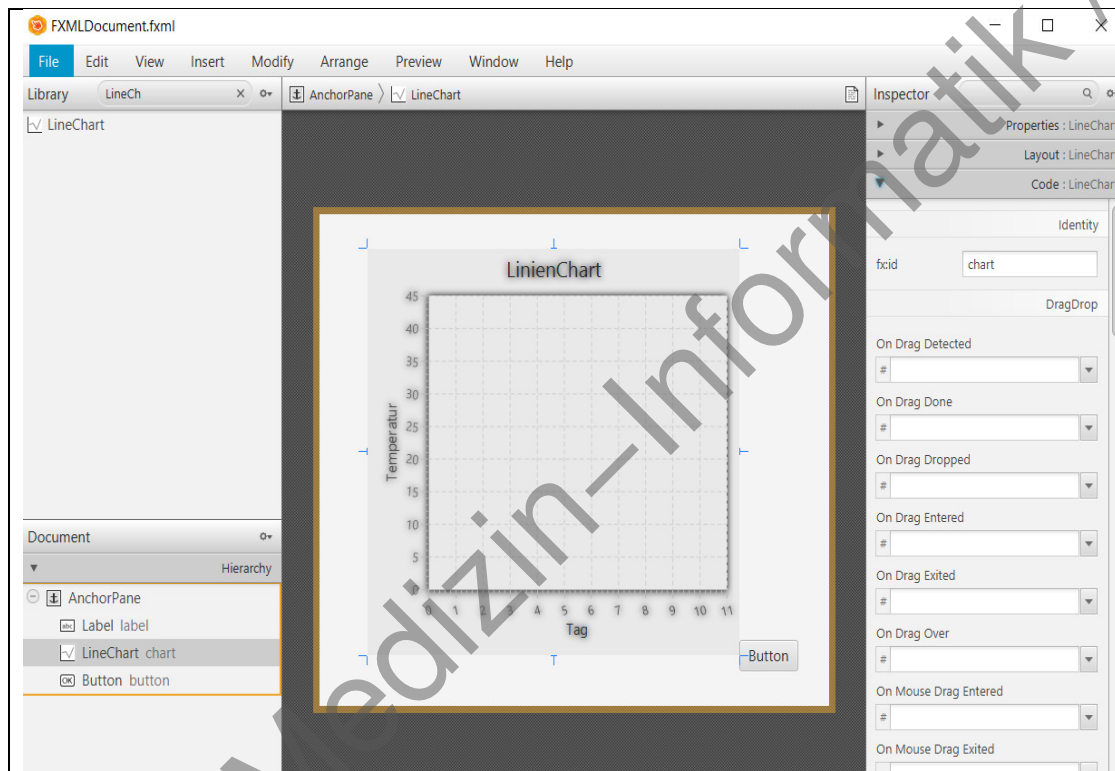


Charts erstellen

Ein Diagramm anzeigen

Alle Charts leiten von der Basisklasse Chart ab, die vor allem für das Zeichnen von Hintergrund, Legende und Titel verantwortlich ist. XYCharts erweitern diese Klasse, um die Fähigkeit, zwei Achsen darzustellen. Mit Ausnahme des PieChart, das keine Achsen benötigt und nur eindimensionale Daten darstellt, leiten alle anderen Klassen davon ab.

LineChart

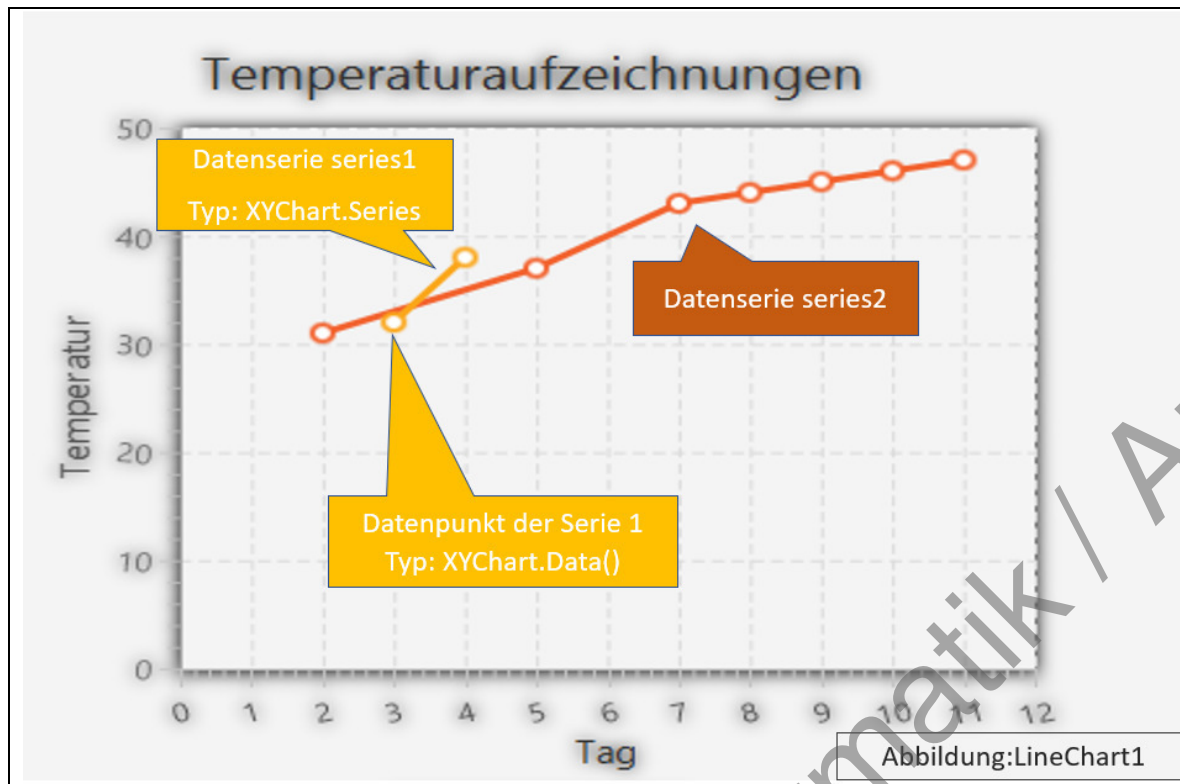


Mit dem SceneBuilder wird ein LineChart positioniert. Das Chart erhält eine id.

Beispiel
VorzeigenLineChart
(FXMLDocument.fxml)

<pre> <xAxis> <NumberAxis label="Tag" lowerBound="1.0" prefHeight= </xAxis> <yAxis> <NumberAxis label="Temperatur" lowerBound="30.0" < <tickLabelFont> </tickLabelFont></NumberAxis> </yAxis> </pre>	<p>FXML-Darstellung der Graphik. Ich wähle für dieses Beispiel 2 numerische Achsen. Es kann auch eine CategoryAxis gewählt werden</p>	<p>Beispiel VorzeigenLineChart (FXMLDocument.fxml)</p>
<ul style="list-style-type: none"> • Für die XYCharts werden eine x- und eine y-Achse definiert. • Wir werden dabei vor allem mit der NumberAxis zu tun haben, deren Daten als Zahlen definiert sind • Bei der CategoryAxis, erfolgt die Wertzuordnung anhand von Strings. 		

Das LineChart im Detail



Ein LineChart besteht aus einer oder mehreren Serien. Die Serie besteht aus mehreren Datenpunkten.

Beispiel
VorzeigenLineChart
(FXMLDocument.fxml)

```

public void initialize(URL url, ResourceBundle rb) {
    chart.setTitle("Temperaturaufzeichnungen ");
    //defining a series
    XYChart.Series series1 = new XYChart.Series();
    series1.setName("Tag 1");
    //populating the series with data
    series1.getData().add(new XYChart.Data(2, 31));
    series1.getData().add(new XYChart.Data(5, 37));

    XYChart.Series series2 = new XYChart.Series();
    series2.setName("Tag 2");
    //populating the series with data
    series2.getData().add(new XYChart.Data(3, 32));
    series2.getData().add(new XYChart.Data(4, 38));

    ObservableList<XYChart.Series<Integer, Integer>> 1
    =FXCollections.observableArrayList();
    1.add(series1);
    1.add(series2);
    chart.setData(1);
}

```

Titel des Diagrammes

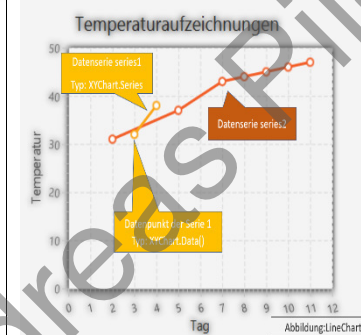
Erstellung der Serie

Erstellung der Serie

Serien müssen in
einer ObservableList
zur Verfügung
gestellt werden

Setzen der Daten im
Diagramm

Abbildung:LineChart2



Beispiel
VorzeigenLineChart
(FXMLDocument.fxml
)

@FXML

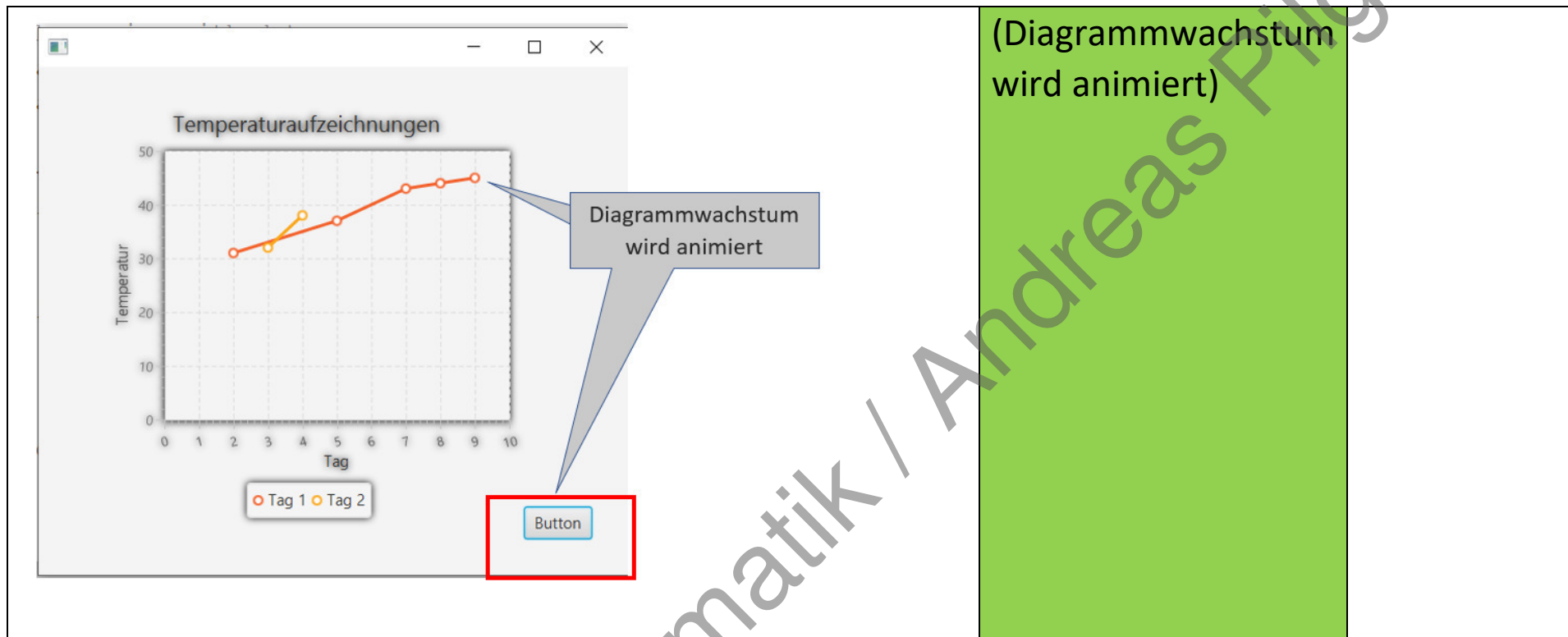
```

private void buttonClick(ActionEvent event) {
    ((XYChart.Series) chart.getData().get(0)).getData()
    .add(new XYChart.Data(6+x, 42+x));
    x++;
}

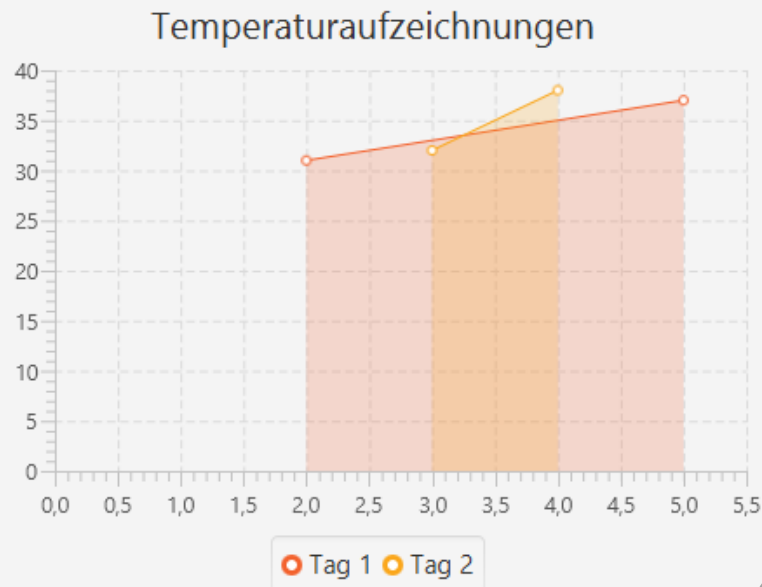
```

Beschaffung der
Serie1 und
Hinzufügen eines
neuen Datenpunktes

Beispiel
VorzeigenLineChart
(FXMLDocument.fxml)

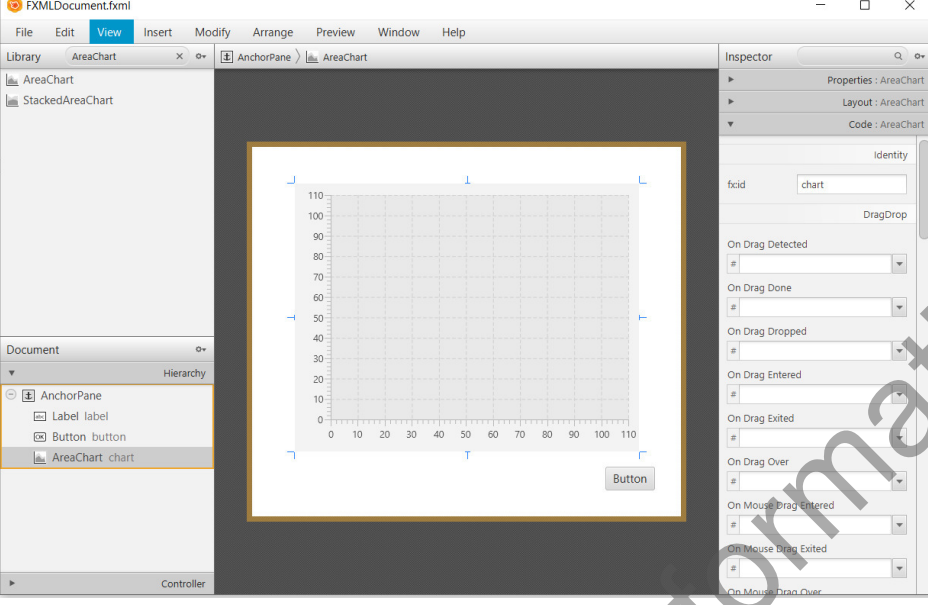


AreaChart		
-----------	--	--

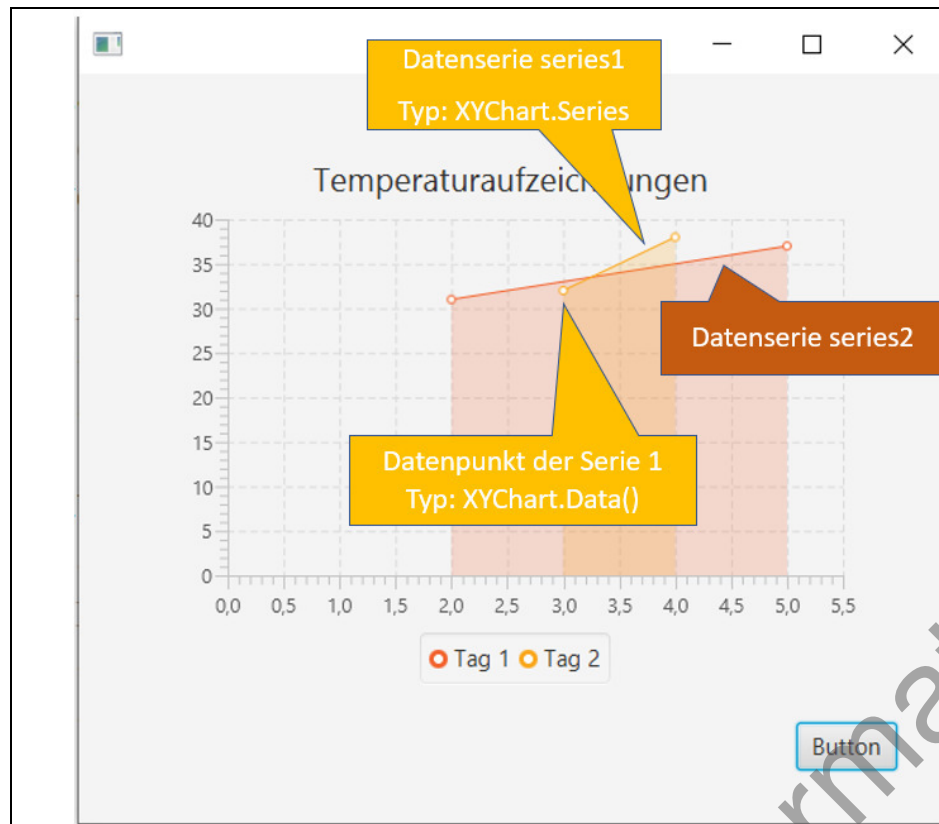


Button

- Das AreaChart unterscheidet sich von dem LineChart dadurch, dass die Flächen zwischen x-Achse und den Verbindungslinien zwischen den Datenpunkten einer Zeitreihe farbig ausgefüllt sind.
- Man unterscheidet zwischen gestapelten und überlappenden Flächendiagrammen. Das überlappende Flächendiagramm dient demselben Zweck wie das Liniendiagramm und macht es einfacher, Trends zu verfolgen, da

	<p>zusammengehörige Datenpunkte durch eine gleichmäßig eingefärbte Fläche dargestellt werden.</p>	
	<p>Mit dem SceneBuilder wird ein AreaChart positioniert. Das Chart erhält eine id.</p>	<p>Beispiel VorzeigenAreaChart (FXMLDocument.fxml)</p>

<pre> <AreaChart fx:id="chart" layoutX="56.0" layoutY="48.0" <xAxis> <NumberAxis side="BOTTOM" /> </xAxis> <yAxis> <NumberAxis side="LEFT" /> </yAxis> </AreaChart> </pre>	<p>FXML-Darstellung der Graphik. Ich wähle für dieses Beispiel 2 numerische Achsen. Es kann auch eine CategoryAxis gewählt werden</p>	<p>Beispiel VorzeigenAreaChart (FXMLDocument.fxml l)</p>
<p>Für die XYCharts werden eine x- und eine y-Achse definiert.</p> <ul style="list-style-type: none"> • Wir werden dabei vor allem mit der NumberAxis zu tun haben, deren Daten als Zahlen definiert sind • Bei der CategoryAxis, erfolgt die Wertzuordnung anhand von Strings. 		<p>Beispiel VorzeigenAreaChart (FXMLDocument.fxml l)</p>



Ein AreaChart besteht aus einer oder mehreren Serien. Die Serie besteht aus mehreren Datenpunkten

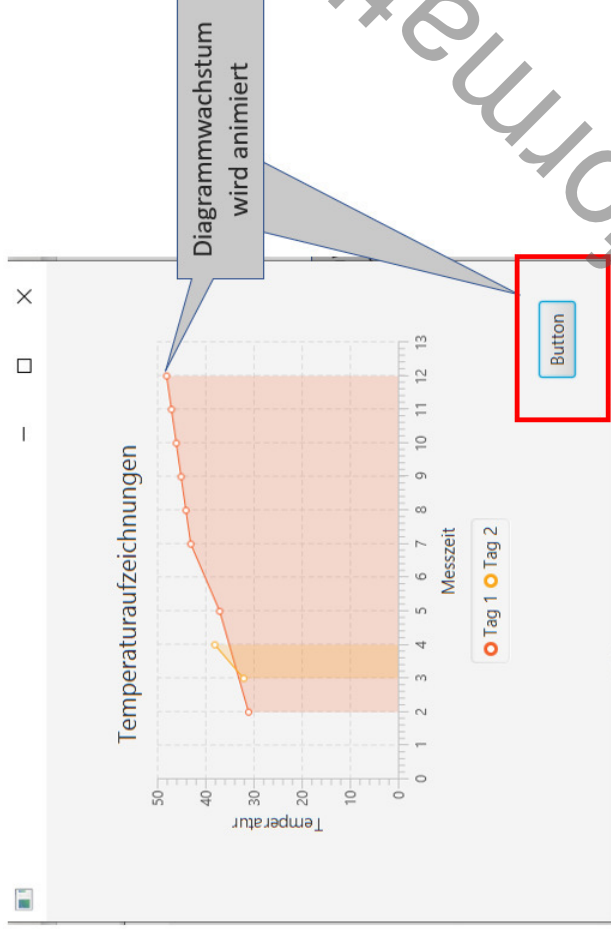
Beispiel
VorzeigenAreaChart
(FXMLDocument.fxm
l)

<pre> public void initialize(URL url, ResourceBundle rb) { // TODO chart.setTitle("Temperaturaufzeichnungen "); //defining a series XYChart.Series series1 = new XYChart.Series(); series1.setName("Tag 1"); //populating the series with data series1.getData().add(new XYChart.Data(2, 31)); series1.getData().add(new XYChart.Data(5, 37)); XYChart.Series series2 = new XYChart.Series(); series2.setName("Tag 2"); //populating the series with data series2.getData().add(new XYChart.Data(3, 32)); series2.getData().add(new XYChart.Data(4, 38)); ObservableList<XYChart.Series<Integer, Integer>> FXCollections.observableArrayList(); l.add(series1); l.add(series2); chart.setData(l); </pre> <div data-bbox="891 240 1093 272">Titel des Diagrammes</div> <div data-bbox="904 328 1079 360">Erstellung der Serie</div> <div data-bbox="904 416 1079 448">Erstellung der Serie</div> <div data-bbox="904 655 1079 767">Serien müssen in einer ObservableList zur Verfügung gestellt werden</div> <div data-bbox="607 735 801 791">Setzen der Daten im Diagramm</div> <div data-bbox="837 791 1032 815">Abbildung:AreaChart2</div>		<p>Beispiel VorzeigenAreaChart (FXMLDocument.fxml l)</p>
	<p>Beschaffung der Serie1 und Hinzufügen eines neuen Datenpunktes(Diagrammwachstum wird animiert)</p>	<p>Beispiel VorzeigenAreaChart (FXMLDocument.fxml l)</p>

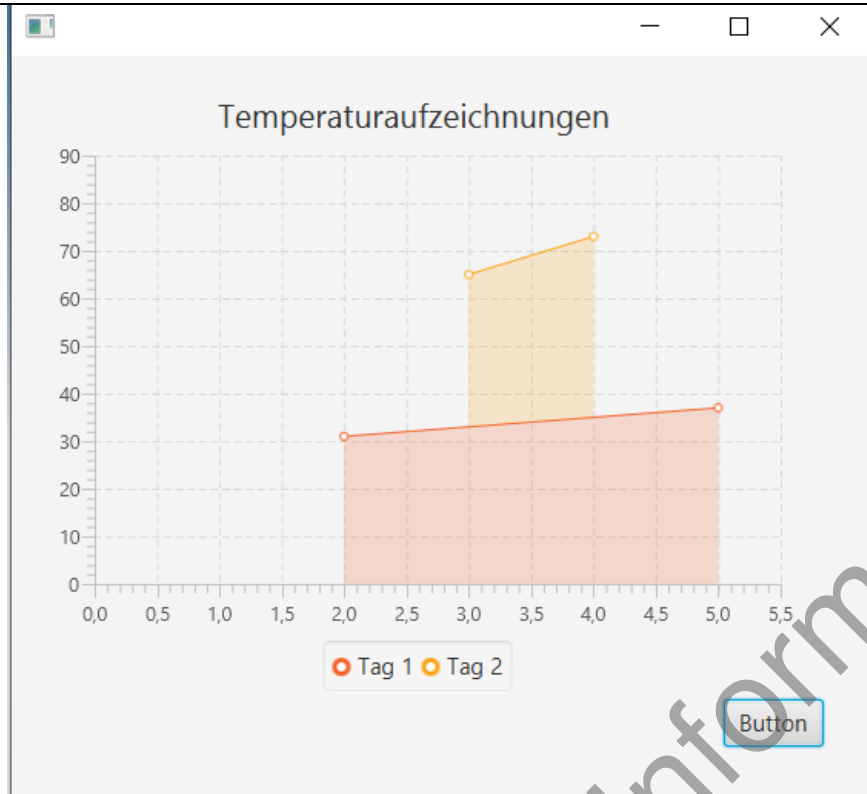
```

@FXML
private void buttonClick(ActionEvent event) {
    (XYChart.Series) chart.getData().get(0)
        .getData().add(new XYChart.Data(6+x, 42+x));
    x++;
}

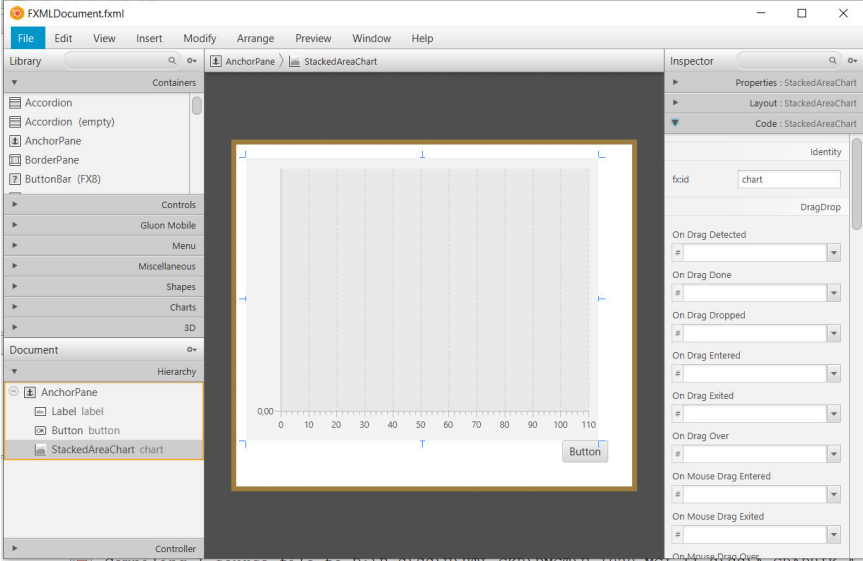
```



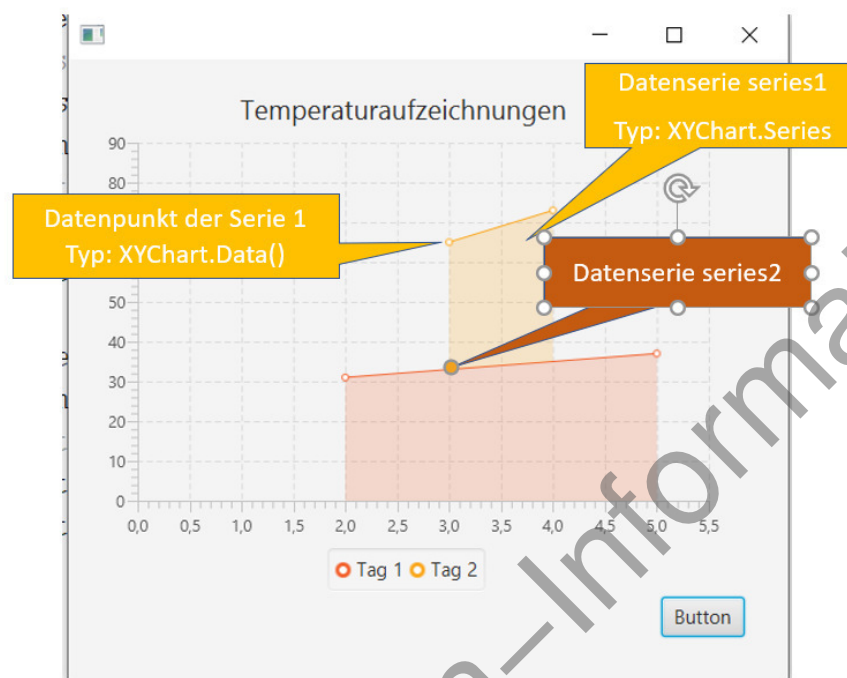
StackedAreaChart



- Das StackedAreaChart unterscheidet sich von dem LineChart dadurch, dass die Flächen zwischen x-Achse und den Verbindungslinien zwischen den Datenpunkten einer Zeitreihe farbig ausgefüllt sind und gestapelt.

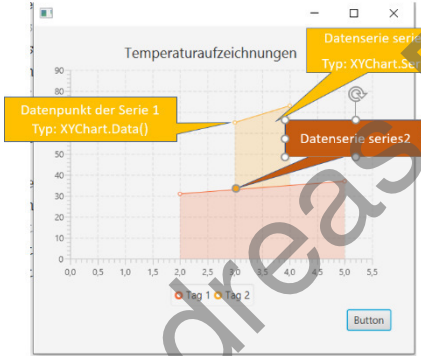
	<p>Mit dem SceneBuilder wird ein StackedAreaChart positioniert. Das Chart erhält eine id.</p>	<p>Beispiel VorzeigenStackedAreaChart (FXMLDocument.fxml)</p>
<pre><StackedAreaChart fx:id="chart" layoutX="14.0" layoutY="19.0"> <xAxis> <NumberAxis side="BOTTOM" /> </xAxis> <yAxis> <NumberAxis side="LEFT" /> </yAxis> </StackedAreaChart></pre>	<p>FXML-Darstellung der Graphik. Ich wähle für dieses Beispiel 2 numerische Achsen. Es kann auch eine CategoryAxis gewählt werden</p>	<p>Beispiel VorzeigenStackedAreaChart (FXMLDocument.fxml)</p>
<p>Für die XYCharts werden eine x- und eine y-Achse definiert.</p>		<p>Beispiel VorzeigenStackedAreaChart (FXMLDocument.fxml)</p>

- Wir werden dabei vor allem mit der NumberAxis zu tun haben, deren Daten als Zahlen definiert sind
- Bei der CategoryAxis, erfolgt die Wertzuordnung anhand von Strings.



Ein StackedAreaChart besteht aus einer oder mehreren Serien. Die Serie besteht aus mehreren Datenpunkten

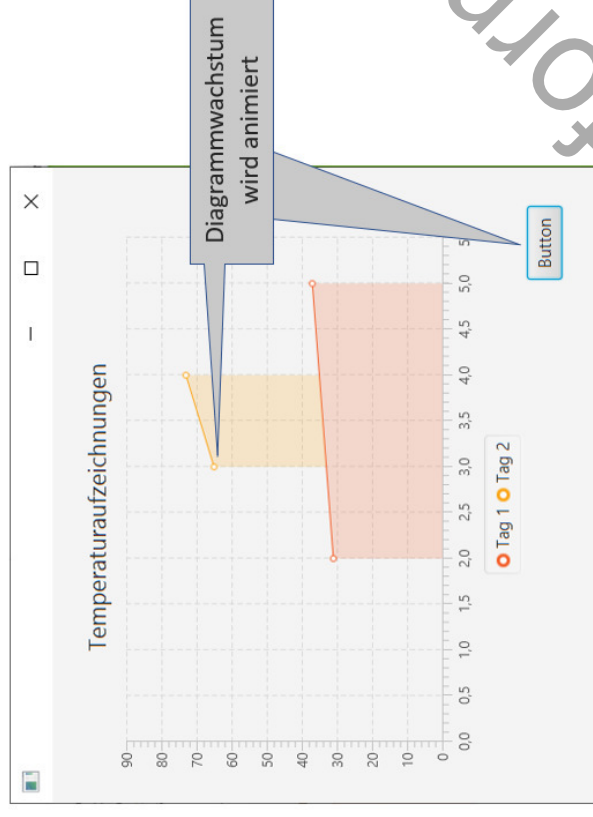
Beispiel
VorzeigenStackedAreaChart
(FXMLDocument.fxml
)

<pre> public void initialize(URL url, ResourceBundle rb) { // TODO chart.setTitle("Temperaturaufzeichnungen "); //defining a series XYChart.Series series1 = new XYChart.Series(); series1.setName("Tag 1"); //populating the series with data series1.getData().add(new XYChart.Data(2, 31)); series1.getData().add(new XYChart.Data(5, 37)); XYChart.Series series2 = new XYChart.Series(); series2.setName("Tag 2"); //populating the series with data series2.getData().add(new XYChart.Data(3, 32)); series2.getData().add(new XYChart.Data(4, 38)); ObservableList<XYChart.Series<Integer, Integer>> FXCollections.observableArrayList(); l.add(series1); l.add(series2); chart.setData(l); </pre> <div data-bbox="846 236 1025 268">Titel des Diagrammes</div> <div data-bbox="853 316 1019 347">Erstellung der Serie</div> <div data-bbox="853 395 1019 427">Erstellung der Serie</div> <div data-bbox="846 619 1012 721">Serien müssen in einer ObservableList zur Verfügung gestellt werden</div> <div data-bbox="571 691 750 738">Setzen der Daten im Diagramm</div> <div data-bbox="786 738 992 770">Abbildung:AreaChart2</div>		<p>Beispiel VorzeigenStackedAreaChart (FXMLDocument.fxml)</p>
	<p>Beschaffung der Serie1 und Hinzufügen eines neuen Datenpunktes(Diagrammwachstum wird animiert)</p>	<p>Beispiel VorzeigenStackedAreaChart (FXMLDocument.fxml)</p>

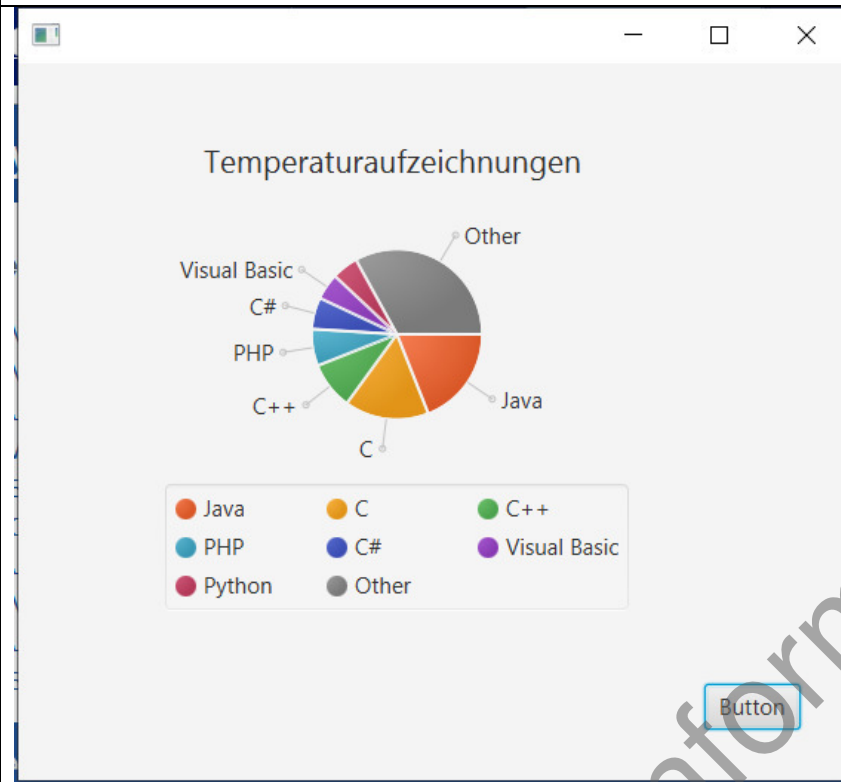

```

@FXML
private void buttonClick(ActionEvent event) {
    XYChart.Data<Integer, Integer> temp
    = (XYChart.Data<Integer, Integer>)
    ((XYChart.Series) chart.getData()
    .get(1)).getData().get(0);
    temp.setYValue(temp.getYValue() + 10);
}

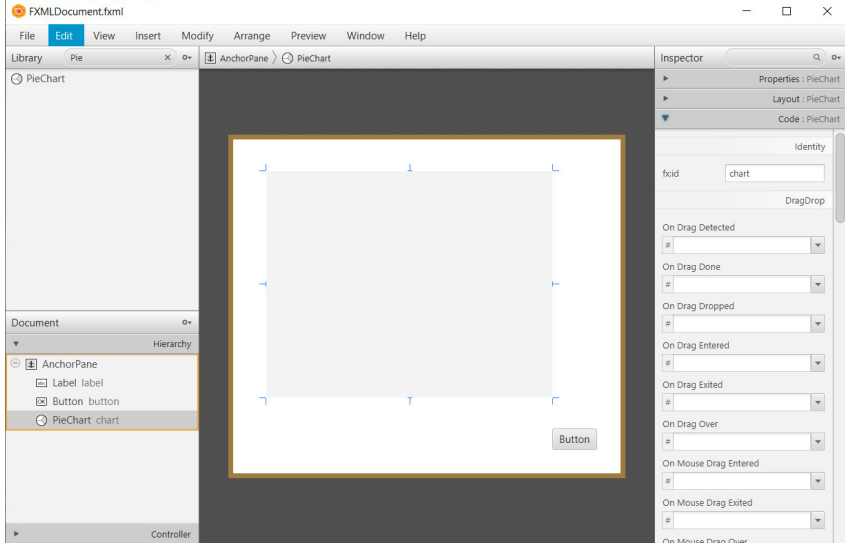
```

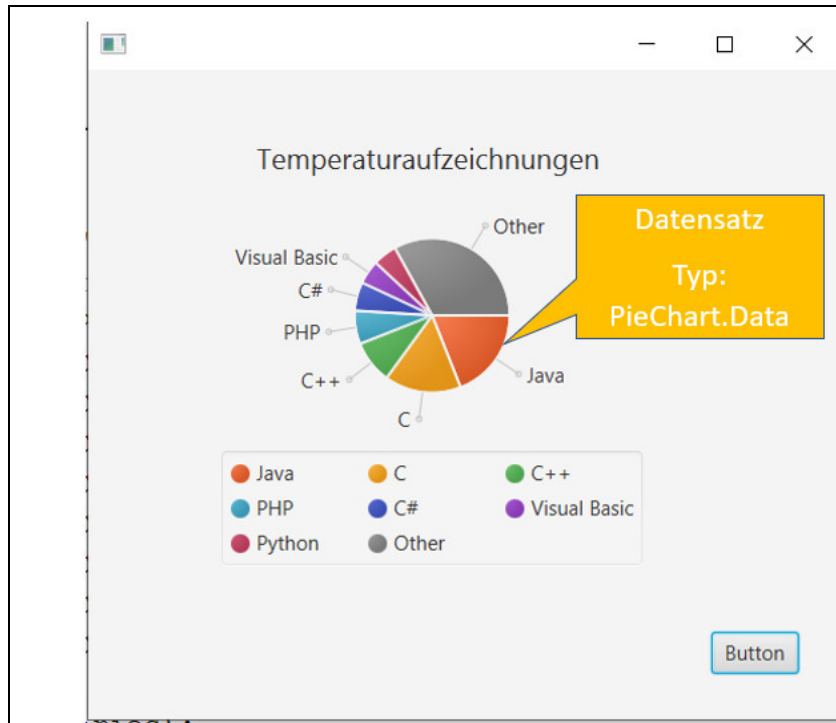


PieChart



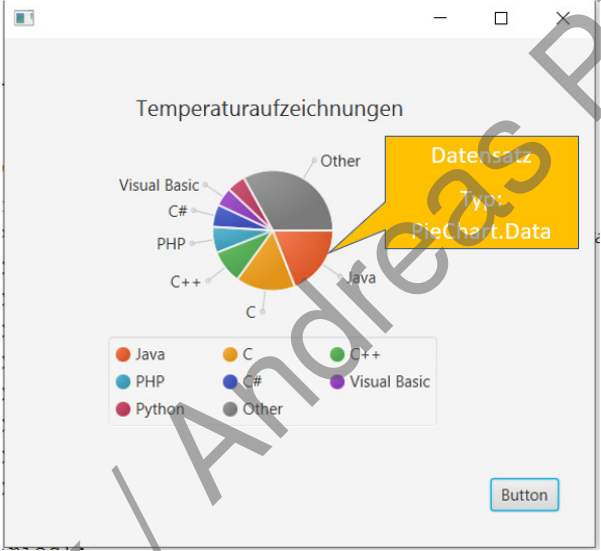
- PieCharts oder Tortendiagramme werden meist verwendet, um den prozentualen Anteil einer Kategorie an einer Gesamtmenge zu verdeutlichen. Sie haben im Unterschied zu LineChart und AreaChart nur eine Dimension und keine Achsen.

	<p>Mit dem SceneBuilder wird ein PieChart positioniert. Das Chart erhält eine id.</p>	<p>Beispiel VorzeigenPieChart (FXMLDocument.fxml)</p>
<pre><PieChart fx:id="chart" layoutX="48.0" layoutY="46.0" prefHeight="328.0" prefWidth="415.0" /></pre>	<p>FXML-Darstellung der Graphik. Ein PieChart hat keine Achsen</p>	<p>Beispiel VorzeigenPieChart (FXMLDocument.fxml)</p>
		<p>Beispiel VorzeigenPieChart (FXMLDocument.fxml)</p>



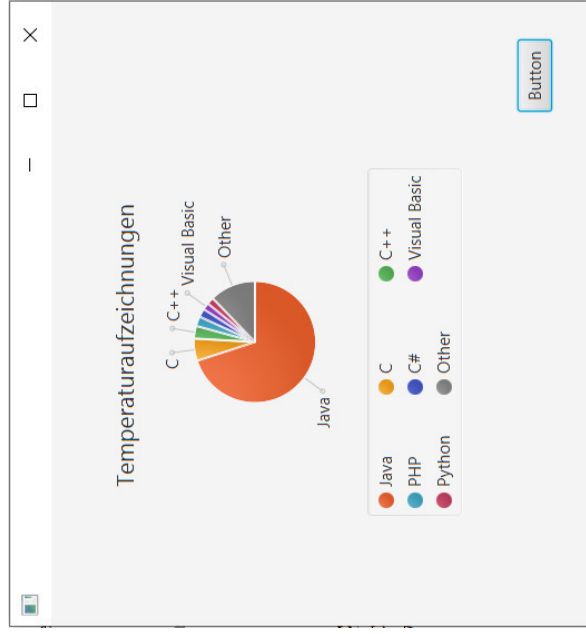
Ein PieChart besteht aus mehreren Segmenten, welche in einer ObservableList zur Verfügung gestellt werden.

Beispiel
VorzeigenPieChart
(FXMLDocument.fxml)

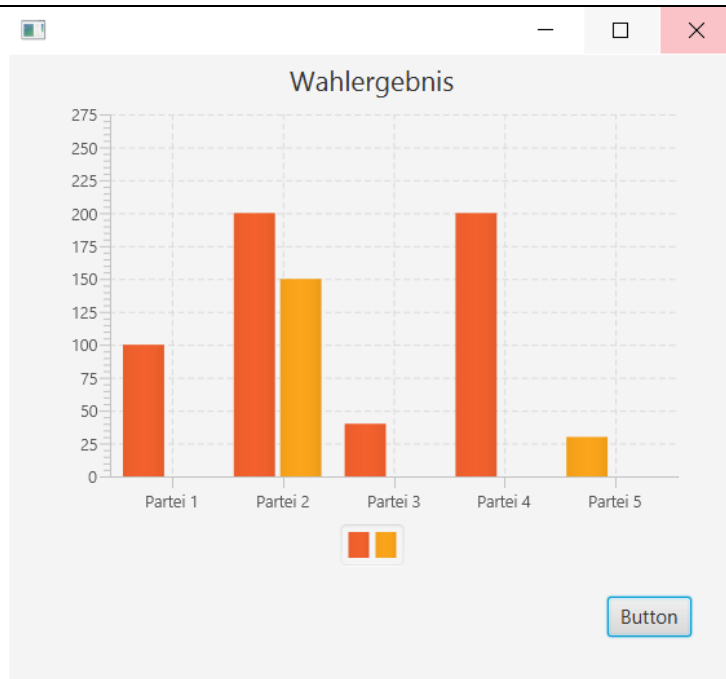
<pre> @Override public void initialize(URL url, ResourceBundle rb) { // TODO chart.setTitle("Temperaturaufzeichnungen"); //defining a series ObservableList<PieChart.Data> series = FXCollections.observableArrayList(new PieChart.Data("Java", 19), new PieChart.Data("C", 16), new PieChart.Data("C++", 9), new PieChart.Data("PHP", 7), new PieChart.Data("C#", 6), new PieChart.Data("Visual Basic", 5), new PieChart.Data("Python", 5), new PieChart.Data("Other", 33)); // chart.setData(series); </pre> <p>Titel des Diagrammes</p> <p>Daten müssen in einer ObservableList zur Verfügung gestellt werden</p> <p>Setzen der Daten im Diagramm</p>		<p>Beispiel VorzeigenPieChart (FXMLDocument.fxml)</p>
	<p>Beschaffung eines Segmentes und Hinzufügen eines neuen Datenpunktes(Diagrammwachstum wird animiert)</p>	<p>Beispiel VorzeigenPieChart (FXMLDocument.fxml)</p>

@FXML

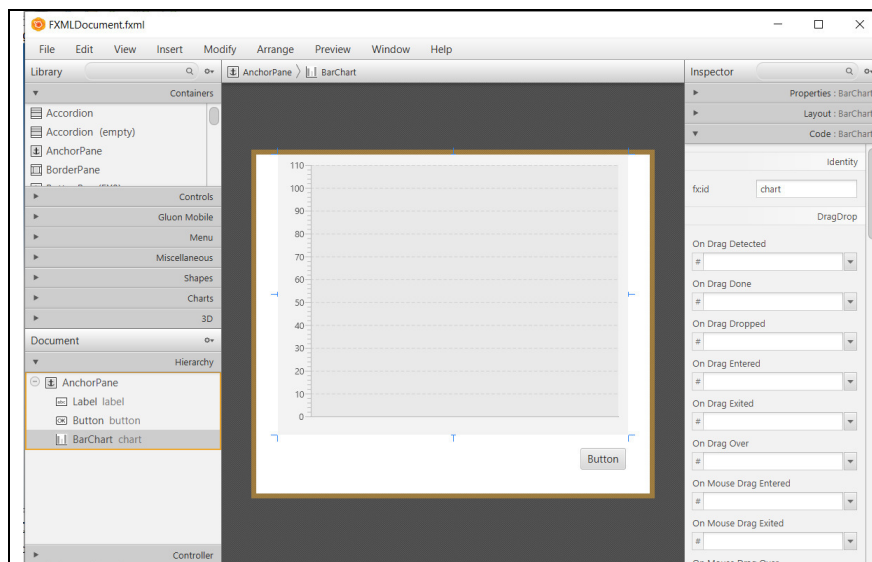
```
private void buttonClick(ActionEvent event) {  
    PieChart.Data dCurrent = chart.getData().get(0);  
    dCurrent.setPieValue (dCurrent.getPieValue()+10);  
}
```



BarChart



- Balken- und Säulendiagramme werden mithilfe der Klasse BarChart erstellt.
- Sie werden verwendet, um Daten verschiedener Kategorien zu vergleichen.



Mit dem SceneBuilder wird ein BarChart positioniert. Das Chart erhält eine id.

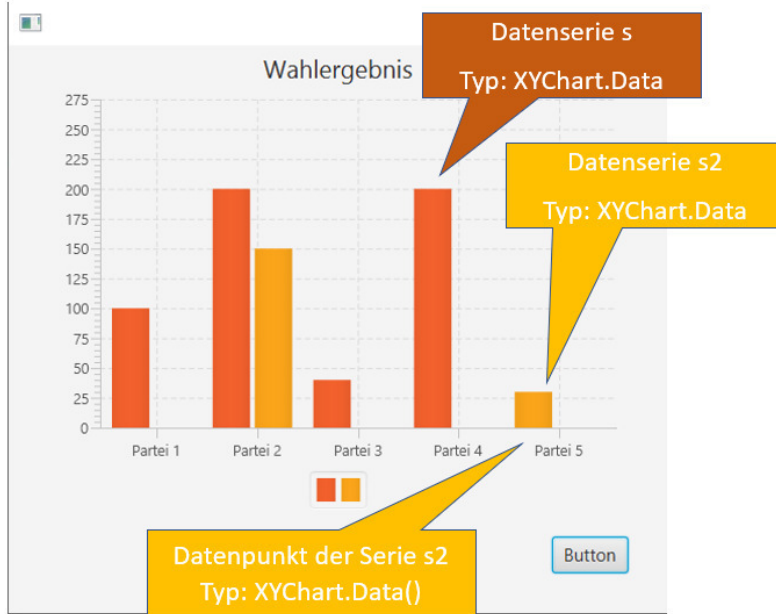
Beispiel
VorzeigenBarChart
(FXMLDocument.fxml)

```
<BarChart fx:id="chart" layoutX="31.0">
  <xAxis>
    <CategoryAxis side="BOTTOM" />
  </xAxis>
  <yAxis>
    <NumberAxis side="LEFT" />
  </yAxis>
</BarChart>
```

Die Klasse erwartet eine Number oder ValueAxis kombiniert mit einer CategoryAxis.

Beispiel
VorzeigenBarChart
(FXMLDocument.fxml)

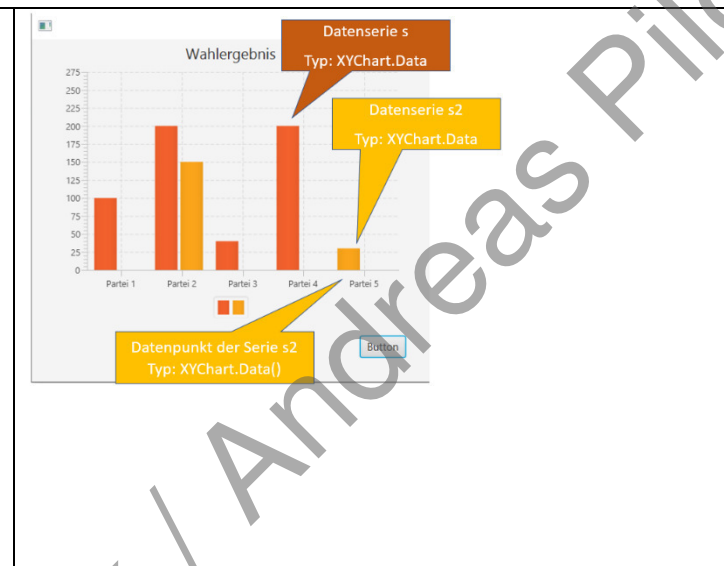
Beispiel
VorzeigenBarChart
(FXMLDocument.fxml)



Ein BarChart besteht aus mehreren Kategorien (hier gelb und orange), welche in einer ObservableList zur Verfügung gestellt werden.

Beispiel
VorzeigenBarChart
(FXMLDocument.fxml)

<pre> @Override public void initialize(URL url, ResourceBundle rb) { chart.setTitle("Wahlergebnis"); //defining a series XYChart.Series<String, Number> s = new XYChart.Series<String, Number>(); ObservableList l = FXCollections.observableArrayList(new XYChart.Data<String, Number>("Partei 1", 100), new XYChart.Data<String, Number>("Partei 2", 200)); l.add(new XYChart.Data<String, Number>("Partei 3", 50)); l.add(new XYChart.Data<String, Number>("Partei 4", 250)); s.setData(l); XYChart.Series<String, Number> s2 = new XYChart.Series<String, Number>(); ObservableList l2 = FXCollections.observableArrayList(new XYChart.Data<String, Number>("Partei 5", 30), new XYChart.Data<String, Number>("Partei 2", 150)); l.add(new XYChart.Data<String, Number>("Partei 3", 40)); l.add(new XYChart.Data<String, Number>("Partei 4", 200)); s2.setData(l2); ObservableList<XYChart.Series<String, Number>> os = FXCollections.observableArrayList(s); chart.getData().add(s); chart.getData().add(s2); } </pre>	<div>Titel des Diagrammes</div> <div>Erstellung der Serie</div> <div>Belegen der Werte bei den Kategorien</div> <div>Setzen der Daten im Diagramm</div>
---	---



Beispiel
VorzeigenBarChart
(FXMLDocument.fxml)

<pre> @FXML private void buttonClick(ActionEvent event) { XYChart.Data<String, Number> dCurrent = chart.getData().get(0).getData().get(0); dCurrent.setYValue(dCurrent.getYValue().intValue()+20); } </pre>	
---	--

Beschaffung eines Balken in einer Serie und verändern des Wertes (Diagrammwachstum wird animiert)

Beispiel
VorzeigenBarChart
(FXMLDocument.fxml)

