



# Challenge Traffic Light (met Enum)

FHICT

We gaan (wederom) verkeerslichten programmeren. Als je de TrafficLight Challenge uit een vorig hoofdstuk al hebt gedaan, zeker als je de voorbeeldcode had overgenomen (met de fouten er in) zul je gemerkt hebben dat het gebruik van een *String* voor de toestand (kleur) van een TrafficLight snel leidt tot fouten! Zo zijn *Orange*, *Oranje* en *orange* allemaal door de compiler geaccepteerd worden maar de waarden zijn verschillend: hierdoor kunnen allerlei bugs ontstaan:

Neem bijvoorbeeld 2 TrafficLights, zeg *trafficLight1* en *trafficLight2*. Als de *color* van *trafficLight1* waarde *Oranje* heeft en de *color* van *trafficLight2* waarde *Orange* dan zal een vergelijking als

```
[ if (trafficLight1.color == trafficLight2.color) {  
    ...  
}]
```

*false* opleveren terwijl de programmeur *true* verwacht: een tikfout die grote gevolgen kan hebben voor het gedrag van de software.

Om die bugs te voorkomen gaan we er ditmaal een Enum bij gebruiken: Door het gebruik van een *Enum* kan de compiler helpen fouten te ontdekken en voorkomen!

Voor de *color* (toestand) van het TrafficLight maken we een Enum aan:

```
[ public enum TrafficlightColors {  
    Red,  
    Orange,  
    Green  
:]
```

```
[
}
```

## Analyse

- De code is in het Engels.
- Ik wil de mogelijkheid hebben meerdere objecten van type *TrafficLight* te maken, maar ik wil het maar 1 keer programmeren.
- Een *TrafficLight* kan de kleuren (*toestanden*) 'Green', 'Orange', 'Red' hebben: gebruik voor deze toestanden Engelse benamingen.
- Voor de veiligheid wordt de toestand bij het maken van een *TrafficLight* altijd op *rood* gezet.
- Vanuit *Red* kan de toestand alleen *Green* worden, dan *Orange*, dan weer *Red*.

## Ontwerp

- Een class *TrafficLight*.
- Deze class heeft een private Field *color* van type *TrafficLightColors*.
- Initieële waarde van *color*: *TrafficLightColors.Red*.
- De class krijgt een method *NextState()* die het *TrafficLight* de volgende waarde van *color* geeft.
- We willen aan een *TrafficLight* ook kunnen opvragen wat de huidige kleur is: `public TrafficLightColors GetCurrentColor()`

## Realisatie

- Maak de in het ontwerp genoemde zaken aan.
- De method *NextState* geeft de kleur **na** het veranderen van de kleur terug: `public TrafficLightColors NextState() {...}`

Een Console app heeft een *main*-method (`public static void Main(string[] args)`) waarin je code kunt zetten als:

```
TrafficLight trafficLight = new TrafficLight();
// color has to be TrafficLightColors.Red.
Console.WriteLine(trafficLight.GetCurrentcolor());
trafficLight.NextState();
// color has to be TrafficLightColors.Green.
Console.WriteLine(trafficLight.GetCurrentColor());
trafficLight.NextState();
// color has to be TrafficLightColors.Orange.
Console.WriteLine(trafficLight.GetCurentColor());
```

```
trafficLight.NextState();  
// and TrafficlightColors.Red again!  
Console.WriteLine(trafficLight.GetCurentColor());
```

Is deze code slordig in elkaar gezet? Kijk goed of er geen fouten in staan!  
Verbeter ze zonodig en test het programma uit.

Kun je verbeteringen op het programma bedenken?