**' касательная из точки**

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Dim i As Object

Dim iDynamicArray As Kompas6API5.DynamicArray ' Интерфейс ksDynamicArray

' Создать интерфейс динамического массива математических точек

iDynamicArray = iKompasObject.GetDynamicArray(POINT\_ARR)

Dim iMathPointParam As Kompas6API5.MathPointParam ' Интерфейс ksMathPointParam

' Создать интерфейс параметров математической точки

iMathPointParam = iKompasObject.GetParamStruct(Kompas6Constants.StructType2DEnum.ko\_MathPointParam)

Dim P1 As PointDouble = New PointDouble(40, 80)

' Интерфейсы созданы

If (Not iDynamicArray Is Nothing) And (Not iMathPointParam Is Nothing) Then

iDocument2D.ksPoint(P1.X, P1.Y, 3) ' Отрисовка точки

iDocument2D.ksCircle(50, 10, 40, 1) ' Отрисовка окружности

' Получить точки касания окружности и прямой, проходящей через заданную точку

' Координаты внешней точки (10, 50), Координаты центра (50, 10),

' радиус окружности 40

iMathematic2D.ksTanLinePointCircle(P1.X, P1.Y, 50, 10, 40, iDynamicArray)

DrawPointByArray(iDynamicArray) ' Отрисовка точек пересечения

' Отрисовка касательных

For i = 0 To iDynamicArray.ksGetArrayCount - 1

iDynamicArray.ksGetArrayItem(i, iMathPointParam) ' Параметры текущей точки

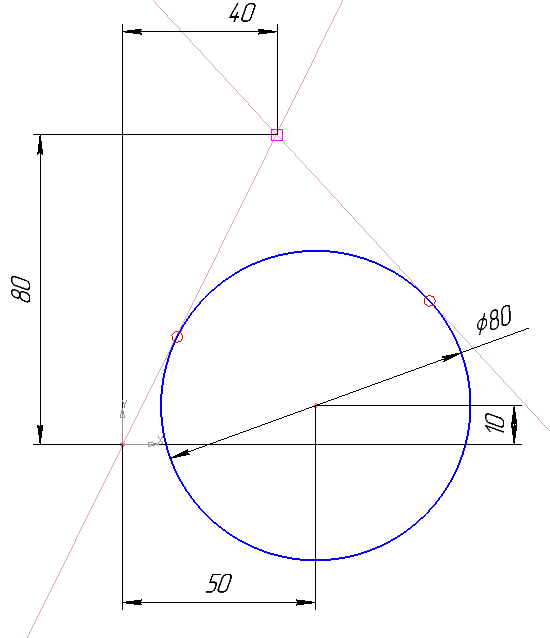
iDocument2D.ksLine(P1.X, P1.Y, iMathematic2D.ksAngle(P1.X, P1.Y, iMathPointParam.x, MathPointParam.y))

Next

iDynamicArray.ksDeleteArray() ' Удаление массива

End If

End Sub



**' касательная под углом**

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

Dim i As Object

Dim iDynamicArray As Kompas6API5.DynamicArray ' Интерфейс ksDynamicArray

' Создать интерфейс динамического массива математических точек

iDynamicArray = iKompasObject.GetDynamicArray(POINT\_ARR)

Dim iMathPointParam As Kompas6API5.MathPointParam ' Интерфейс ksMathPointParam

' Создать интерфейс параметров математической точки

iMathPointParam = iKompasObject.GetParamStruct(Kompas6Constants.StructType2DEnum.ko\_MathPointParam)

' Угол касательной прямой 30

Dim angle As Double = 30

' Интерфейсы созданы

If (Not iDynamicArray Is Nothing) And (Not iMathPointParam Is Nothing) Then

'iDocument2D.ksLineSeg(0, 40, 100, 40, 1) ' Отрисовка отрезка

iDocument2D.ksCircle(50, 10, 40, 1) ' Отрисовка окружности

' Получить точки касания окружности и прямой, проходящей под заданным углом

' Координаты центра (50, 10), радиус окружности 40

iMathematic2D.ksTanLineAngCircle(50, 10, 40, angle, iDynamicArray)

DrawPointByArray(iDynamicArray) ' Отрисовка точек пересечения

' Отрисовка касательных

For i = 0 To iDynamicArray.ksGetArrayCount - 1

iDynamicArray.ksGetArrayItem(i, iMathPointParam) ' Параметры текущей точки

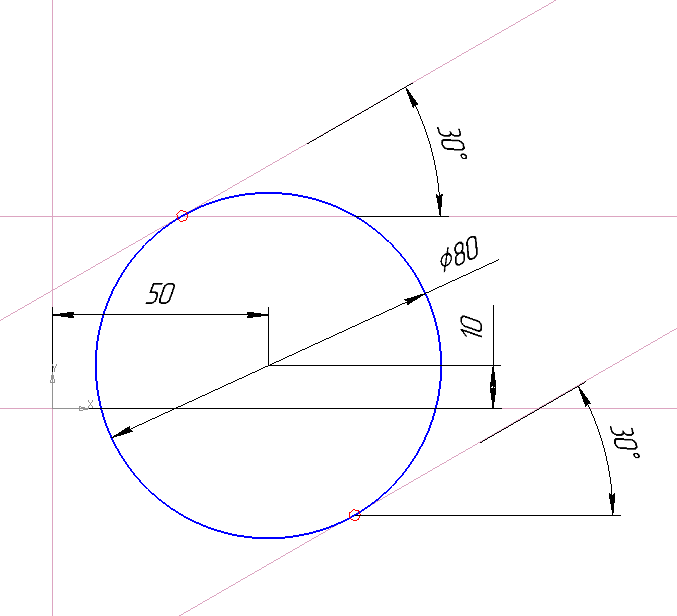
iDocument2D.ksLine(iMathPointParam.x, iMathPointParam.y, angle)

Next

iDynamicArray.ksDeleteArray() ' Удаление массива

End If

End Sub



**' сопрягающие окружности к двум прямым**

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

Dim i As Object

Dim iCON As Kompas6API5.CON ' Интерфейс ksCON

' Создать интерфейс массива координат точек сопряжения

iCON = iKompasObject.GetParamStruct(Kompas6Constants.StructType2DEnum.ko\_CON)

If Not iCON Is Nothing Then ' Интерфейс создан

' Отрисовка прямых

iDocument2D.ksLine(50, 70, 45) ' Первая прямая

iDocument2D.ksLine(50, 70, -45) ' Вторая прямая

' Получить параметры окружностей, касательной к двум прямым

' Радиус сопряжения 20

iMathematic2D.ksCouplingLineLine(50, 70, 45, 50, 70, -45, 20, iCON)

' Отрисовка сопрягающихся окружностей и точек касания

For i = 0 To iCON.GetCount

'iDocument2D.ksCircle(iCON.GetXc(i), iCON.GetYc(i), 20, 2)

iDocument2D.ksPoint(iCON.GetX1(i), iCON.GetY1(i), i)

iDocument2D.ksPoint(iCON.GetX2(i), iCON.GetY2(i), i)

ActiveDoc2D.ksArcByPoint(iCON.GetXc(i), iCON.GetYc(i), 20,

iCON.GetX1(i), iCON.GetY1(i),

iCON.GetX2(i), iCON.GetY2(i), 1, 1) ' Отрисовка дуги по центру и конечным точкам

Next

End If

End Sub

