



SoundSharp Object Oriënted module

Opdracht 2- De *TrackList* Class

De volgende Class die ontwikkeld wordt is de TrackList Class. Deze Class stelt een lijst met tracks voor.

Stap 1: Nieuwe Class aanmaken

Open in Visual Studio de AudioDevices solution. Selecteer in de solution explorer het AudioDevices project en vervolgens uit het context menu (rechtermuis klik) Add | New item.... Kies in de template dialog de template Class en geef deze de naam TrackList.cs in het geval je voor C# hebt gekozen of TrackList.vb als je voor VB.NET hebt gekozen.

Verander de namespace definitie in:

C#

```
namespace AudioDevices.Tracks
```

VB.NET

```
Namespace Tracks
```

Verander de naam van de Class in: TrackList.

Stap 2: Private fields toevoegen

De TrackList Class bevat Track objecten. Voor de implementatie van de TrackList Class maken we gebruik van de .NET class CollectionBase. Deze class bevindt zich in de namespace System.Collections. Gebruik CollectionBase als de base class voor de TrackList Class.

Tip: De CollectionBase Class heeft een protected property List waar je gebruik van moet maken.

Stap 3: Indexer toevoegen

Voeg een indexer toe aan de TrackList Class die gebruik maakt van de indexer van de property List.

Stap 4: Methods toevoegen

De TrackList Class dient de volgende public methods te bevatten:

Add (...)

C#: `public void Add(Track t)`

VB.NET: `Public Sub Add(ByVal t As Track)`

De Method Add(...) accepteert een Track object als parameter en voegt deze Track vervolgens toe aan de interne lijst met tracks.

Remove (...)

C#: `public void Remove(Track t)`

VB.NET: `Public Sub Remove(ByVal t As Track)`

De Method Remove(...) accepteert een Track object als parameter en verwijdert deze Track vervolgens van de interne lijst met tracks.

```
GetShuffledList()
```

De method `Shuffle()` levert een lijst op met tracks van het tracklist object van het type:

C#: `TrackList`

VB.NET: `TrackList`

Deze lijst dient een random (willekeurig verdeelde) versie te zijn de interne lijst met tracks. Het is echter de bedoeling dat de volgorde van de tracks in de interne lijst niet veranderd wordt als men `GetShuffledList()` aanroept.

Tips:

- Maak gebruik van de .NET Class Random.
- Maak gebruik van een kopie om ervoor te zorgen dat de interne volgorde van tracks niet verandert.

Stap 5: Properties toevoegen

Voeg in de `TrackList` Class de volgende properties toe. De kolommen Get en Set geven aan of de property respectievelijk een getter en/of een setter dient te bevatten.

Naam	Type	Get	Set	Omschrijving
TotalTime*	time	✓	✗	Totale tijd van de tracklist

TotalTime*

Om de totale tijd te berekenen van alle tracks in de tracklist, kan je het beste met een loop de track objecten die de tracks variabele bevat, doorlopen:

C#: `foreach (Track track in this.List)`

M.b.v. de `Track` method `GetLength()` kan je van elke track de duur opvragen en met behulp van interne variabelen de totale tijd bijwerken.

Stap 6: TrackList Class testen

De TrackList Class is nu compleet.

Voer nu in Tester testcode uit, om te controleren of de TrackList Class goed functioneert. Bedenk zelf welke code je moet gebruiken om alle functionaliteit goed te testen.