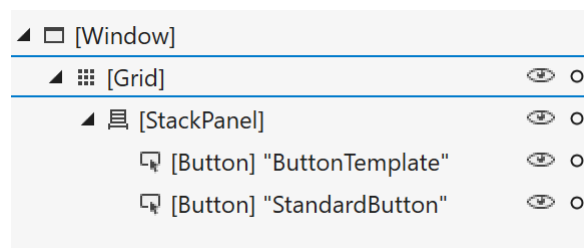


# Windowsprogrammierung – WiSe17

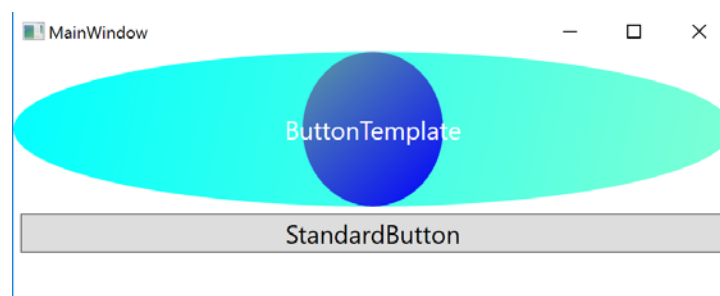
## Übung 9

### Aufgabe 9.1 Trigger und Control Templates

Schreiben Sie ein WPF-Programm, dessen Dokumentengliederung folgendermaßen aussieht:



Die Schaltfläche mit der Aufschrift „ButtonTemplate“ hat ein eigenes `ControlTemplate`, der andere Button (Aufschrift „StandardButton“) ist eine Standard-Schaltfläche der WPF. Wenn das Programm gestartet wird, sieht es folgendermaßen aus:



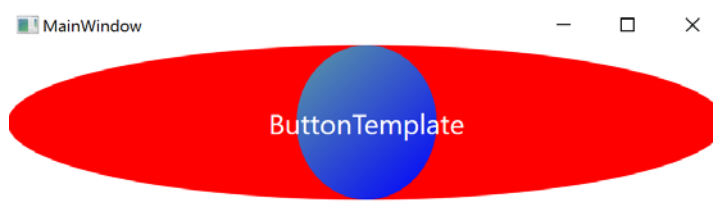
Das `ControlTemplate` des Buttons ist folgendermaßen aufgebaut:

Es besteht aus einem `Grid` mit einer Zeile und einer Spalte. Darin sind zwei Ellipsen unterschiedlicher Größe enthalten, deren verschiedene Füllfarben (XAML-Attribut `Fill`) Sie nach Ihrem Geschmack wählen können. Als Randbedingung sollten Sie hierbei mit einem Element des Typs `LinearGradientBrush` arbeiten. Die Beschriftung des Buttons (hier „ButtonTemplate“) sollte in jedem Buttonzustand zu sehen sein.

Die Schriftgröße für beide Schaltflächen ist 20.

Um das Aussehen des Buttons etwas zu animieren, weisen Sie der Triggers-Property des ControlTemplates zwei Property-Trigger zu:

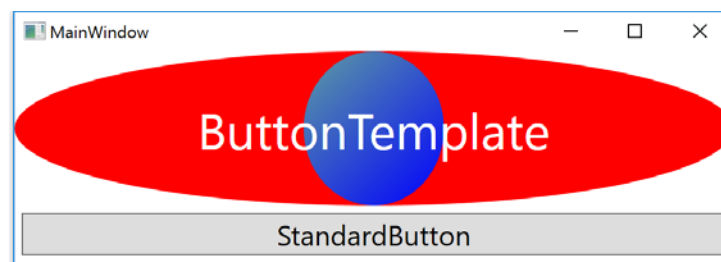
- a) Wenn die Dependency-Property `IsMouseOver` der Button-Klasse den Wert `true` angenommen hat, soll die größere Ellipse in eine Farbe Ihrer Wahl wechseln. Der Button sieht dann beispielsweise folgendermaßen aus:



- b) Wenn die Dependency-Property `IsPressed` der Button-Klasse den Wert `true` angenommen hat, soll der Button auf 90% seiner Größe sowohl in X- als auch in Y-Richtung verkleinert werden.

Außerdem soll auf **beide Buttons** noch eine Animation der folgenden Art wirken, die über `EventTrigger` realisiert wird:

- c) Wenn das `RoutedEvent MouseEnter` auftritt, dann soll die Schriftgröße der beiden Buttons auf 36 anwachsen.



- d) Wenn das `RoutedEvent MouseLeave` auftritt, dann soll die Schriftgröße der beiden Buttons wieder auf 20 zurückgehen.