

Opdracht 1 ('MVC')

Je kent ze vast wel, die displays in de trein waarop het volgende station wordt getoond, met de verwachte aankomsttijd en spoor; in elk treinstel hangen er wel een paar. We gaan dit nabouwen met een (Windows Forms) applicatie waarin deze schermen (Views) zijn opgenomen, tezamen met de benodigde datastructuur (Model) en een controlpanel (gebruikmakend van een Controller) waarmee de machinist kan aangeven dat het volgende station bereikt is. In het echt gebeurt dat uiteraard automatisch middels sensoren in de trein en op de stations, maar om het te kunnen testen zonder hardware, gebruiken we dus een simpele controlpanel met een aantal knoppen (om aan te geven dat het volgende station is bereikt, en om aan te geven dat de terugreis begint).

De applicatie draait in een trein die continue op en neer gaat, bijvoorbeeld van Den Helder naar Nijmegen (zie www.ns.nl op welke stations er allemaal gestopt wordt tijdens een reis).

Model

Maak gebruik van de volgende classes voor het model:

```
public class TreinStation
{
    public string Naam { get; set; }
    public string AankomstSpoor { get; set; }
    public DateTime AankomstTijd { get; set; }
    public DateTime VertrekTijd { get; set; }

    public TreinStation(string naam, string aankomstSpoor, DateTime aankomstTijd, DateTime vertrekTijd)
    {
        Naam = naam;
        AankomstSpoor = aankomstSpoor;
        AankomstTijd = aankomstTijd;
        VertrekTijd = vertrekTijd;
    }
}

public class TreinReis : ITreinReis
{
    private List<TreinStation> stations;
    private int huidigStation;

    public TreinReis()
    {
        stations = new List<TreinStation>();
        // todo: voeg (hardcoded) Stations toe...
        huidigStation = 0;
    }

    // ...
}
```

Bedenk zelf wat voor properties/methoden er nog meer nodig zijn in class TreinReis (die interface 'ITreinReis' implementeert). Denk in ieder geval aan de Observer methoden (AddObserver, RemoveObserver), waarmee de Views op de hoogte gebracht kunnen worden van wijzigingen.

Controller

Maak voor de Controller een interface 'ITreinController' en een class 'TreinController' die deze interface implementeert. Geef de controller een instantie mee van ITreinReis (de interface!) zodat de akties/methoden via de controller vertaald kunnen worden naar akties/methoden van de TreinReis.

Je kunt de (standaard) main form v/d Windows Form applicatie gebruiken voor de weergave v/d controlpanel. Zoals gezegd kan de machinist via deze controlpanel aangeven dat het volgende station bereikt is. Verder kan de machinist aangeven dat de terugreis begint, oftewel dat het eindstation is bereikt en de reis 'achterste-voren' vervolgt (de lijst met stations zal dus 'omgedraaid' moeten worden, of 'achterste-voren' verwerkt moeten worden). Let erop dat deze controlpanel (main form) alleen met de Controller communiceert, niet met het model (treinreis)!

Views

Maak een apart Windows Form voor de View; je kunt hiermee dan eenvoudig meerdere schermen aanmaken (die hetzelfde tonen) via 'ITreinDisplay display1 = new TreinDisplay()'. Wat getoond moet worden is informatie over het volgende station: de naam van het station, de verwachte aankomsttijd en het spoor waar de trein zal aankomen. De informatie op het scherm wordt alleen geupdate als het model (de treinreis) wijzigt, en de betreffende update-methode van de View wordt aangeroepen. Let er op dat alleen de Controller (en niet de View) het model manipuleert!

Maak ook eventueel een andere (2^{de}) display waarin naast genoemde informatie ook een soort lijst getoond wordt met alle stations die nog komen t/m het eindstation. Kijk maar eens in de trein hoe deze informatie wordt weergegeven.

