows the 'Spionshop' application window. On the left is a listbox containing the following items: 'Hari Mata (1) Paris', 'Chapman Anna (2) Moscou', 'Browser Mary (3) Richmond', 'Wake Nancy (4) Wellington', 'Rosenberg Ethel (5) London', 'Bond James (6) London', and 'Craig Daniel (7) Londen'. In the center are three buttons: 'Add', 'Update', and 'Delete'. On the right are four text input fields: 'Voornaam', 'Naam', 'Geboortedatum', and 'Woonplaats'. Below these are two more fields: 'Gebruikersnaam' and a password field represented by seven dots.

Met de knop **BtnAdd** wordt de gegevens uit de tekstvakken lezen en de rij wordt toegevoegd aan de database Spionshop.

Met de knop **BtnUpdate** wordt de geselecteerde rij uit de listbox in de database gewijzigd.

Met de knop **BtnDelete** wordt de geselecteerde rij uit de listbox uit de database verwijderd.

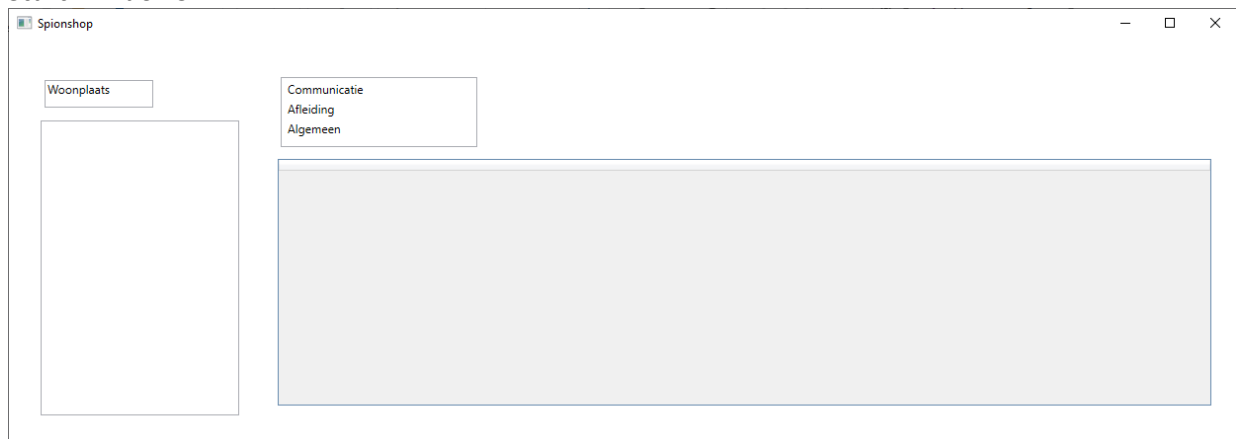
The screenshot shows the 'Spionshop' application window after a new record has been added. The listbox now contains 8 items: 'Hari Mata (1) Paris', 'Chapman Anna (2) Moscou', 'Browser Mary (3) Richmond', 'Wake Nancy (4) Wellington', 'Rosenberg Ethel (5) London', 'Bond James (6) London', 'Craig Daniel (7) Londen', and 'Saenz Manuela (12) Peru'. The 'Add' button is still visible. The input fields on the right are now filled with data: 'Manuela' in 'Voornaam', 'Saenz' in 'Naam', '27-11-1945' in 'Geboortedatum', 'Peru' in 'Woonplaats', 'SaM' in 'Gebruikersnaam', and the password field remains as seven dots. The 'Update' and 'Delete' buttons are also present.

Maak een toepassing om de spionnen op te zoeken uit een bepaalde stad.

De toepassing zoekt ook de verschillende categorieën in de dB Spion en geeft die weer in een listbox. In een DataGrid worden de artikelen uit de geselecteerde categorie geselecteerd. De database Spionshop bevindt zich op de SQL Server.

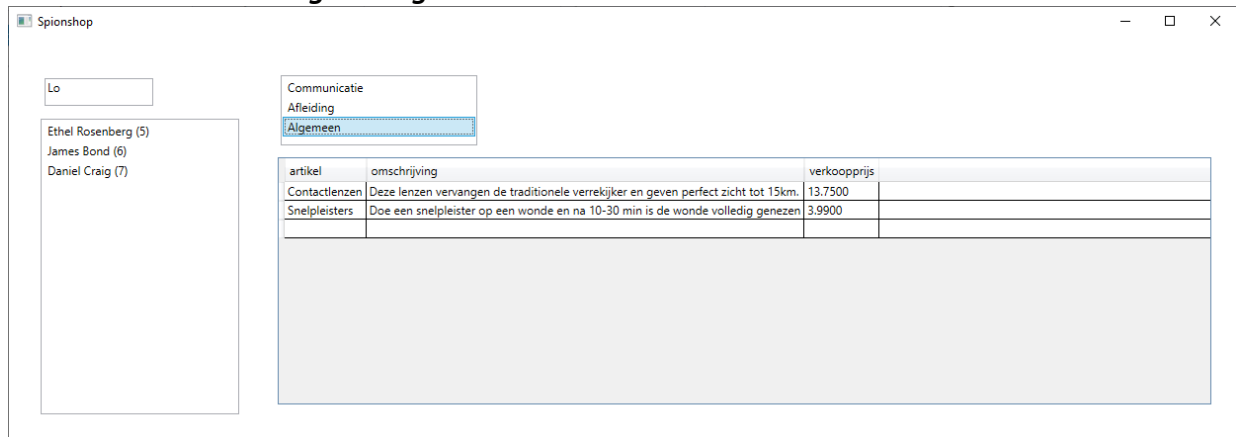
Wanneer je de toepassing opstart, wordt onderstaande gegevens getoond.

Start Windows



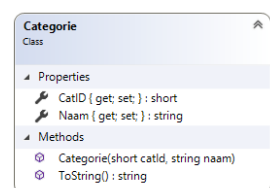
Je kan in het tekstvak *TxtWoonplaats* een plaats opzoeken waar de spionnen actief zijn. Vb. *L...* of *Lo...* of *Lon...* volstaat om de spionnen uit Londen weer te geven in een listbox *LstWoonplaats*. In de listbox *LstCategorie* wordt *Algemeen* geselecteerd en de artikels uit de categorie worden opgehaald en getoond in de DataGrid *DgdArtikel*.

Selectie van Lo en categorie Algemeen.



Programmeervoorwaarden

- Bij de Load wordt gebruikt gemaakt van de klasse *Categorie* en de Stored Procedure *ArtikeluitCategorie* om de verschillende categorieën uit de database op te halen en in de listbox *LstCategorie* toe te voegen.

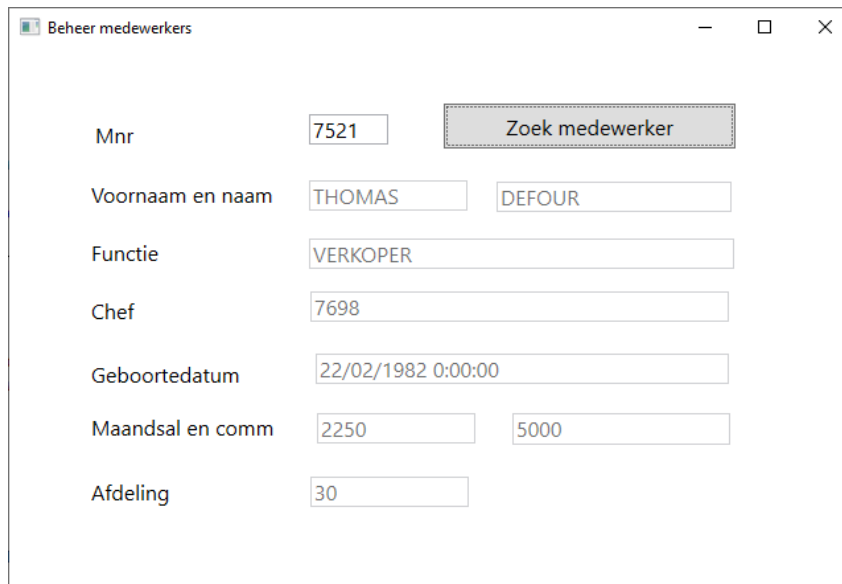


- Maak een Stored Procedure aan in SQL Server om de artikels van een opgegeven categorie op te halen.

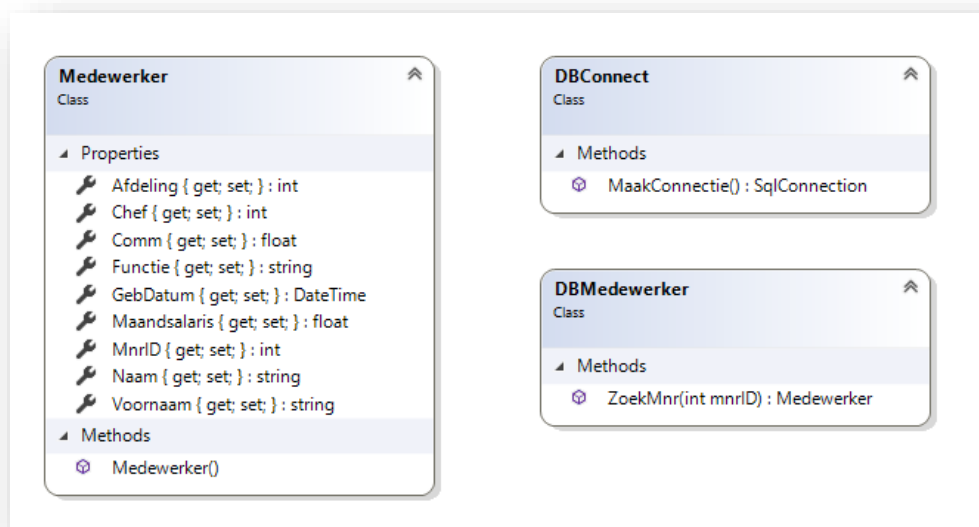
```
// === STORED PROCEDURE ===  
CREATE PROCEDURE[dbo].[ArtikelUitCategorie]  
@CatID smallint  
AS  
SELECT artikel, omschrijving, verkoopprijs  
FROM artikel  
WHERE cat_id = @CatID  
ORDER BY artikel
```

Maak een toepassing om de medewerkers op te zoeken vanuit de database op SQL Server.

Op basis van het ingevoerde medewerkersnummer worden de corresponderende gegevens van de medewerker getoond. Indien het medewerkersnummer niet bestaat dan moet er onderstaand berichtenvenster getoond worden en alle tekstvakken gewist worden.



Maak gebruik van volgende klassen:



De klasse **Medewerker** bevat alle kolomnamen (evenveel eigenschappen) van de tabel Medewerkers.

De klasse **DBConnect** bevat een statische void-method *MaakConnectie()* dat de connectiestring bevat.

De klasse **DBMedewerker** bevat een statische functie *ZoekMnr(int mnrID)* dat een klasse Medewerker met de juiste waarden teruggeeft op basis van het opgezochte medewerkersnummer.

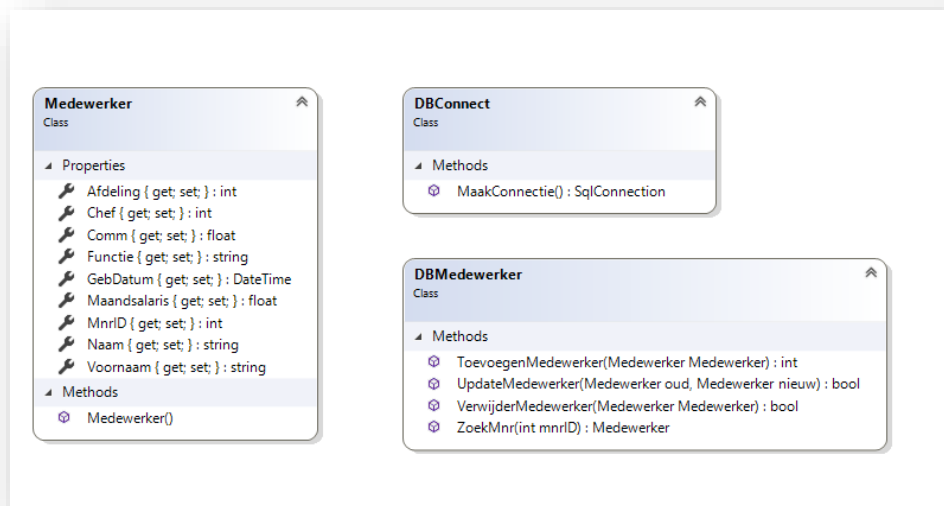
Wanneer er een fout optreedt, moet er een foutmelding gegeven worden. Dus werkt zoveel mogelijk met de Try.. Catch.

db Medewerkers

	mnr	naam	voom	functie	chef	gbdatum	maandsal	comm	afd
1	7369	CASPERS	JANA	TRAINER	7902	1985-12-17	1800.00	NULL	20
2	7499	ALLARD	NELE	VERKOPER	7698	1981-02-20	1600.00	3000.00	30
3	7521	DEFOUR	THOMAS	VERKOPER	7698	1982-02-22	2250.00	5000.00	30
4	7566	JACOBS	EMMA	MANAGER	7839	1987-04-02	4975.00	NULL	20
5	7654	MARTENS	RAF	VERKOPER	7698	1976-09-28	2250.00	3400.00	30
6	7698	BRIERS	ANDREA	MANAGER	7839	1983-11-01	5850.00	NULL	30
7	7782	CLERCKX	AN	MANAGER	7839	1985-06-09	3450.00	NULL	10
8	7788	SWINNEN	CHRIS	TRAINER	7566	1979-11-26	4000.00	NULL	20
9	7839	DE KONING	LIEVE	DIRECTEUR	NULL	1972-11-17	7000.00	NULL	10
10	7844	DEN RUYTER	JOACHIM	VERKOPER	7698	1988-09-28	2500.00	0.00	30
11	7876	SLECHTEN	TOM	TRAINER	7788	1986-12-30	2700.00	NULL	20
12	7900	JACOBS	SIMON	BOEKHOUDER	7698	1989-12-03	2800.00	NULL	30
13	7902	DE COOMAN	DORIEN	TRAINER	7566	1979-02-13	4000.00	NULL	20
14	7934	WOUTERS	SVEN	BOEKHOUDER	7782	1982-01-23	2300.00	NULL	10

Op basis van de vorige toepassing willen we ook de gegevens kunnen toevoegen, wijzigen of verwijderen.

Voeg de knoppen **BtnToevoegen**, **BtnWijzigen** en **BtnVerwijderen** toe. Pas tevens je klasse **DBMedewerker** aan zodat je in je database dB Medewerker gegevens kan toevoegen, verwijderen of wijzigen.

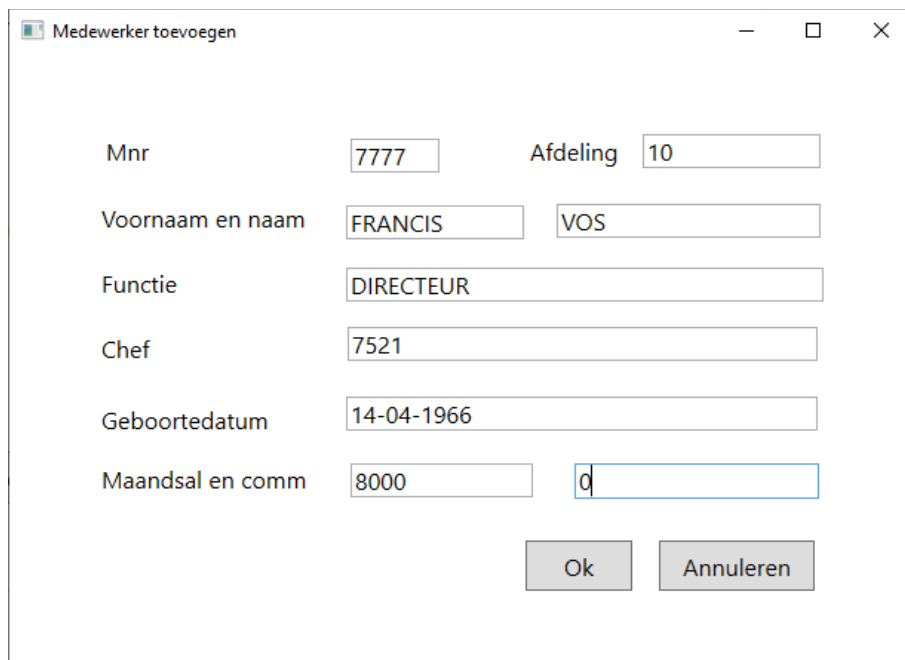


Medewerker met mnr= 7788 zoeken.

The image shows a Windows application window titled "Beheer medewerkers". It contains the following fields and buttons:

- Mnr:** A text box containing "7788" and a button labeled "Zoek medewerker".
- Voornaam en naam:** Two text boxes, one containing "CHRIS" and the other "SWINNEN".
- Functie:** A text box containing "TRAINER".
- Chef:** A text box containing "7566".
- Geboortedatum:** A text box containing "26/11/1979 0:00:00".
- Maandsal en comm:** Two text boxes, one containing "4000" and the other "0".
- Afdeling:** A text box containing "20".
- Buttons:** Four buttons at the bottom: "Toevoegen", "Wijzigen", "Verwijderen", and "Afsluiten".

Medewerker met mnr=7777 toegevoegd



Medewerker toevoegen

Mnr Afdeling

Voornaam en naam

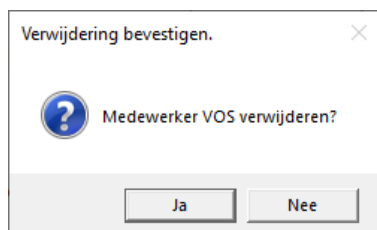
Functie

Chef


Geboortedatum

Maandsal en comm

Bevestiging vragen bij het verwijderen van een medewerker.



Verwijdering bevestigen.

 Medewerker VOS verwijderen?