

Autor & Copyright: Dipl.-Ing. Harald Nahrstedt

Version: 2016 / 2019 / 2021 / 365

Erstellungsdatum: 09.09.2022

Überarbeitung: 01.12.2023

Beschreibung:

Das *Chart*-Objekt ist ein Diagramm in einem *Report*-Objekt unter *Project*. *Project* verwendet für ein *Chart*-Objekt die gleichen Elemente wie für *Chart*-Objekte in anderen Anwendungen. In *Project* ist ein *Chart*-Objekt ein Unterobjekt eines *Shape*-Objekts oder einer *ShapeRange*-Auflistung.

Anwendungs-Datei: 17-01-04_ChartObjekt.mpp

1 Das Chart-Objekt

Das *Chart*-Objekt ist ein Diagramm in einem *Report*-Objekt unter *Project*. Project verwendet für ein Chart-Objekt die gleichen Elemente wie für Chart-Objekte in anderen Anwendungen. In Project ist ein Chart-Objekt ein Unterobjekt eines *Shape*-Objekts oder einer *ShapeRange*-Auflistung und besitzt keine Ereignisprozeduren.

2 Methoden

Eine Makroaufzeichnung zur Herstellung eines Diagramms ist in Project nicht möglich. Nachfolgend werden die wichtigsten Methoden betrachtet.

Die *Shapes.AddChart*-Methode fügt einem Bericht ein Diagramm hinzu. Mit der *HasChart*-Methode lässt sich feststellen, ob ein *Shape*- oder ein *ShapeRange*-Objekt ein Diagramm enthält.

Codeliste 1. Die Prozedur erstellt ein einfaches Diagramm

```
Sub CreateReport ()
    Dim objReport As Report
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    On Error Resume Next
    ActiveProject.Reports(sName).Delete
    Set objReport = ActiveProject.Reports.Add(sName)
    Set shpChart = ActiveProject.Reports(sName).Shapes.AddChart()
    shpChart.Chart.SetElement(msoElementChartTitleCenteredOverlay)
    shpChart.Chart.ChartTitle.Text = "Vorgangsarbeiten"
    Set objReport = Nothing
    Set shpChart = Nothing
End Sub
```

Das Diagramm zeigt *Ist-Arbeit*, *Restarbeit* und *Arbeitsstandard* der Vorgänge (Bild 1).

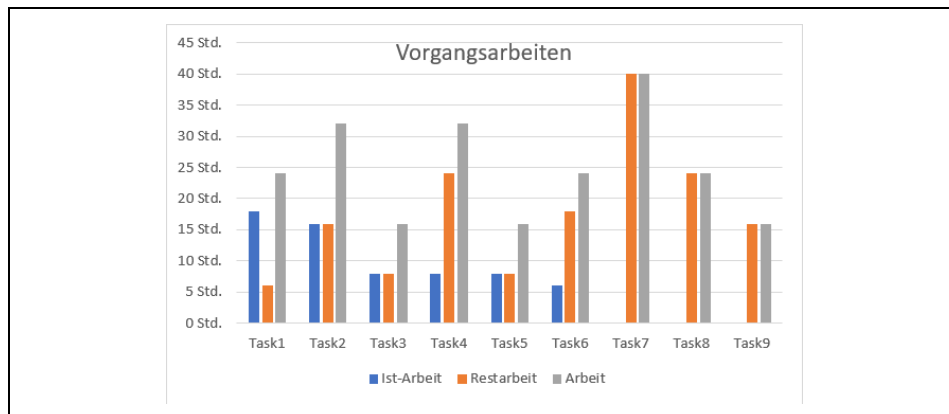


Bild 1. Arbeitswerte der Vorgänge im aktiven Projekt

Die Prozedur *CreateChart* prüft zunächst, ob es einen Report unter dem vorgegebenen Namen bereits gibt, und löscht ihn. Ansonsten wird der Report direkt erstellt. Dank der Anweisung *On Error Resume Next* gibt es keine Fehlermeldung.

Um das Diagramm zu löschen, kann die Form gelöscht werden, die das Diagramm enthält. Die folgende Prozedur löscht lediglich das Diagramm und zeigt dann einen leeren Report.

Codeliste 2. Die Prozedur löscht lediglich das Diagramm

```
Sub DeleteShape()
    Dim iCount As Integer
    Dim sName As String
    Dim shpChart As Shape

    sName = "Vorgangsarbeiten"
    For iCount = 1 To ActiveProject.Reports.Count
        If ActiveProject.Reports(iCount).Name = sName Then
            Set shpChart = ActiveProject.Reports(iCount).Shapes(1)
            shpChart.Delete
            Set shpChart = Nothing
        End If
    Next iCount
End Sub
```

Um auch den Report zu löschen, muss die folgende Prozedur verwendet werden.

Codeliste 3. Die Prozedur löscht auch den Report

```
Sub DeleteReport()
    Dim iCount As Integer
    Dim sName As String

    sName = "Vorgangsarbeiten"
    ViewApplyEx Name:="Gantt-Diagramm"
    ActiveProject.Reports(sName).Delete
End Sub
```

Vor dem Löschen wird zunächst eine andere Ansicht (hier Gantt-Diagramm) gewählt. Die Methode *ApplyCustomType* weist dem *Chart*-Objekt einen *ChartType* zu. Die Syntax lautet

```
ApplyCustomType (ChartType, Typename)
```

Tabelle 1. Die Parameter der ApplyCustomMethode

Parameter	Option	Datentyp	Beschreibung
ChartType	Erforderlich	Office.XlChartType	The type of chart.
TypeName	Optional	Variant	The name of the chart type.

Die folgende Prozedur ändert den Diagrammtyp im zuvor erstellten Report in gestapelte 3D-Säulen (Bild 2).

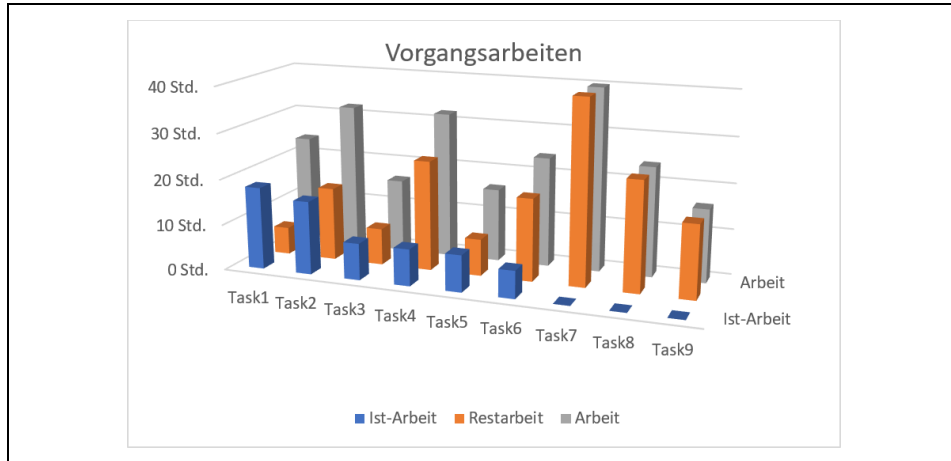


Bild 2. 3Dsäulen-Diagramm

Codeliste 4. Die Prozedur ändert den Diagrammtyp in ein 3Dsäulen-Diagramm

```

Sub ChangeChartType()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.ApplyCustomType (xl3DColumn)
    Set shpChart = Nothing
End Sub

```

Tabelle 2. XLChartType-Konstante

Konstante	Wert	Beschreibung
xl3DArea	-4098	3D Fläche
xl3DAreaStacked	78	3D Fläche (gestapelt)
xl3DAreaStacked100	79	3D Fläche (100 % gestapelt)
xl3DBarClustered	60	3D-Balken (gruppiert)
xl3DBarStacked	61	3D Balken (gestapelt)
xl3DBarStacked100	62	3D Balken (100 % gestapelt)
xl3DColumn	-4100	3D Säulen
xl3DColumnClustered	54	3D Säulen (gruppiert)
xl3DColumnStacked	55	3D Säulen (gestapelt)
xl3DColumnStacked100	56	3D Säulen (100 % gestapelt)
xl3DLine	-4101	3D Linie
xl3DPie	-4102	3D Kreis

Konstante	Wert	Beschreibung
xl3DPieExploded	70	Explodierter 3D Kreis
xlArea	1	Bereich
xlAreaStacked	76	Gestapelte Fläche
xlAreaStacked100	77	Gestapelte Fläche (100%)
xlBarClustered	57	Gruppierte Balken
xlBarOfPie	71	Balken aus Kreis
xlBarStacked	58	Gestapelter Balken
xlBarStacked100	59	Gestapelte Balken (100 %)
xlBubble	15	Blase.
xlBubble3DEffect	87	Blase mit 3D-Effekten
xlColumnClustered	51	Gruppierte Säulen
xlColumnStacked	52	Gestapelte Säulen
xlColumnStacked100	53	Gestapelte Säulen (100 %)
xlConeBarClustered	102	Gruppierte Kegelbalken
xlConeBarStacked	103	Gestapelte Kegelbalken
xlConeBarStacked100	104	Gestapelte Kegelbalken (100 %)
xlConeCol	105	3D Kegelsäulen
xlConeColClustered	99	Gruppierte Kegelsäulen
xlConeColStacked	100	Gestapelte Kegelsäulen
xlConeColStacked100	101	Gestapelte Kegelsäulen (100 %)
xlCylinderBarClustered	95	Gruppierte Zylinderbalken
xlCylinderBarStacked	96	Gestapelte Zylinderbalken
xlCylinderBarStacked100	97	Gestapelte Zylinderbalken (100 %)
xlCylinderCol	98	3D Zylindersäulen
xlCylinderColClustered	92	Gruppierte Zylindersäulen
xlCylinderColStacked	93	Gestapelte Zylindersäulen
xlCylinderColStacked100	94	Gestapelte Zylindersäulen (100 %)
xlDoughnut	-4120	Krapfen.
xlDoughnutExploded	80	Explodierter Ring
xlLine	4	Linie.
xlLineMarkers	65	Linie mit Datenpunkten

Konstante	Wert	Beschreibung
xlLineMarkersStacked	66	Gestapelte Linie mit Datenpunkten
xlLineMarkersStacked100	67	Gestapelte Linie mit Datenpunkten (100 %)
xlLineStacked	63	Gestapelte Linie
xlLineStacked100	64	Gestapelte Linie (100 %)
xlPie	5	Pie.
xlPieExploded	69	Explodierter Kreis
xlPieOfPie	68	Kreis aus Kreis
xlPyramidBarClustered	109	Gruppierte Pyramidenbalken
xlPyramidBarStacked	110	Gestapelte Pyramidenbalken
xlPyramidBarStacked100	111	Gestapelte Pyramidenbalken (100 %)
xlPyramidCol	112	3D Pyramidensäulen
xlPyramidColClustered	106	Gruppierte Pyramidensäulen
xlPyramidColStacked	107	Gestapelte Pyramidensäulen
xlPyramidColStacked100	108	Gestapelte Pyramidensäulen (100 %)
xlRadar	-4151	Radar.
xlRadarFilled	82	Gefülltes Netz
xlRadarMarkers	81	Netz mit Datenpunkten
xlRegionMap	140	Kartendiagramm.
xlStockHLC	88	High-Low-Close.
xlStockOHLC	89	Öffnen-Hoch-Niedrig-Schließen.
xlStockVHLC	90	Volume-High-Low-Close.
xlStockVOHLC	91	Volume-Open-High-Low-Close.
xlSurface	83	3D Oberfläche
xlSurfaceTopView	85	Oberfläche (Draufsicht)
xlSurfaceTopViewWireframe	86	Oberfläche (Draufsicht, Drahtmodell)
xlSurfaceWireframe	84	3D Oberfläche (Drahtmodell)
xlXYScatter	-4169	Scatter.
xlXYScatterLines	74	Punkte mit Linien
xlXYScatterLinesNoMarkers	75	Punkte mit Linien ohne Datenpunkte
xlXYScatterSmooth	72	Punkte mit interpolierten Linien
xlXYScatterSmoothNoMarkers	73	Punkte mit interpolierten Linien ohne Datenpunkte

Die Methode *ApplyDataLabels* blendet die Werte mit jedem Aufruf ein und aus. Die Syntax lautet

```
ApplyDataLabels (Typ, IMsoLegendKey, AutoText, HasLeaderLines,
ShowSeriesName, ShowCategoryName, ShowValue, ShowPercentage,
ShowBubbleSize, Separator)
```

Tabelle 3. Die optionalen Parameter der ApplyDataLabels-Methode

Parameter	Datenart	Beschreibung
Typ	Office.XIDataLabelsType	Die Art des Datenetiketts, das angewendet wird. Der Standardwert ist xlDataLabelsShowValue.
IMsoLegendKey	Variante	Getreu dem Punkt, den Legendenschlüssel zu zeigen. Der Standardwert ist False.
AutoText	Variante	Es stimmt, wenn das Objekt automatisch einen geeigneten Text basierend auf dem Inhalt generiert.
HasLeaderLines	Variante	Es stimmt, wenn die Serie Führungslinien hat.
ShowSeriesName	Variante	Getreu dem Seriennamen für das Datenlabel zu ermöglichen. Falsch, um den Seriennamen zu verkabeln.
ShowCategoryName	Variante	Getreu, den Kategorienamen für das Datenlabel zu ermöglichen. Falsch, um den Kategorienamen zu stören.
ShowValue	Variante	Getreu dem Wert für das Datenlabel. Falsch, um den Wert zu unterzuwerten. Wenn der TypeTyp-Parameter nicht angegeben ist, wird ShowValue als wahr angenommen.
ZeigenPercentage	Variante	Getreu, den Prozentsatz für das Datenlabel zu ermöglichen. Falsch, um den Prozentsatz zu vermindern.
ShowBubbleSize	Variante	Getreu, um die Blasengröße für das Datenetikett zu ermöglichen. Falsch, um die Blasengröße zu verkabeln.
Separator	Variante	Der Trenner für das Datenlabel.

Die Methode *ApplyDataLabels* wird auf jeden Datenpunkt angewendet (Bild 3).

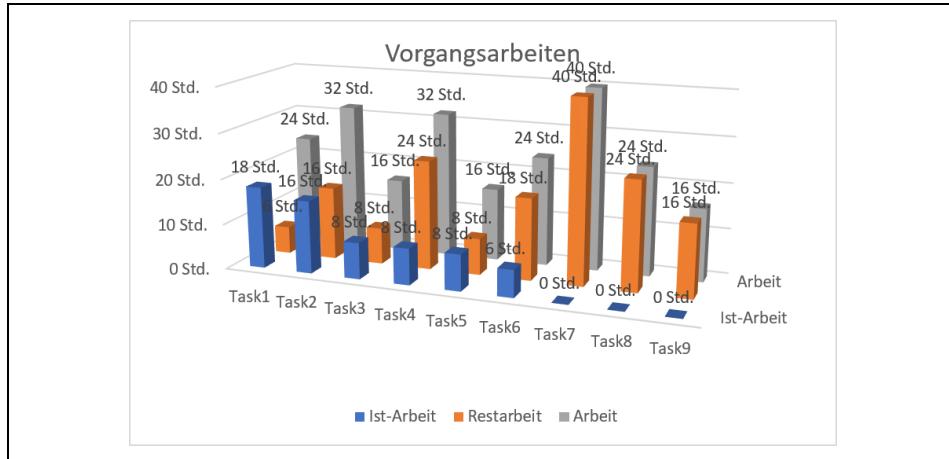


Bild 3. Eingblendete Datenwerte

Codeliste 5. Die Prozedur blendet die Datenwerte mit jedem Aufruf ein und aus

```

Sub SetDataLabels ()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.ApplyDataLabels
    Set shpChart = Nothing
End Sub

```

Die Methode *ApplyLayout* ändert das Chart-Layout. Ihre Syntax lautet

```
ApplyLayout (Layout, varChartType)
```

Tabelle 4. Die Parameter der *ApplyLayout*-Methode

Parameter	Option	Datentyp	Beschreibung
Layout	Required	Long	Gibt den Layouttyp an, wobei der Wert einem Schnelllayoutelement im Menüband entspricht.
varChartType	Optional	Variant	Kann eine der Office.XlChartType-Konstanten sein.

Die folgende Prozedur ändert das Layout des *Chart*-Objekts (Bild 4).

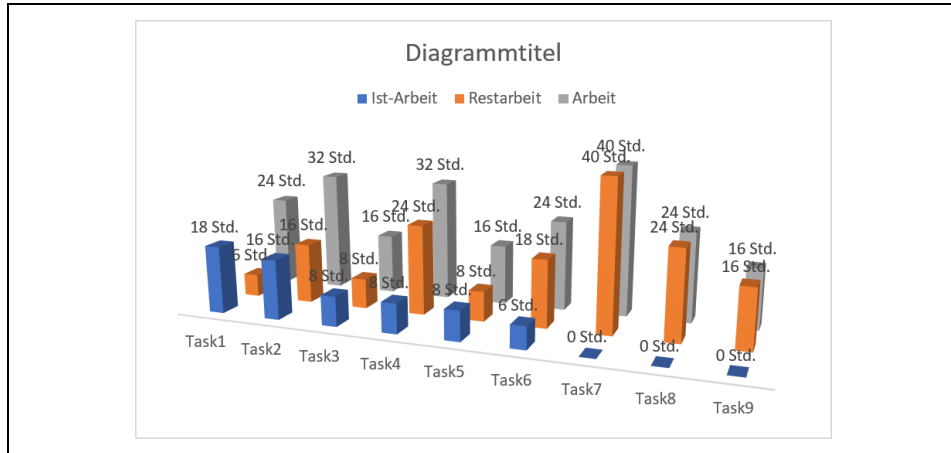


Bild 4. Geändertes Layout

Codeliste 6. Die Prozedur ändert das Layout des Chart-Objekts

```

Sub ChangeChartLayout ()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.ApplyLayout Layout:=2, _
        varChartType:=Office.XlChartType.xlLine
    Set shpChart = Nothing
End Sub

```

Die Methode *AutoFormat* ändert das Diagramm in ein Standardformat für einen anderen Diagrammtyp. Die Syntax lautet

```
AutoFormat (rGallery, varFormat)
```

Tabelle 5. Die Parameter der *AutoFormat*-Methode

Parameter	Option	Datentyp	Beschreibung
rGallery	Erforderlich	Long	Eine der Office.XlChartType-Konstanten, die den Diagrammtyp angibt.
varFormat	Optional	Variant	Die Optionsnummer für die integrierten Autoformate. Kann je nach Galerietyp eine Zahl zwischen 1 und 10 sein. Wenn das Argument <i>varFormat</i> weggelassen wird, wählt Project einen Standardwert basierend auf dem Katalogtyp und der Datenquelle.

Die folgende Prozedur ändert das Diagramm in das Standardformat *3D-Stacked Area* (Bild 5).

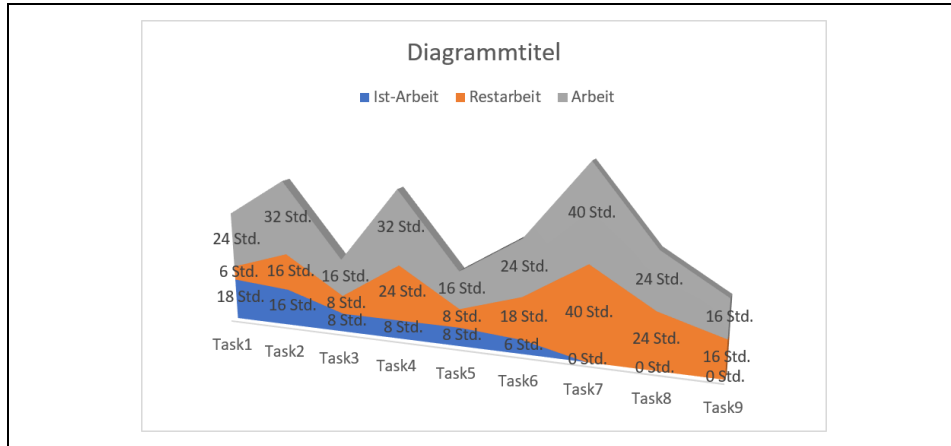


Bild 5. Diagrammformat 3Dfläche

Codeliste 7. Die Prozedur ändert das Standardformat

```

Sub ChangeAutoFormat ()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(reportName).Shapes(1)
    shpChart.Chart.AutoFormat Office.XlChartType.xl3DAreaStacked
    Set shpChart = Nothing
End Sub

```

Die Methode *Axes* spricht die Achsen des Diagramms an. Die Syntax lautet

```
Axes (Type, AxisGroup)
```

Tabelle 6. Die Parameter der *Axes*-Methode

Parameter	Option	Datentyp	Beschreibung
Type	Optional	Variant	Gibt die zurückzugebende Achse an. Kann eine der folgenden <i>Office.XlAxisType</i> -Konstanten sein: <i>xlValue</i> , <i>xlCategory</i> oder <i>xlSeriesAxis</i> (<i>xlSeriesAxis</i> ist nur für 3D-Diagramme gültig).
AxisGroup	Optional	Office.XlAxisGroup	Gibt die Achsengruppe an. Der Standardwert ist <i>xlPrimary</i> ; Das heißt, wenn das <i>AxisGroup</i> -Argument weggelassen wird, wird die primäre Gruppe verwendet. 3D-Diagramme haben nur eine Achsengruppe.

Die folgende Prozedur fügt den Titel der x-Achse ein (Bild 6).

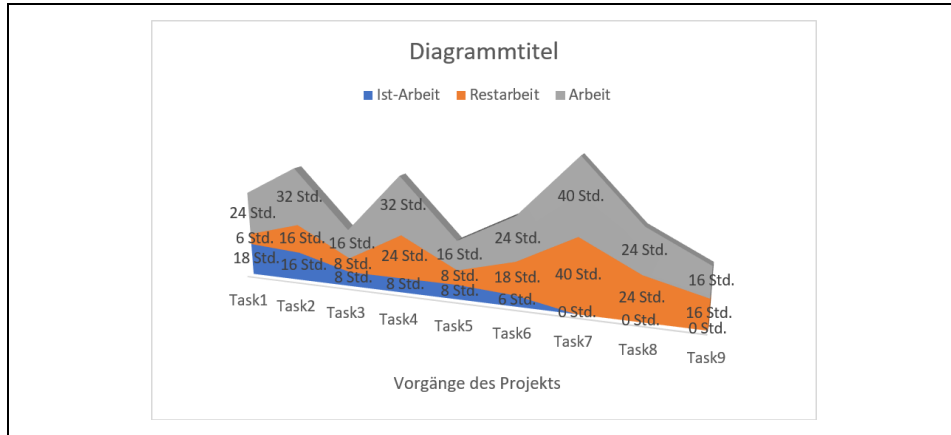


Bild 6. Eingfügter Titel zur x-Achse

Codeliste 8. Die Prozedur fügt den Titel der x-Achse ein

```

Sub SetAxisTitle()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    With shpChart.Chart.Axes(xlCategory)
        .HasTitle = True
        .AxisTitle.Text = "Vorgänge des Projekts"
    End With
    Set shpChart = Nothing
End Sub

```

Die folgende Prozedur fügt Gitterlinien zur Kategorieachse im Diagramm hinzu (Bild 7).

Codeliste 9. Die Prozedur fügt Gitterlinien ein

```

Sub AddCategoryGridlines()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.Axes(xlCategory).HasMajorGridlines = True
    Set shpChart = Nothing
End Sub

```

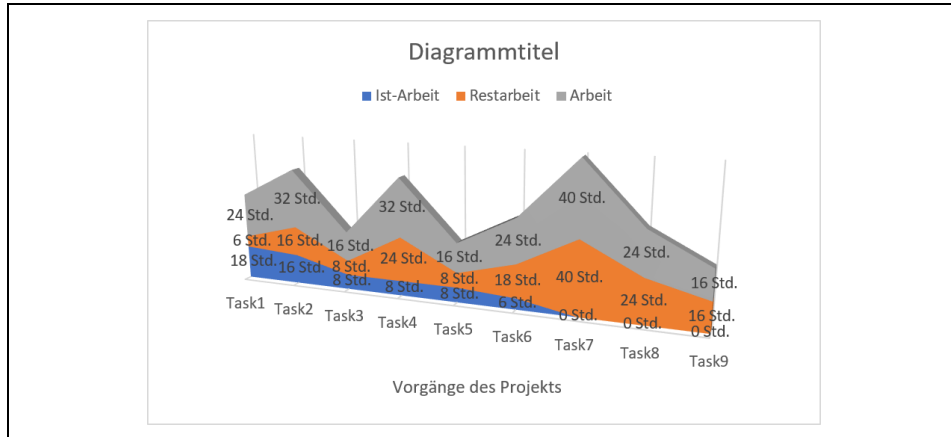


Bild 7. Eingefügte Gitternetzlinien

Die folgende Prozedur entfernt die wichtigsten Gitterlinien sowohl von der Kategorie- als auch von der Werteachse (Bild 8).

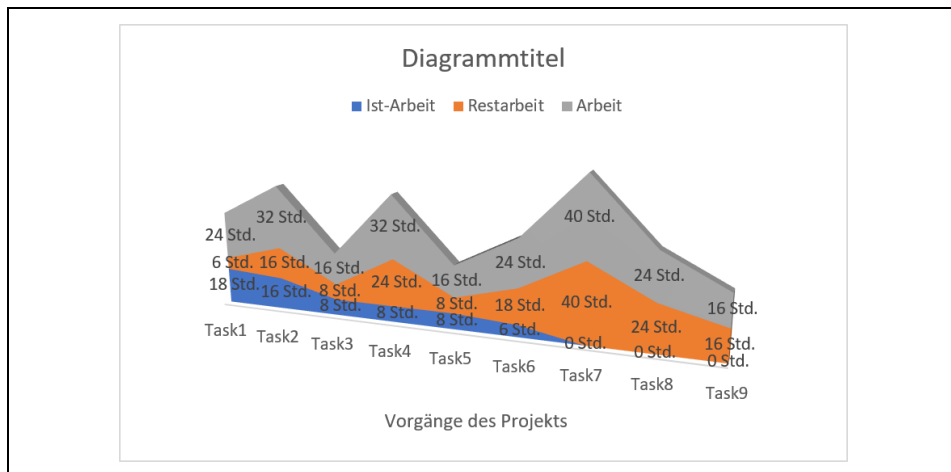


Bild 8. Diagramm mit entfernten Gitterlinien

Codeliste 10. Die Prozedur entfernt die Gitterlinien im Diagramm

```

Sub RemoveGridlines ()
    Dim shpChart As Shape
    Dim sName As String
    Dim axes As Object
    Dim a As Object

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.axes(xlCategory).HasMajorGridlines = False
    shpChart.Chart.axes(xlValue).HasMajorGridlines = False
    Set shpChart = Nothing
End Sub

```

Die Methode *ChartWizard* ändert die Eigenschaften und Formatierung eines Diagramms. Die Syntax lautet

```
ChartWizard (varSource, varGallery, varFormat, varPlotBy,
varCategoryLabels, varSeriesLabels, varHasLegend, varTitle,
varCategoryTitle, varValueTitle, varExtraTitle)
```

Tabelle 7. Die optimalen Parameter der *ChartWizard*-Methode

Parameter	Datentyp	Beschreibung
varSource	Variant	Die Quelldaten für ein neues Diagramm. Wenn das Argument <i>varSource</i> weggelassen wird, bearbeitet Project den aktiven Bericht oder das ausgewählte Diagramm im aktiven Bericht.
varGallery	Variant	Eine der <i>Office.XlChartType</i> -Konstanten, die den Diagrammtyp angibt.
varFormat	Variant	Die Optionsnummer für die integrierten Autoformate. Kann je nach Galerietyp eine Zahl zwischen 1 und 10 sein. Wenn das Argument <i>varFormat</i> weggelassen wird, wählt Project einen Standardwert basierend auf dem Katalogtyp und der Datenquelle.
varPlotBy	Variant	Gibt an, ob die Daten für jede Reihe in Zeilen oder Spalten vorliegen. Kann eine der folgenden <i>Office.XlRowCol</i> -Konstanten sein: <i>xlRows</i> oder <i>xlColumns</i> .
varCategoryLabels	Variant	Eine Ganzzahl, die die Anzahl der Zeilen oder Spalten im Quellbereich angibt, die Kategoriebezeichnungen enthalten. Die Werte können zwischen 0 (Null) und eins kleiner als die maximale Anzahl der entsprechenden Kategorien oder Serien liegen.
varSeriesLabels	Variant	Eine Ganzzahl, die die Anzahl der Zeilen oder Spalten im Quellbereich angibt, die Serienbezeichnungen enthalten. Die Werte können zwischen 0 (Null) und eins kleiner als die maximale Anzahl der entsprechenden Kategorien oder Serien liegen.
varHasLegend	Variant	True schließt eine Legende ein.
varTitle	Variant	Diagrammtitel
varCategoryTitle	Variant	Titel der Kategorieachse
varValueTitle	Variant	Titel der Wertachse
varExtraTitle	Variant	Der Titel der Reihenachse für 3D-Diagramme oder der Titel der zweiten Wertachse für 2D-Diagramme.

Die folgende Prozedur formatiert das Diagramm im aktiven Bericht als Liniendiagramm neu, fügt eine Legende hinzu und fügt Kategorie- und Wertachsentitel hinzu (Bild 9).

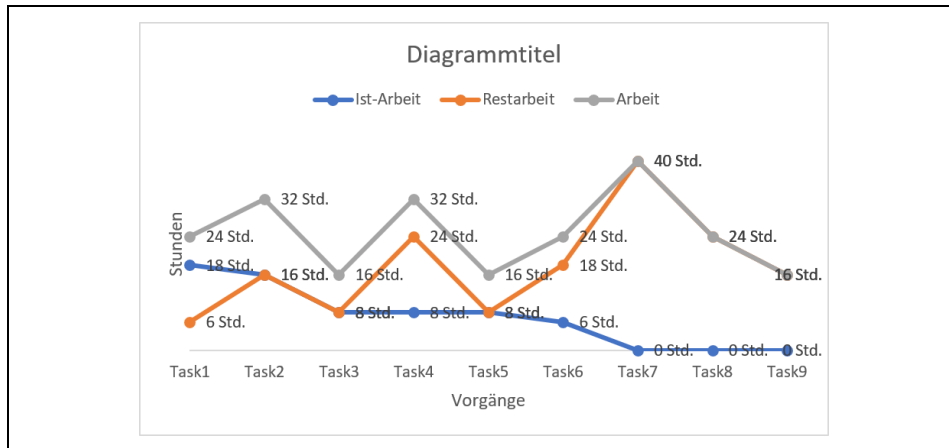


Bild 9. Diagramm nach der Anwendung von ChartWizard

Codeliste 11. Die Prozedur ändert Eigenschaften und Formatierung des Diagramms

```

Sub TestChartWizard()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.ChartWizard _
        varGallery:=xlLine, _
        varHasLegend:=True, _
        varCategoryTitle:="Vorgänge", _
        varValueTitle:="Stunden"
    Set shpChart = Nothing
End Sub

```

Die Methode Copy fügt das Diagramm in die Zwischenablage. Die folgende Prozedur kopiert das Diagramm in die Zwischenablage und fügt es von da als Bild in den Report (Bild 10).

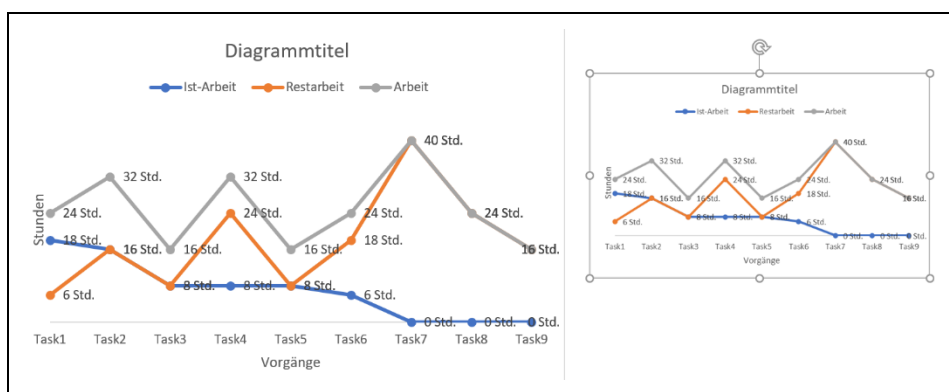


Bild 10. Eingefügtes Bild aus der Zwischenablage

Codeliste 12. Die Prozedur kopiert und fügt ein Bild ein

```
Sub CopyAndPasteChart()  
    Dim shpChart As Shape  
    Dim sName As String  
    Dim chrKopie As Chart  
  
    sName = "Vorgangsarbeiten"  
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)  
    shpChart.Chart.Copy  
    Application.PasteAsPicture  
    Set shpChart = Nothing  
End Sub
```

Die Methode *Refresh* zeichnet ein Diagramm neu, um aktualisierte Informationen anzuzeigen.

3 Eigenschaften

Die Eigenschaft *ChartColor* ruft den Index der Diagrammfarben für das aktive Diagramm ab, oder legt diesen fest. Die folgende Prozedur legt das Farbschema des Diagramms auf monochromatisch Grün fest (Bild 11).

Codeliste 13. Die Prozedur legt das Farbschema des Chart-Objekts fest

```
Sub SetChartColor()  
    Dim shpChart As Shape  
    Dim sName As String  
  
    sName = "Vorgangsarbeiten"  
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)  
    shpChart.Chart.ChartColor = 26  
End Sub
```

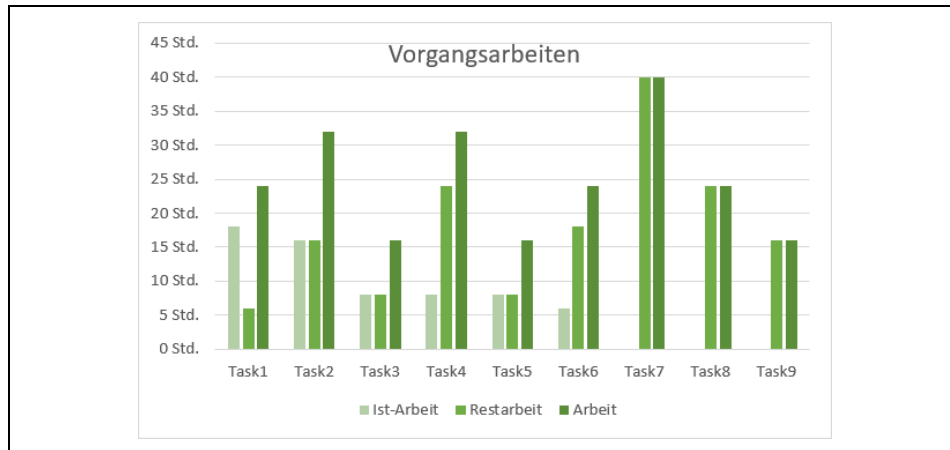


Bild 11. Diagramm im monochromatischen Grün

Die Eigenschaft *ChartStyle* ruft den Diagrammstil für ein Diagramm ab oder legt diesen fest. Die folgende Prozedur besitzt eine *For-Next*-Schleife, in der verschiedene Werte der *ChartStyle*-Eigenschaft zugewiesen werden. Die *Stop*-Anweisung lässt die Betrachtung

der Änderungen zu (Bild 12). Mit der F5-Taste wird nach einem Stop die Prozedur fortgeführt.

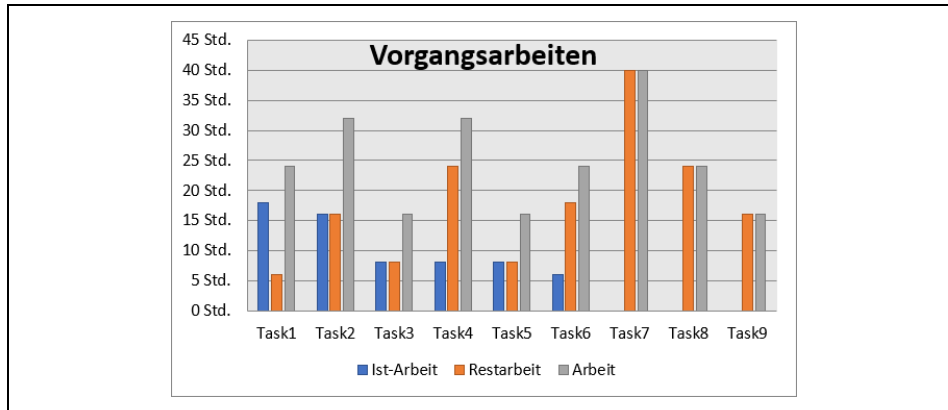


Bild 12. Diagramm mit *ChartStyle = 34*

Codeliste 14. Die Prozedur weist dem Diagramm unterschiedliche *ChartStyle* zu

```

Sub CycleThroughStyles ()
    Dim shpChart As Shape
    Dim sName As String
    Dim iStyle As Integer

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    For iStyle = 1 To 48
        shpChart.Chart.ChartStyle = iStyle
        shpChart.Chart.Refresh
        Stop
    Next iStyle
    Set shpChart = Nothing
End Sub

```

Die Eigenschaft *ChartTitle* ruft ein *Office.InsoChartTitle*-Objekt ab, das den Titel des angegebenen Diagramms darstellt. Die folgende Prozedur ändert den Diagrammtitel und legt seine Position über dem Diagramm fest (Bild 13).

Codeliste 15. Die Prozedur ändert den Diagrammtitel

```

Sub ChangeChartTitle ()
    Dim shpChart As Shape
    Dim sName As String
    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    With shpChart.Chart
        If Not .HasTitle Then
            .HasTitle = True
        End If
        .ChartTitle.Text = "Arbeitsanteile"
        .SetElement (msoElementChartTitleAboveChart)
    End With
    Set shpChart = Nothing
End Sub

```

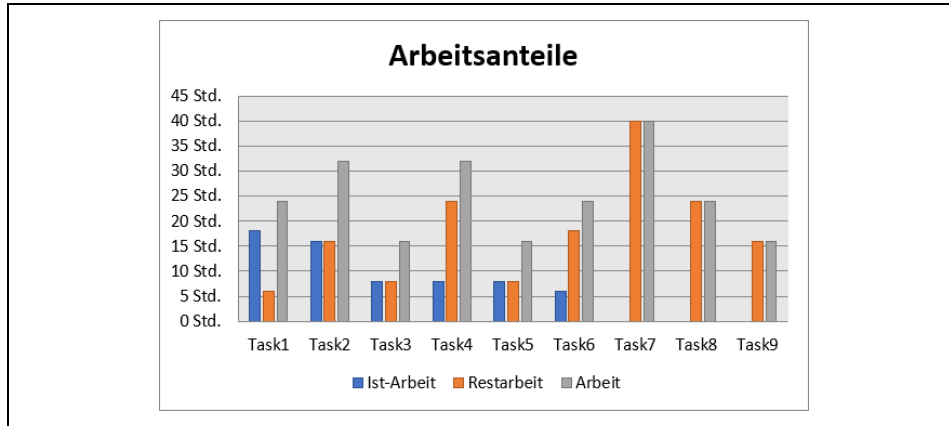



Bild 13. Neuer Diagrammtitel

Die Eigenschaft ChartType ruft den Diagrammtyp ab oder legt in fest. Die folgende Prozedur ändert den Diagrammtyp in 3D-Säulendiagrammtyp (Bild 14).

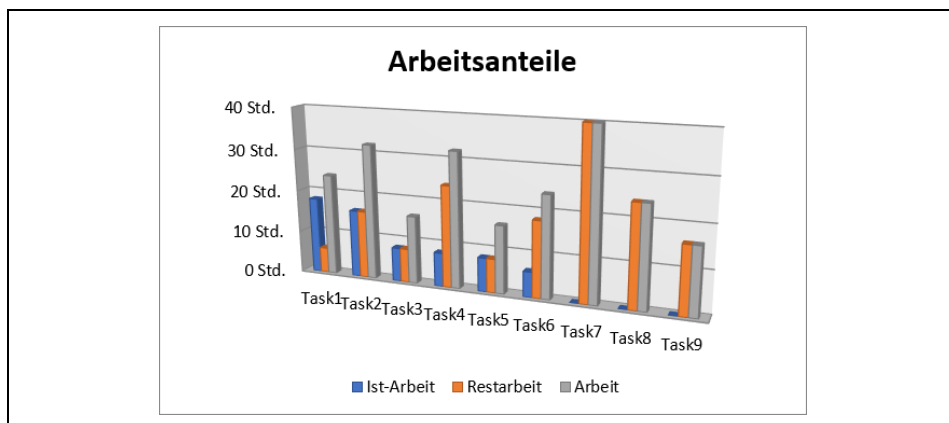


Bild 14. Neuer ChartType 3D-Säulen

Codeliste 16. Die Prozedur ändert den Diagrammtyp

```

Sub SwitchChartTo3D()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    With shpChart.Chart
        Debug.Print .ChartType
        If .ChartType = xlColumnClustered Then
            .ChartType = xl3DColumnClustered
        End If
    End With
    Set shpChart = Nothing
End Sub

```

Die Eigenschaft *DepthPercent* ruft die Tiefe eines 3D-Diagramms als Prozentsatz der Diagrammbreite (zwischen 20 % und 2000 %) ab oder legt sie fest. Die folgende Prozedur legt die Tiefe des angegebenen Diagramms auf 500 % seiner Breite fest (Bild 15).

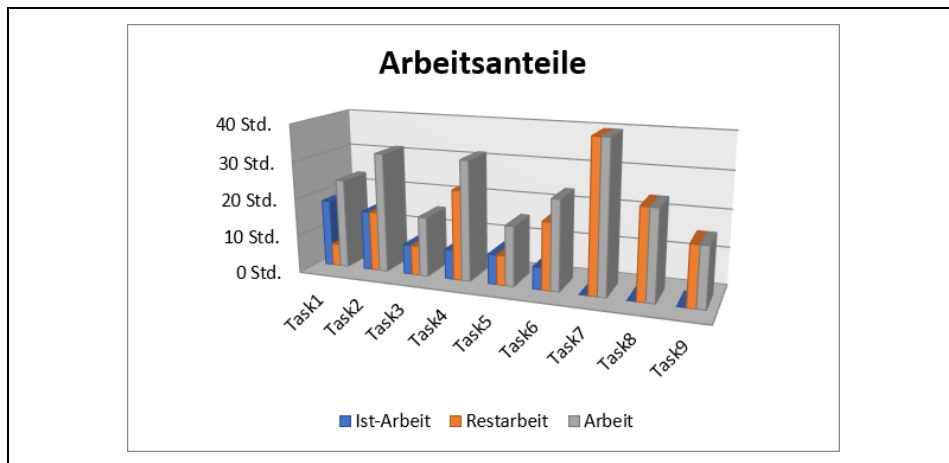


Bild 15. Veränderte Tiefe in der Darstellung

Codeliste 17. Die Prozedur ändert die Tiefe eines 3D-Diagramms

```
Sub SetDepthPercent()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.DepthPercent = 500
    Set shpChart = Nothing
End Sub
```

Die Eigenschaft *Elevation* ruft die Höhe eines 3D-Diagramms in Grad ab oder legt sie fest. Die folgende Prozedur legt die Höhe des Diagramms auf 40 Grad fest (Bild 16).

Codeliste 18. Die Prozedur verändert die Höhe eines 3D-Diagramms

```
Sub SetElevation()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.Elevation = 40
    Set shpChart = Nothing
End Sub
```

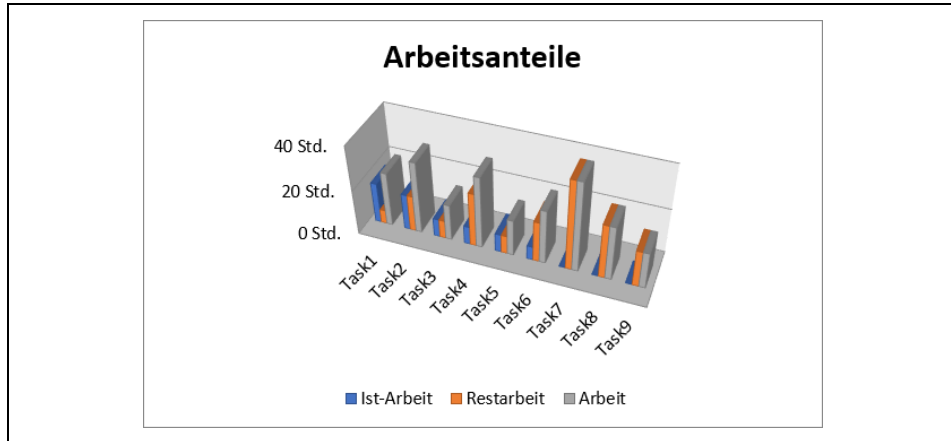


Bild 16. Veränderte Höhe in der Darstellung

Die Eigenschaft Floor ruft ein Office.ImsoFloor-Objekt ab, das den Boden eines 3D-Diagramms darstellt. Die folgende Prozedur setzt die Bodenfarbe des Diagramms auf Blau (Bild 17).

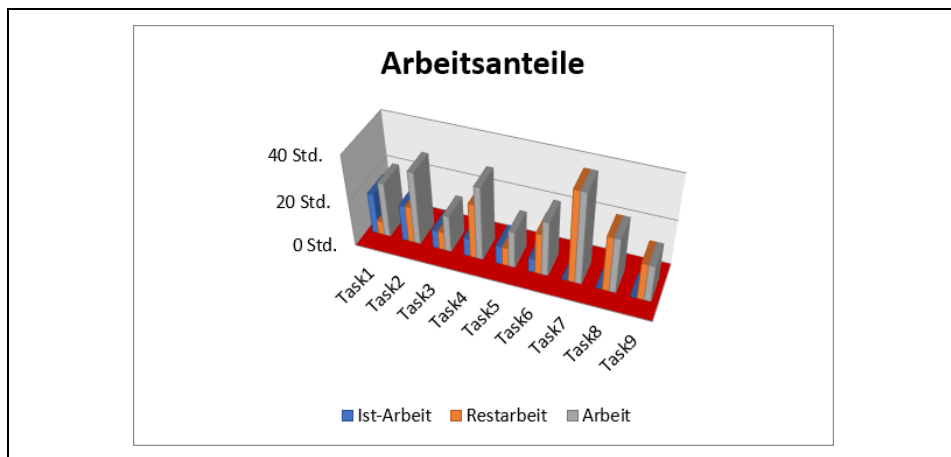


Bild 17. Veränderte Bodenfarbe

Codeliste 19. Die Prozedur verändert die Bodenfarbe eines 3D-Diagramms

```

Sub SetFloorColor()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.Floor.Interior.ColorIndex = 5
    Set shpChart = Nothing
End Sub

```

Die Eigenschaft *GapDepth* ruft den Abstand zwischen den Datenreihen in einem 3D-Diagramm als Prozentsatz der Markierungsbreite ab oder legt ihn fest. Die folgende Prozedur legt den Abstand zwischen den Datenreihen im Diagramm auf 200 % der Markierungsbreite fest (Bild 18).

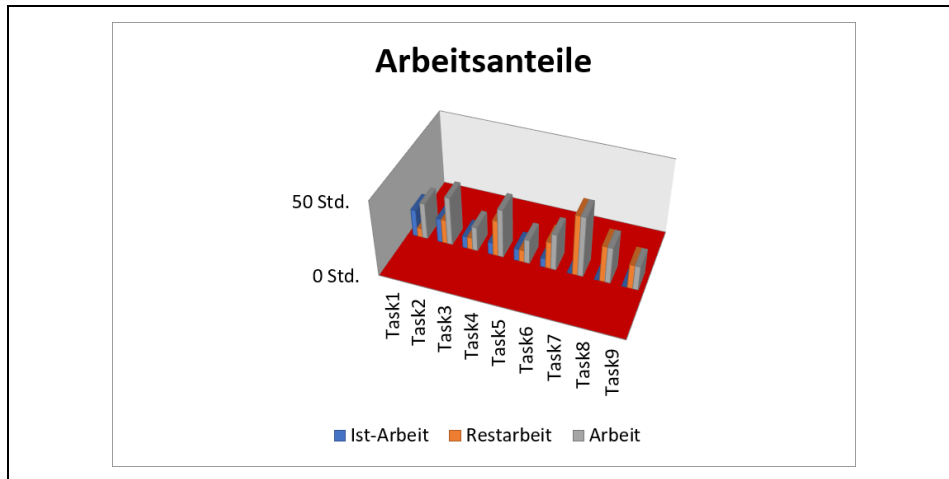


Bild 18. Veränderter Abstand zwischen den Datenreihen

Codeliste 20. Die Prozedur verändert den Abstand zwischen den Datenreihen

```
Sub SetGapDepth()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.GapDepth = 500
    Set shpChart = Nothing
End Sub
```

Die Eigenschaft *PlotArea* ruft ein *Office.IMsoPlotArea*-Objekt ab, das den Plotbereich eines Diagramms darstellt. Die folgende Prozedur vergrößert die Innenhöhe des Plotbereichs um 100 Punkte größer als zuvor.

Codeliste 21. Die Prozedur verändert den Plotbereich im Diagramm

```
Sub SetChartPlotAreaHeight()
    Dim shpChart As Shape
    Dim sName As String
    Dim dHeight As Double

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    dHeight = shpChart.Chart.PlotArea.insideHeight
    shpChart.Chart.PlotArea.insideHeight = dHeight + 100
    Set shpChart = Nothing
End Sub
```

Die Eigenschaft *Rotation* ruft oder setzt die Drehung der 3D-Chartansicht (die Drehung des Handlungsbereichs um die z-Achse) in Grad. Der Rotationswert muss zwischen 0 bis 360 sein, mit Ausnahme von 3D-Barkarten, bei denen der Wert von 0 bis 44 sein muss. Der Standardwert ist 20. Die folgende Prozedur setzt die Drehung des Diagramms auf 45 Grad (angewendet auf das Anfangs-3D-Diagramm in Bild 2) (Bild 19).

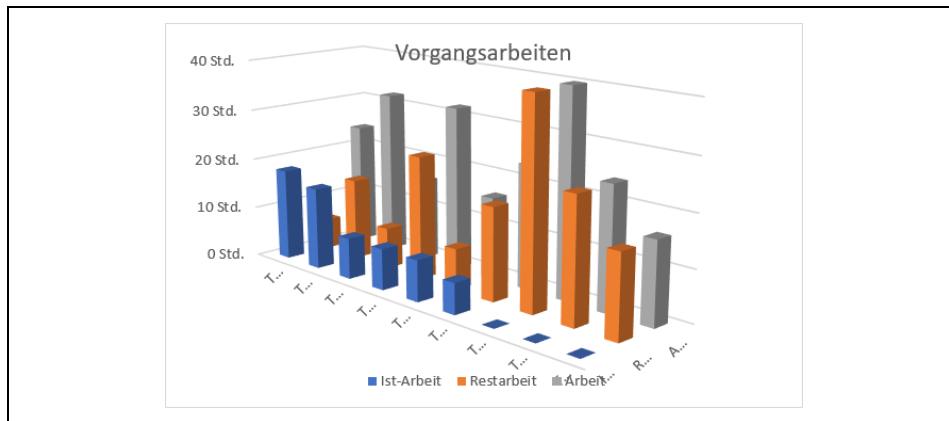


Bild 19. Gedrehte Darstellung

Codeliste 22. Die Prozedur dreht eine 3D-Darstellung

```
Sub SetRotation()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.Rotation = 45
    Set shpChart = Nothing
End Sub
```

Die Eigenschaft *SideWall* ruft ein Office.IMsoWalls-Objekt auf, mit dem der Benutzer die Seitenwand eines 3D-Diagramms einzeln formatieren kann. Die folgende Prozedur färbt die Seitenwand des 3D-Diagramms blau. Im Projekt ist Rot das letzte Stück eines Hexadezimalwertes (Bild 20).

Codeliste 23. Die Prozedur färbt die Seitenwand eines 3D-Diagramms

```
Sub FormatSideWall()
    Dim shpChart As Shape
    Dim sName As String

    sName = "Vorgangsarbeiten"
    Set shpChart = ActiveProject.Reports(sName).Shapes(1)
    shpChart.Chart.SideWall.Fill.ForeColor.RGB = &HFF0000
    Set shpChart = Nothing
End Sub
```

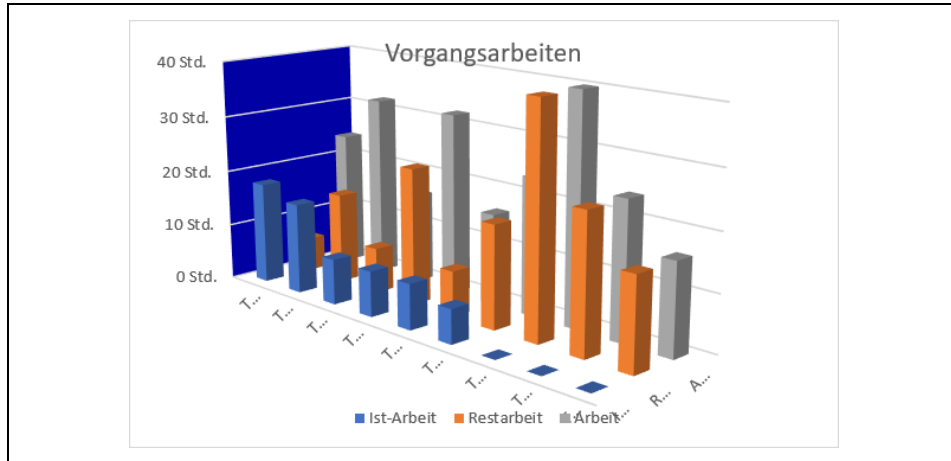


Bild 20. Gefärbte Seitenwand