

CODING STANDARDS

Probeer je tijdens dit project een beetje aan deze regeltjes te houden zodat we allemaal een vergelijkbare programmeerstijl aanhouden en de code netjes documenteren. Het is prima als je de eerste dagen van een week code hebt die niet helemaal voldoet hieraan, maar zorg dat vrijdagavond toch al je code in het goede format staat. Zo niet, dan nodig ik je uit om “100 Ways To Love A Cat” te kijken als marteling.

Afspraken

1. Boven elke klasse staat kort beschreven wat de klasse voorstelt en in welke context deze gezien moet worden.
2. Elke methode heeft een gecommmentarieerde header waar de functionaliteit kort beschreven staat. Als er bepaalde specifieke implementatiedetails zijn die belangrijk zijn om te weten, zet deze er dan ook bij.
3. Ook instance variabelen mogen best een gecommmentarieerde header krijgen. Niet noodzakelijk, maar wel zo netjes.
4. Als je ergens een stukje code hebt waarvan je denkt dat anderen dit niet binnen een paar seconden kunnen begrijpen, zet er dan in commentaar bij wat voor iets raars je code zit uit te spoken daar.
5. We gebruiken camelCase voor de naamgeving van variabelen, klassen en methodes. Variabelen beginnen altijd met een kleine letter, de rest met een hoofdletter. Static variabelen en methodes hebben nog een `_` vóór de eerste letter.

Voorbeeldcode

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Models
{
    /// <summary>
    /// Een track is een serie van segmenten tussen een begin-
    /// en eindstation. Aan deze track worden treinen gekoppeld.
    /// </summary>
    public class Track
    {
        /// <summary>
        /// Unieke code voor deze track.
        /// </summary>
        private string trackCode;

        /// <summary>
        /// Serie van segmenten die samen deze track vormen.
        /// </summary>
        private List<TrackSegment> segments;

        /// <summary>
        /// Retourneert startstation van deze track.
        /// </summary>
    }
}
```

```

public Station StartStation
{
    get
    {
        return this.segments.First().GetStartStation();
    }
}

/// <summary>
/// Retourneert eindstation van deze track.
/// </summary>
public Station EndStation
{
    get
    {
        return this.segments.Last().GetEndStation();
    }
}

/// <summary>
/// Construeert track met meegegeven parameter als
/// unieke code.
/// </summary>
/// <param name="code">Code voor track</param>
public Track(string code)
{
    this.trackCode = code;
}

/// <summary>
/// Construeert track met meegegeven code en segmenten.
/// </summary>
/// <param name="code">Code voor track</param>
/// <param name="segments">Lijst van segmenten</param>
public Track(string code, List<TrackSegment> segments)
{
    this.trackCode = code;
    this.segments = segments;
}

/// <summary>
/// Voegt meegegeven segment toe aan deze track.
/// </summary>
/// <param name="segment">Tracksegment om track mee uit
/// te breiden.</param>
public void AddSegment(TrackSegment segment)
{
    this.segments.Add(segment);
}
}
}

```