# Oefeningen Xml Technologie Les 3

1. Schrijf de 2 utility methods die in de slides uitgelegd worden om namespaces te maken. Zet die methods in een aparte class met naam *utilities*, ergens op een goede plaats in je project. Je kan die dan aanroepen in je oefeningen.

Zet de *Save* method van vorige week ook in deze class. Maak er allemaal **static** methods van.

**public static Namespace getDefaultNamespace();**

**public static Namespace getXNamespace();**

Als je de extra oefening van de vorige reeks wil afwerken, heb je deze methods nodig. Gebruik deze methods ook in de volgende oefeningen, anders heb je geen werkende Xaml.

Oefening 2 tot 4 hangen samen. Bekijk eerst even de file *gridvoorbeeld.xaml*, dan weet je welke richting je uit moet.

1. Schrijf de volgende utility method waarmee je een background kan geven aan een Xaml control zoals Canvas of Grid. Werk met een gradient brush.

**private** Element makeGradientBackground(

String control,

String startColor,

String endColor)

{

/\* Voorbeeld van wat het gereturnde element kan bvb zijn

<Grid.Background>

<LinearGradientBrush>

<GradientStop Color="#FF93C5E8"/>

<GradientStop Color="#FF3B596E" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

\*/

}

Zoals je ziet, geef je 3 argumenten mee :

De control waarvoor je de achtergrond wil maken (Grid of Canvas of …)

De start kleur

De eindkleur

Je ziet ook dat de method een Element returnt. Dit is het xml element waarin de background wordt gedefiniëerd, en dat je dan aan de control in kwestie kan hangen. Een voorbeeld van wat er in dat element moet zitten, staat er bij in commentaar.

1. Schrijf 2 methods :

**private** Element getGridRows(**int** number)

**private** Element getGridColumns(**int** number)

Deze methods returnen een element met een aantal row- en column definitions.

1. Schrijf nu een method die een grid maakt van 3 rijen en 3 kolommen, met een gradient achtergrond die loopt van Ivory naar Red. Roep hiervoor de methods van de vorige oefeningen aan.

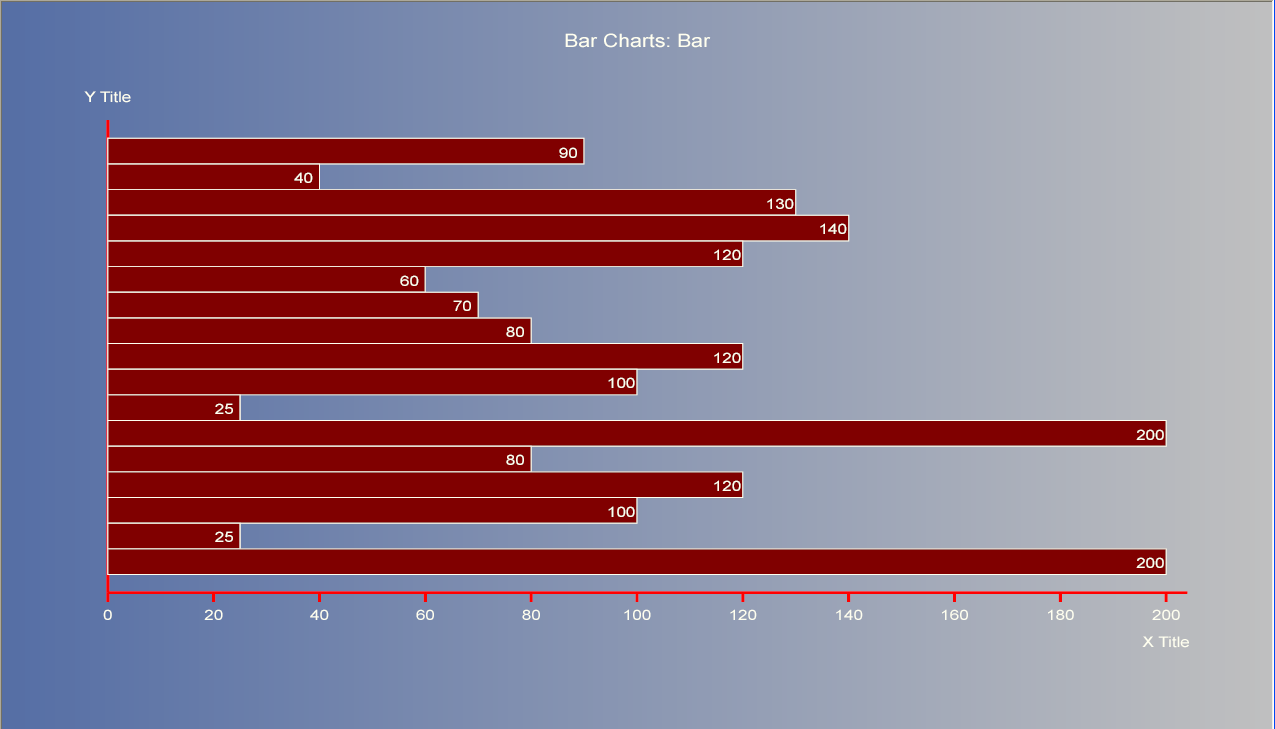
Zet in elke cel de indexen van de cel, bvb *[0,1]*. Gebruik hiervoor het Xaml element TextBlock (surf daar eens rond, is volledig vanzelfsprekend). Controlleer je versies in KaXaml.

1. Je vindt gegevens over een user interface in het bestand *users.xml*, waaruit je dus zal moeten lezen. Vraag vanuit Java een naam op, en genereer de overeenkomende xaml (een page met een canvas).

**private** **void** makeUserInterface(String name) {}

Als de ingegeven string bvb “Dirk Cassiers” is, dan maak je een page met een Canvas waarvan de achtergrond een gradient background is die loopt van de in de xml file opgegeven startkleur naar de eindkleur. Test met de 3 mogelijke strings, en voorzie ook een default Canvas voor een user die niet in de xml file zit. Hiervan moet de achtergrond een SolidColorBrush zijn op de kleur DarkGray.

1. Synthese oefening. Lees de data uit de file BarchartValues 1.xml en maak er de volgende barchart van.



Hier volgen wat tips :

* Gebruik een Canvas met een gradient achtergrond die van 556EA5 tot C0C0C0 loopt. Je hebt hier al een method voor geschreven.
* Voor de titels en de waarden gebruik je TextBlocks. Haal de teksten zelf uit de xml file. Als je dan een andere xml file gebruikt, past de inhoud zich aan.
* Schrijf een method drawLine die een lijn (Xaml element : Line) tekent van een bepaalde kleur en dikte op een bepaalde plaats. Roep deze method aan om de X en de Y assen te tekenen. Voor de X-as mag je er van uitgaan dat 200 de maximale waarde is (om de onderverdelingen op de as uit te kunnen rekenen). Voor de Y-as kan je werken met het aantal barelementen.
* Schrijf een method drawRectangle die een Border met kleur maroon tekent, en er de waarde rechts bijzet in een TextBlock.

Dan heb je alle methods die je kan aanroepen om de chart te maken.

Extra : zorg er voor dat de chart volledig met de 2 xml bestanden werkt. Je moet dus eerst uitrekenen hoeveel pixels de hoogte van een rectangle mag zijn om ze er allemaal op te krijgen. Zoek ook wat de grootste waarde is, en gebruik die waarde om de X-as te tekenen en onder te verdelen. Allemaal niet zo simpel.