

C# Adv Console Oefeningen: LINQ

1. Los de volgende vragen op met behulp van een LINQ query.

Gebruik de volgende IEnumerable objecten:

```
List<int> driehoek = new List<int>()
{
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
    1, 3, 6, 10, 15, 21, 28, 36, 45,
    1, 4, 10, 20, 35, 56, 84, 120,
    1, 5, 15, 35, 70, 126, 210,
    1, 6, 21, 56, 126, 252,
    1, 7, 28, 84, 210,
    1, 8, 36, 120,
    1, 9, 45,
    1, 10,
    1
};

string[] tekst = new string[]
{
    "langspeelfilm", "vluchtmisdrijf", "dovemansgesprek", "containerpark",
    "confituur", "inox", "valavond", "enerverend", "lopen", "bank", "vieruurtje",
    "waterzooi", "maaltijdcheque", "arrondissement", "eindtermen", "nieuwjaarsbrief"
};<
```

- Druk al de woorden af in de array, tekst, die langer zijn dan tien karakters.
- Bereken het gemiddelde voor de getallen in de list driehoek die groter zijn dan 100.
- Druk al de unieke getallen af uit de List, driehoek.
- Druk alle unieke getallen af uit de List, driehoek, die deelbaar zijn door drie of vijf. De getallen worden in een oplopende volgorde afgedrukt.
- Druk de woorden uit de array, tekst, af in een aflopende volgorde op basis van lengte. Verder zijn woorden met dezelfde lengte gesorteerd op een oplopende alfabetische volgorde.
- Maak een groep op basis van de startletter van ieder woord voor al de woorden in de array, tekst.
- Vind alle paren van getallen en woorden waar de lengte van het woord uit de array, tekst, gelijk is aan een getal uit de List, driehoek. Druk het alle gevonden paren af.
- Maak een uitbreiding op de vorige oefening die er voor zorgt dat elk woord slechts één maal voorkomt in de resulterende paren.
- Druk alle woorden af uit de array, tekst, die minstens drie klinkers bevatten.

2. Een bedrijf heeft magazijn over heel België. Gebruik LINQ en model classes om hun queries uit te voeren. Maak twee model classes met de volgende properties:

- Warehouse:
 - BuildingName
 - WarehouseID
 - City
 - PostCode
 - Street
 - HouseNumber
 - StorageCapacity (in m²)
 - EmployeeSatisfactionRating (een lijst van scores van één tot vijf)
- Employee:
 - FirstName
 - LastName
 - ID
 - WarehouseID

Gebruik de volgende code om data in te laden in je programma en los vervolgens de gevraagde queries op:

```
List<Warehouse> warehouses = new List<Warehouse>
{
    new Warehouse("Brug4", 0, "Arendonk", 2370, "Holstraat", 14, 3000, new List<int>{ 4, 3, 1, 5 }),
    new Warehouse("Brug1", 1, "Arendonk", 2370, "Holstraat", 3, 8000, new List<int>{ 1, 4, 3, 5, 2,
3, 3, 4, 4}),
    new Warehouse("Poort1", 2, "Gent", 9000, "Stropkaai", 12, 7000, new List<int>{ 5, 4, 3, 4, 4}),
    new Warehouse("Rijstellers", 3, "Hasselt", 3500, "Industrielaan", 1, 2500, new List<int>{ 5, 4,
3, 5, 2, 3, 4, 4}),
    new Warehouse("Automa klein", 4, "Berchem", 2600, "Nieuwevaart", 77, 10000, new List<int>{ 4 }),
    new Warehouse("Schuifla", 5, "Hasselt", 3500, "Industrielaan", 15, 1500, new List<int>{ 3, 5, 2
}),
    new Warehouse("Automa groot", 6, "Berchem", 2600, "Nieuwevaart", 76, 30000, new List<int>{ 5 }),
    new Warehouse("Brug2", 7, "Arendonk", 2370, "Molenweg", 7, 3000, new List<int>{ 4, 3, 5, 2 }),
    new Warehouse("Veerhal", 8, "Melle", 9090, "Merelstraat", 48, 500, new List<int>{ 5, 5 }),
    new Warehouse("Poort2", 9, "Gent", 9000, "Burgstraat", 113, 6600, new List<int>{ 1, 2, 1, 1, 2,
3}),
    new Warehouse("D1", 10, "Knokke", 8300, "Vaart", 2, 2200, new List<int>{ 5, 4, 1 }),
    new Warehouse("Brug3", 11, "Arendonk", 2370, "Molenweg", 8, 8000, new List<int>{ 5, 2, 3, 5, 5
}),
    new Warehouse("D2", 12, "Knokke", 8300, "Vaart", 4, 2200, new List<int>{ 2, 3, 4 }),
    new Warehouse("D3", 13, "Knokke", 8300, "Vaart", 6, 2200, new List<int>{ 3, 4, 3 })
};

List<Employee> employees = new List<Employee>
{
    new Employee("Jos", "Jansen", 0, 1),
    new Employee("Ted", "Bériault", 1, 0),
    new Employee("Tony", "Hawk", 2, 3),
    new Employee("Peggy", "Corona", 3, 12),
    new Employee("Edna", "Goosen", 4, 0),
    new Employee("Mac", "Kowalski", 5, 11),
```

```

new Employee("Alejandro", "Mendoza", 6, 8),
new Employee("Aysha", "Lyon", 7, 7),
new Employee("Tyson", "Dyer", 8, 4),
new Employee("Nanou", "Hahn", 9, 6),
new Employee("Kevin", "Hahn", 10, 5),
new Employee("Kris", "Jacobsen", 11, 1),
new Employee("Boros", "Orzsebet", 12, 2),
new Employee("Buday", "Gedeon", 13, 2),
new Employee("Szölôsi", "Taksony", 14, 1),
new Employee("Kocsis", "Gyula", 15, 8),
new Employee("Asif", "Atiyeh", 16, 7),
new Employee("Ruwayd", "Akram", 17, 13),
new Employee("Makary", "Sobczak", 18, 12),
new Employee("Pawel", "Symanski", 19, 1),
new Employee("Settimio", "Calabresi", 20, 10),
new Employee("Ivo", "Bellucci", 21, 7),
new Employee("Matthieu", "Camus", 22, 9),
new Employee("Jacques", "Huard", 23, 8),
new Employee("Melville", "Bériaault", 24, 4),
new Employee("René", "Michaud", 25, 9)
};

```

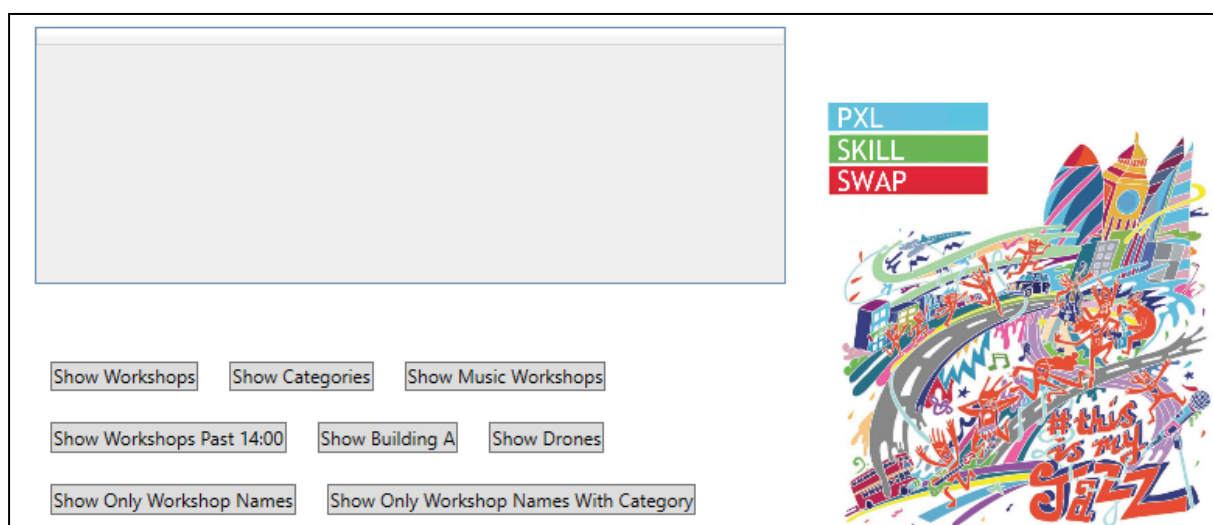
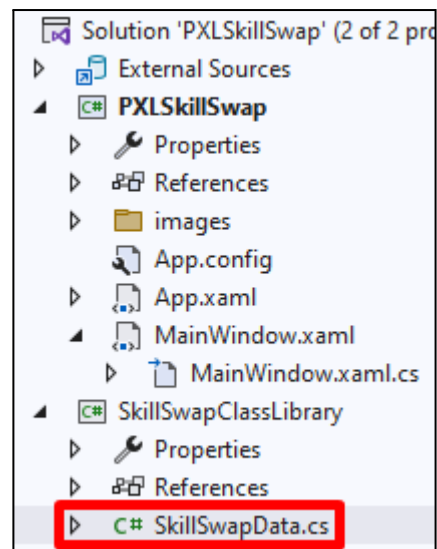
3. Los de volgende verzoeken op met behulp van LINQ
 - a. Wat zijn de namen van de magazijnen in Berchem?
 - b. Wat zijn de namen en steden van al de magazijnen geordend (van hoog naar laag) op meeste ratings van hun werknemers
 - c. Wat zijn de volledige namen en id van al de werknemers alfabetisch geordend op hun achternaam? (a tot z)
 - d. Rangschik al de magazijnen van hoog naar laag op basis van gemiddelde rating.
 - e. Wat zijn de magazijnen met postcode onder 4000 gegroepeerd per stad ?
 - f. Wat zijn de volledige namen van de werknemers die werken voor een magazijn met een opslagcapaciteit groter dan 2000? Vermeld ook de naam en capaciteit van het magazijn.
 - g. Wat zijn de id's en voornamen van werknemers die een achternaam delen met een andere werknemer van het magazijn?
 - h. Wat zijn de voornamen van de werknemers die werken voor een magazijn met een opslagcapaciteit die groter is dan 5000. Vermeld ook de volledige locatie (stad, postcode, straat, huisnummer) van het magazijn.
 - i. Groepeer de werknemers per straat van het magazijn waarvoor ze werken.

C# Adv WPF Oefeningen: PXL Skill Swap

De PXL organiseert een festival waarin de verschillende departementen workshops organiseren waaraan studenten kunnen deelnemen. In deze workshops worden de “skills” van de verschillende departementen “geswapt”.

In deze oefening gaan we een DataSet (disconnected datamodel) gebruiken om vragen te beantwoorden over het festival Skill Swap. Gebruik het skelet van het project PXL Skill Swap om de antwoorden in te laden in het DataGrid. In de SkillSwapClassLibrary vind je een klasse SkillSwapData met een static methode GetSkillSwapDataSet() die je een DataSet teruggeeft, genaamd festivalDataset.

Gebruik de methode .AsEnumerable() op de DataTable om Linq Queries te bouwen op de verzameling van rijen.



Los de volgende vragen op met LINQ en ADO.NET:

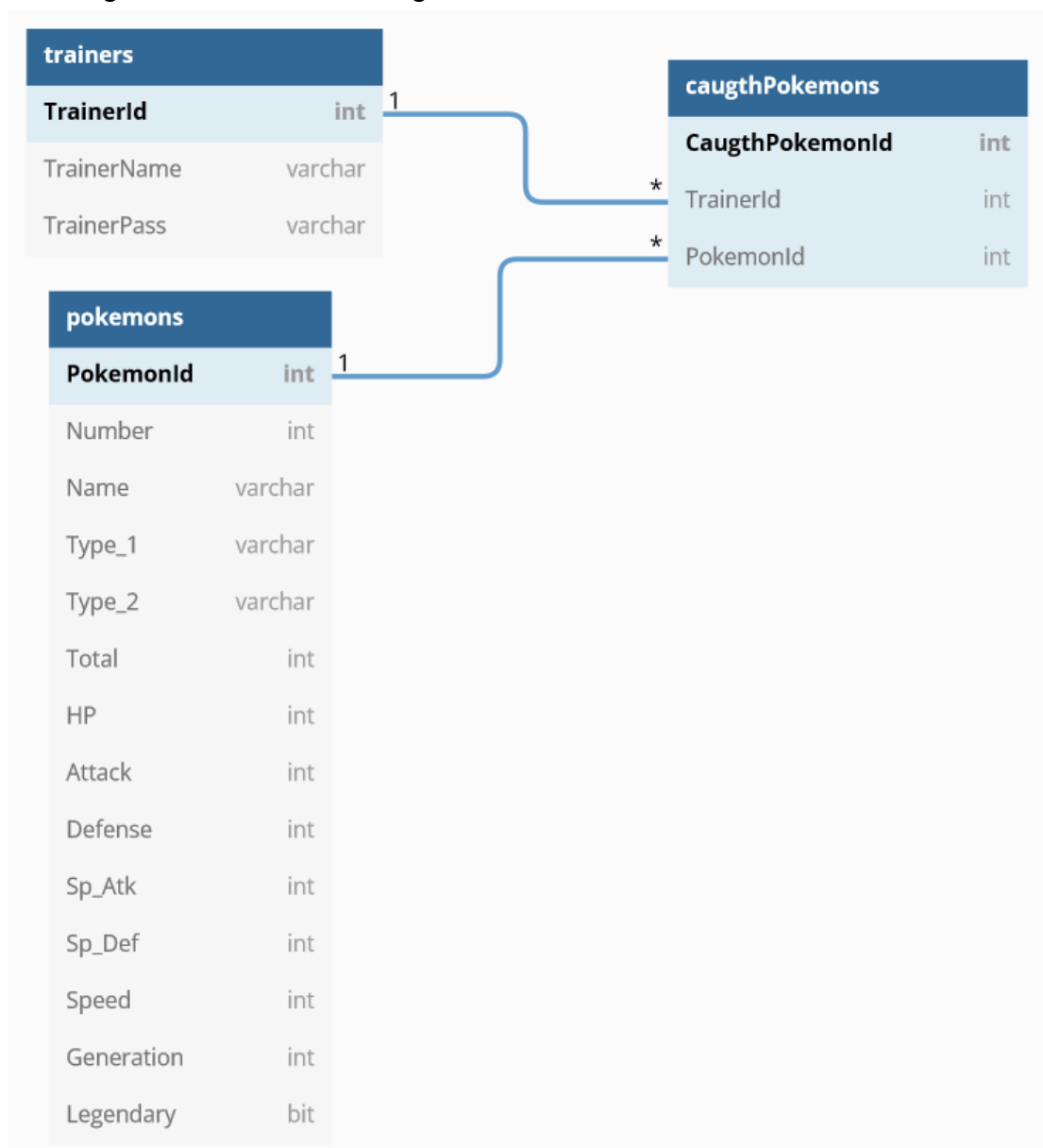
1. Toon alle workshops uit de festivalDataset in het DataGrid.
2. Toon alle categorieën uit de festivalDataset in het DataGrid.
3. Toon alle workshops van de categorie “Music”.
4. Toon alle workshops die na 14u00 worden georganiseerd.
5. Toon alle workshops die plaatsvinden in gebouw A.
6. Toon alle workshops die een sessie omtrent “drones” doen.
7. Toon alle workshops in een lijst van enkel de namen. (Geen andere kolommen mogen zichtbaar zijn).
8. Toon alle workshops in een lijst in het formaat: “workshopnaam - categorie”.

C# Adv WPF Oefeningen: LINQ-Pokemon

In deze oefening gaan we een DataSet (disconnected datamodel) bouwen om pokémons, trainers en de gevangen pokemons van trainers op te slaan. Voor een deel van de data gaan we een sql script gebruiken (*PokemonScript.sql*). Vervolgens gebruik je een SqlDataAdapter om dat deel van de DataSet in te laden.

```
-- Als tabel bestaat droppen.  
drop table if exists Pokemons;  
  
CREATE TABLE [dbo].[Pokemons](  
    [PokemonId] [int] NOT NULL,  
    [Number] [int] NOT NULL, ...
```

Het volledige model ziet er als volgt uit:



Applicatie layout:

De layout van de applicatie is al volledig voorzien. Je zal de uitgewerkte XAML terug vinden in het Skelet van de oefening.

Op het eerste venster kan de gebruiker zich inloggen. Elke trainer zal zijn eigen gevangen pokemons kunnen terugvinden.




Naam

Wachtwoord

Nieuwe Gebruiker Login

Eens ingelogd worden er willekeurige pokemons getoond elke vijf seconden. Indien de gebruiker snel genoeg de naam kan ingeven, dan zal hij deze pokemon “vangen” en terug kunnen vinden in de ListBox onderaan het venster.



PokemonId	Number	Name	Type1	Type2	Total	HP	Attac
1	1	Bulbasaur	Grass	Poison	318	45	49
2	2	Ivysaur	Grass	Poison	405	60	62
3	3	Venusaur	Grass	Poison	525	80	82
4	3	VenusaurMega Venusaur	Grass	Poison	625	80	100
5	4	Charmander	Fire		309	39	52
6	5	Charmeleon	Fire		405	58	64
7	6	Charizard	Fire	Flying	534	78	84
8	6	CharizardMega Charizard X	Fire	Dragon	634	78	130
9	6	CharizardMega Charizard Y	Fire	Flying	634	78	104
10	7	Squirtle	Water		314	44	48

Al de mogelijke interacties met de Data laag zijn al bepaald in de Business laag van de MainWindow en PokemonWindow klasse. Je moet echter wel nog de **methodes in de ClassLibrary uitwerken**. Hiervoor zal je zowel **LINQ** als **ADO.NET** moeten gebruiken.

ClassLibrary methodes:

DataBewerking:

1. De methode InitializeDataBewerking() wordt opgeroepen bij het starten van de applicatie. Deze gaat de statische constructors uitvoeren van TrainerData, PokemonData en CaughtPokemonData. Indien de DataSet nog nooit is opgeslagen, dan zal er een **Nieuwe** DataSet gemaakt moeten worden met InitializeNewDataSet().

.NET

2. De methode InitializeNewDataSet() vraagt de DataTables op van TrainerData, PokemonData en CaughtPokemonData en voegt deze toe aan de DataSet. Elke DataTable zal geïnitieerd zijn via de statische constructor.

.NET

3. De methode SaveTrainerDataSet() exporteert de DataSet naar XML formaat.

.NET

PokemonData:

1. De methode InitializePokemonDataTable() zal een connectie maken met de database om de Pokemons tabel uit te lezen. Je gebruikt hier een SqlDataAdapter voor en Fill() methode om de data door te geven. Haal vervolgens de DataTable uit de DataSet en maak er een kopie van voordat je het opslaat (.Copy()).

.NET

2. De methode GetPokemonDataView() geeft een DataView terug van de PokemonDataTable.

.NET

3. De methode GetPokemonDataViewByNameFilter() gebruikt een LINQ query om de DataTable te filteren op Name. Elke pokemon die het gegeven stuk tekst bevat in zijn/haar naam wordt getoond.

LINQ

4. De methode GetMaxPokemonNumber() gebruikt een LINQ query om het hoogste pokemon nummer te vinden in de DataTable.

LINQ

TrainerData:

1. De methode InitializeTrainerDataTable() zal een **Nieuwe** DataTable maken voor TrainerData. De DataTable bevat de kolommen die je kan terugvinden in het datamodel.



2. De methode CreateTrainer(string trainerName, string password) maakt een nieuwe DataRow aan in de TrainerDataTable. Gebruik GetNextTrainerId() om de correcte TrainerId te vinden voor de DataRow.



3. De methode GetNextTrainerId() gebruikt een LINQ query om te bepalen wat de grootste TrainerId is die momenteel gebruikt wordt. Vervolgens wordt deze waarde met één verhoogd en teruggegeven.



4. De methode CheckTrainerLogin(string trainerName, string password) gebruikt een LINQ query om te controleren of dat de gegeven trainerName en password terug te vinden zijn in de DataTable.



5. De methode CheckUniqueTrainerName(string trainerName) gebruikt een LINQ query om te controleren of dat de gegeven trainerName al bestaat.



6. De methode GetTrainerIdByTrainerName(string trainerName) gebruikt een LINQ query om de TrainerId te vinden van de unieke trainerName.



CaughtPokemonData:

1. De methode InitializeCaughtPokemonDataTable() zal een **Nieuwe** DataTable maken voor CaughtPokemonData. De DataTable bevat de kolommen die je kan terugvinden in het datamodel.



2. De methode GetNextCaughtPokemonId() gebruikt een LINQ query om te bepalen wat de grootste CaughtPokemonId is die momenteel gebruikt wordt. Vervolgens wordt deze waarde met één verhoogd en teruggegeven.



3. De methode GetCaughtPokemonIdsByTrainerId(int trainerId) gebruikt een LINQ query om een lijst van integer waardes te selecteren. De lijst bevat alle pokemon nummers die de trainer van het gegeven trainerId gevangen heeft.



4. De methode `CatchPokemon(int trainerid, int pokemonNumber)` voegt een `DataRow` toe aan `CaughtPokemonData`. Gebruik `GetNextCaughtPokemonId()` om de correcte `CaughtPokemonId` te vinden voor de `DataRow`.

