



SoundSharp Object Oriënted module

Opdracht 1 - De *Track* Class

SoundSharp wil de software die ze ontwikkelt voor intern gebruik verder professionaliseren. Besloten is om een centrale Class Library te maken met veel gebruikte Classes. Deze Class Library kan vervolgens bij elke nieuwe applicatie gebruikt worden.

De eerste Class die ontwikkeld wordt is de Track Class. Deze Class stelt een losse MP3 track voor.

Stap 1: Nieuw project aanmaken

Maak in Visual Studio een nieuw project aan met de naam AudioDevices, kies hiervoor de Class Library template. Rename in de solution explorer Class1.cs naar Track.cs in het geval je voor C# hebt gekozen of Class1.vb naar Track.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: Track.

Stap 2: Private fields toevoegen

Voeg in de Track Class de volgende private fields toe:

Naam	Type	Omschrijving
id	integer	Unieke numerieke id voor de track.
name	string	Naam van de track.
artist	string	Naam van de uitvoerende artiest.
albumSource	string	Naam van het album waar de track van afkomstig is.
style	category*	Muziek style categorie.
length	time**	Lengte van de track.

Category type*

Definieer een enumerator met de naam Category om een lijst van muziekstijlen op te slaan. De enumerator moet de volgende waarden bevatten:

Ambient
Blues
Country
Disco
Electro
Hardcore
HardRock
HeavyMetal
Hiphop
Jazz
Jumpstyle
Klassiek
Latin
Other
Pop
Punk
Reggae
Rock
Soul
Trance
Techno

Time type**

Om de lengte van een track op te slaan maak je gebruik van een structure Time. Maak deze structure met de volgende specificaties:

- De structure moet drie integer variabelen bevatten: hours, minutes en seconds.
- De structure moet de volgende drie constructors bevatten:

C#:

```
public Time(int seconds)

public Time(int minutes, int seconds)

public Time(int hours, int minutes, int seconds)
```

VB.NET:

```
Public Sub New(ByVal seconds As Integer)
Public Sub New(ByVal minutes As Integer, ByVal seconds As Integer)
Public Sub New(ByVal hours As Integer, ByVal minutes As Integer,
ByVal seconds As Integer)
```

Zorg ervoor dat de constructor de juiste waarden toekent aan de interne variabelen.

Voorbeeld:

C#

```
Time t = new Time(100)
```

VB.NET

```
Dim t As Time = New Time(100)
```

Nu dient de structure t de volgende waarden te bevatten:

hours:	0
minutes:	1
seconds:	40

Override de ToString() method van de Time structure zodat deze een string terug geeft met de volgende format: <hours> : <minutes> : <seconds>.

Stap 3: Constructors toevoegen

Voeg aan de Track Class de volgende constructors toe:

C#

```
public Track()  
public Track(int id)  
public Track(int id, string name)  
public Track(int id, string artist, string name)
```

VB.NET

```
Public Sub New()  
Public Sub New(ByVal id As Integer)  
Public Sub New(ByVal id As Integer, ByVal name As String)  
Public Sub New(ByVal id As Integer, ByVal artist As String, ByVal name As String)
```

Zorg ervoor dat binnen de constructors de opgegeven waarden in de constructor parameters gebruikt worden om de private fields te initialiseren.

Stap 4: Methods toevoegen

De Track Class dient de volgende twee public methods te bevatten:

GetLength()

De method GetLength() geeft de lengte van de Track terug in de vorm van een Time structure.

GetLengthInSeconds()

De method GetLengthInSeconds() geeft de lengte van de Track in secondes terug in de vorm van een integer variabele. In deze method vindt dus een conversie plaats van de tijd, geregistreerd in de private field length van het type structure Time, naar een integer.

Stap 5: Properties toevoegen

Voeg in de Track 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
Id	integer	✓	✓	Unieke numerieke id voor de track.
Name	string	✓	✓	Naam van de track.
Artist	string	✓	✓	Naam van de uitvoerende artiest.
DisplayName*	string	✓	✗	Naam van de uitvoerende artiest en naam van de track.
Length	time	✗	✓	Lengte van de track.
DisplayLength	string	✓	✗	Lengte van de track als een string.
Style	category	✓	✓	Muziek style categorie.
AlbumSource	string	✓	✓	Naam van het album waar de track van afkomstig is.

DisplayName*

DisplayName toont alleen de naam van de uitvoerende artiest en track als deze beiden een waarde hebben. Als dat niet het geval is dient de tekst: "unknown" getoond te worden.

Stap 6: Track Class testen

De Track Class is nu compleet. Om deze te testen maken we een nieuwe Console Application binnen dezelfde Visual Studio 2005 solution.

Kies File | New Project... en maak een nieuwe Console Application aan met de naam Tester. Zorg ervoor dat dit project bij de bestaande solution wordt toegevoegd (Solution: Add to Solution). Maak nu binnen het Tester project als volgt een referentie naar de AudioDevices Class Library:

- Selecteer References en vervolgens uit het context menu (rechtermuis klik) Add Reference.
- Ga naar de Projects tab.
- Selecteer project AudioDevices en vervolgens [ok].

Verander de namespace in Tester en voeg de AudioDevices.Tracks namespace toe.

C#

```
using AudioDevices.Tracks;
```

VB.NET

Imports Tracks

Voeg nu de volgende testcode toe aan Tester en controleer of de Track Class goed functioneert:

C#

```
Track t1 = new Track(1, "Nelly Furtado", "Maneater");
t1.AlbumSource = "Loose";
t1.Length = new Time(4, 31);
t1.Style = Category.Pop;

Console.WriteLine(t1.DisplayName);
Console.WriteLine(t1.DisplayLength);
Console.WriteLine("Tijd in seconden: {0}", t1.GetLengthInSeconds());
Console.WriteLine("Category: {0}", t1.Style.ToString());
Console.ReadLine();
```