入門編　2-B、2D図形-プロパティープﾛｸﾞﾗﾑ一覧

(A)共通プロパティ

適用オブジェクト: Applicationオブジェクト ドキュメント(図面) / 画層 / 線（線種・線種尺度・線の太さ） / オブジェクト（TrueColor） / マテリアル（Material） / オブジェクト（ハンドル・オブジェクト ID・AutoCAD クラス名・オブジェクト ID・表示/非表示） / 2D図形オブジェクト（法線単位ベクトル・高度・押し出し距離）

1.Application プロパティ（取得）
- 説明: Application オブジェクトを取得
- 指定: 適用なし
- 取得: '***Objのアプリケーションを返す / Dim myApp As AcadApplication / Dim lineObj As AcadLine / Set myApp =lineObj.Application / MsgBox myApp.name

2.Document プロパティ（取得）
- 説明: オブジェクトが属しているドキュメント(図面)を取得
- 指定: 適用無し
- 取得: '***Objのドキュメント名を取得 / Dim currDoc As AcadDocument / Set currDoc = ***Obj.Document

3.Layer プロパティ（画層・指定、取得）
- 説明: オブジェクトの画層の名前を指定、取得 / 参考：ActiveLayer プロパティ：現在のアクティブな画層を設定または取得を行います。
- 指定: 'レイヤーの宣言 / Dim layerObj As AcadLayer / 'レイヤー"ABC"の作成 / Set layerObj = ThisDrawing.Layers.Add("ABC") / '***ObjのレイヤーをABCに指定 / ***Obj.Layer = "ABC"
- 取得: 'レイヤーの宣言 / Dim currLayer As String / '***Objの線種を取得 / currLayer = ***Obj.Layer

4.Linetype プロパティ（線種・指定、取得）
- 説明: オブジェクトの線種を指定 / 参考：ActiveLinetype プロパティ：現在のアクティブな線種の設定または取得を行います。
- 指定: '***Objの線種を指定 / ***Obj.Linetype = "DASHDOT"
- 取得: '線種の宣言 / Dim currLType As String / '***Objの線種を取得 / currLType = ***Obj.Linetype

5.LinetypeScale プロパティ（線種尺度・指定、取得）
- 説明: オブジェクトの線種尺度を指定、取得 / 説明：オブジェクトの線種尺度は、作図単位に対するダッシュ ドット パターンの相対的な長さを指定します。
- 指定: '***Objの線種尺度を指定 / ***Obj.LinetypeScale = 0.5　'正の実数、 既定値は 1.0
- 取得: '線種尺度の宣言 / Dim currLTScale As Double / '***Objの線種尺度を取得 / currLTScale = ***Obj.LinetypeScale

6.Lineweight プロパティ（線の太さ・指定、取得）
- 説明: 個々のオブジェクトの線の太さ、または図面の既定の線の太さを指定、取得（acLineWeight 列挙型: acLnWt000 … acLnWt211 [illegible]）
- 指定: '***Objの線の太さを指定 / ***Obj.Lineweight = acLnWt211　'acLineWeight 列挙型を指定する
- 取得: '線の太さの宣言 / Dim LTWeight As Double / '線の太さを取得 / LTWeight =***Obj.Lineweight

7.TrueColor プロパティ（指定、取得）
- 説明: オブジェクトの True Color を指定 / 説明：色は、AcCmColor オブジェクトによって識別されます。このオブジェクトは、RGB 値、ACI 番号(1 から 255 の整数)、または色名を保持することができます。RGB 値を使用すると、100 万色の中から選択することができます。
- 指定: '色オブジェクトの宣言 / Dim col As New AcadAcCmColor　'カラーオブジェクト / 'True Color の RGB 値を指定 / Call col.SetRGB (125, 175, 235)　'カラーオブジェクトに色を設定 / '***ObjのTrueColor を設定 / ***Obj.TrueColor = col
- 取得: '色オブジェクトの宣言 / Dim retcolor As AcadAcCmColor / '線のTrueColor を取得 / Set retcolor = line.TrueColor / '色をRGBで表示 / MsgBox "Red = " & retcolor.Red & vbLf & _ "Green = " & retcolor.Green & vbLf & _ "Blue = " & retcolor.Blue

8.Material プロパティ（指定、取得）
- 説明: マテリアルの名前を指定
- 指定: 'マテリアルコレクション、オブジェクトの宣言 / Dim oMaterials As AcadMaterials / Dim oMaterial As AcadMaterial / 'マテリアルオブジェクト作成 / Set oMaterial = ThisDrawing.Materials.Add("TestMaterial") / 'アクティブマテリアル設定 / ThisDrawing.ActiveMaterial = oMaterial
- 取得: 'マテリアルオブジェクトの宣言 / Dim oMaterial As AcadMaterial / 'アクティブマテリアルを取得 / oMaterial = ThisDrawing.ActiveMaterial / 'アクティブマテリアルを表示 / MsgBox oMaterial.Name

9.Handle プロパティ（ハンドル・取得）
- 説明: オブジェクトのハンドルを取得（16 バイトの文字列で表現される） / 説明1．オブジェクト ID と固有のハンドルは、オブジェクトを参照する手段です。 / 説明2．ハンドルは、オブジェクトが図面中に存在する間は、不変(同一状態を保つ)です。 / 注意：一般に、オブジェクト ID でなくハンドルを使用します。ただし、MDI環境（複数図面オープン）では重複の可能性ありと説明されている。
- 指定: 適用無し
- 取得: 'エンティティの宣言、ObjectIDの宣言 / Dim entry As AcadEntity / Dim entHandle As String / 'モデル空間コレクションを反復処理し、見つかった各オブジェクトのハンドルを表示 / For Each entry In ThisDrawing.ModelSpace　'モデル空間内を反復処理 / entHandle = entry.Handle / MsgBox "ハンドルは " & entHandle / Next

10.ObjectId プロパティ（オブジェクト ID・取得）
- 説明: オブジェクト ID を取得 (ObjectIdクラスで表現される) / 説明1.オブジェクト ID と固有のハンドルは、オブジェクトを参照する手段です。 / 説明2.オブジェクトIDはセッション毎に更新され図面には保存されない。
- 指定: 適用無し
- 取得: 'エンティティの宣言、ObjectIDの宣言 / Dim entry As AcadEntity / Dim entObjectID As Long / 'モデル空間を反復処理し、見つかった各オブジェクトのObjectIDを表示 / For Each entry In ThisDrawing.ModelSpace　'モデル空間内を反復処理 / entObjectID = entry.ObjectID / MsgBox " ObjectID は " & entObjectID / Next

11.ObjectName プロパティ（AutoCAD クラス名・指定、取得）
- 説明: オブジェクトの AutoCAD クラス名を取得　オブジェクト名はAcDb＊＊＊となっています。
- 指定: '線オブジェクトだけを削除する例 / Dim object As AcadObject / For Each object In ThisDrawing.ModelSpace / If object.ObjectName = "AcDbLine" Then / object.Delete / End If / Next / ThisDrawing.Regen (acActiveViewport)
- 取得: 'エンティティの宣言、objNameの宣言 / Dim entry As AcadEntity / Dim objName As String / 'モデル空間を反復処理し、見つかった各オブジェクトのObjectNameを表示 / For Each entry In ThisDrawing.ModelSpace　'モデル空間内を反復処理 / objName = entry.ObjectName / MsgBox "オブジェクト名は " & objName / Next

12.OwnerID プロパティ（オブジェクト ID・取得）
- 説明: オーナー(親)オブジェクトのオブジェクト ID を取得
- 指定: 適用無し
- 取得: MsgBox" オブジェクトオーナーIDは:" & ＊＊＊Obj.OwnerID

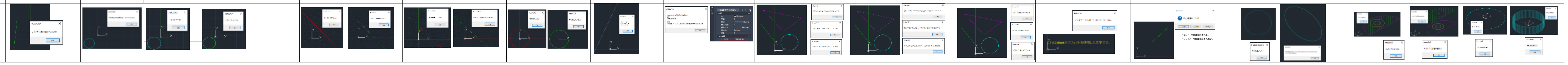
14.Visible プロパティ（表示/非表示・指定、取得）
- 説明: オブジェクトまたはアプリケーションの表示/非表示を指定、取得 / ブール型 / True: オブジェクトまたはアプリケーションを表示 / False: オブジェクトまたはアプリケーションを非表示
- 指定: 'オブジェクト表示 / ***Obj.Visible = True / 'オブジェクト非表示 / ***Obj.Visible = False
- 取得: 'オブジェクト表示、非表示の宣言 / Dim TFVisible As Boolean / TFVisible = ***Obj.Visible

16.Normal プロパティ（法線単位ベクトル・指定、取得）
- 説明: オブジェクトの 3DWCS 法線単位ベクトルを指定、取得 / 注意1、この法線ベクトルは、指定されたオブジェクトの Z 軸を定義します。(X,Y軸ではない) / 注意2、このプロパティが指定するのはベクトルであって、点ではない。（ベクトルは空間内の位置ではなく、法線の方向を定義します。）
- 指定: '3D 法線単位ベクトルの宣言 / Dim newNormal(0 To 2) As Double / '3D 法線単位ベクトルの設定 / newNormal(0) = 1: newNormal(1) = 1: newNormal(2) = -1 / '法線方向の指定 / ***Obj.Normal = newNormal
- 取得: '法線ベクトルの宣言 / Dim currNormal As Variant / '***Objの法線を取得 / currNormal = ***Obj.Normal

17Elevation プロパティ（高度・指定、取得）
- 説明: ハッチング、ポリライン、断面の現在の高度を指定 / 高度：3D 点が必要なのに X と Y の値しか指定されていないときに使用される Z 値のこと。(ElevationModelSpace プロパティを参照のこと)
- 指定: '***Objの高度を指定 / ***Obj.Elevation =5
- 取得: '高度を宣言 / Dim currElevation As Double / '***Objの高度を取得 / currElevation = ***Obj.Elevation

18.Thickness プロパティ（押し出し距離・指定、取得）
- 説明: 2D AutoCAD オブジェクトがその高さの上か下に押し出される距離を指定（押し出し距離、既定値 = 0.0） / 注意：オブジェクトの押し出し距離（厚さ）を変えると、Z 方向の厚さが変わります。(X,Y方向ではありません)
- 指定: '***Objの押し出し距離を指定 / ***Obj.Thickness= 10
- 取得: '押し出し距離を宣言 / Dim currThickness As Double / '***Objの押し出し距離を取得 / currThickness =***Obj.Thickness

例　実行例



(B)指定プロパティ

1、StartPoint プロパティ（Arc、Ellipse、Line）
（Arc、Ellipse、Line のプロパティだが 指定は Lineのみ可能）
- 説明: 線の始点を指定 / Line オブジェクトのみ指定可能（Arc、Ellipseは指定できない）
- 指定: lineObjは作成済みとする。 / '線の始点座標を宣言 / Dim startPoint(0 To 2) As Double　'線分の始点（3D WCS 座標） / '線の始点座標を設定 / startPoint(0) = 0: startPoint(1) = 0: startPoint(2) = 0 / '線の始点座標を指定 / lineObj.startPoint = startPoint
- 取得: '線の始点座標を宣言　取得の場合はバリアントであることに注意！！！ / Dim startPoint As Variant / '線の始点座標を取得 / startPoint= lineObj.startPoint / '線の始点座標を表示 / MsgBox "線の始点座標は " & _ startPoint(0) & "," & startPoint(1) & "," & startPoint(2)

2、EndPoint プロパティ（Arc、Ellipse、Line）
（Arc、Ellipse、Line のプロパティだが 指定は Lineのみ可能）
- 説明: 線の終点を指定 / Line オブジェクトのみ指定可能（Arc、Ellipseは指定できない）
- 指定: lineObjは作成済みとする。 / '線の終点座標を宣言 / Dim endPoint(0 To 2) As Double　'線分の終点（3D WCS 座標） / '線の終点座標を設定 / endPoint(0) = 0: endPoint(1) = 0: endPoint(2) = 0 / '線の終点座標を指定 / lineObj.endPoint = endPoint
- 取得: '線の終点座標を宣言　取得の場合はバリアントであることに注意！！！ / Dim endPoint As Variant / '線の終点座標を取得 / endPoint= lineObj.endPoint / '線の終点座標を表示 / MsgBox "線の終点座標は " & _ endPoint(0) & "," & endPoint(1) & "," & endPoint(2)

3.Center プロパティ（Arc、Circle、Ellipse）
- 説明: 円弧、円、楕円の中心を指定
- 指定: circObjは作成済みとする。 / '中心座標を宣言 / Dim newCenterPt(0 To 2) As Double / '中心座標を設定 / newCenterPt(0) = 25: newCenterPt(1) = 25: newCenterPt(2) = 0 / '中心座標を指定 / circObj.center = newCenterPt
- 取得: '円の中心を宣言　取得の場合は バリアントであることに注意！！！ / Dim centerPoint As Variant / '中心位置を取得 / centerPoint = circObj.center / '中心位置を表示 / MsgBox "円の中心の座標は " & _ centerPoint(0) & "," & centerPoint(1) & "," & centerPoint(2)

6.Radius プロパティ（Arc、Circle）
- 説明: 円弧、円の半径を指定
- 指定: Arc、Circleは作成済みとする。 / ***Obj.Radius = 5#　'＃はDOUBLE型を示します。
- 取得: '円弧半径を宣言 / Dim arcradius as Double / '円弧の半径を取得 / arcradius = arcObj.radius / '円弧の半径を表示 / MsgBox "円弧の半径は" & arcradius

4.Coordinate プロパティ（LWPolyline、Polyline）
- 説明: オブジェクト（LWPolyline、Polyline）の単一の頂点の座標を指定
- 指定: LWPolyline、Polylineは作成済みとする。 / 'インデックス位置Nの頂点座標を10に変更 / ＊＊＊Obj.Coordinate(N-1) = 10 / index　頂点配列のインデックス（頂点の配列は 0 で始まるから (N-1) となる）
- 取得: 'インデックス位置の頂点座標を宣言　バリアントであることに注意！！！ / Dim coord As Variant / '最初のインデックス位置の頂点の座標を取得 / coord = plineObj.Coordinate(0) / '最初のインデックス位置の頂点の座標を表示 / MsgBox "ポリラインの最初のインデックス位置の座標は、:" & _ coord(0) & "," & coord(1) & "," & coord(2)

5.Coordinates プロパティ（LWPolyline、Polyline）
- 説明: オブジェクト（LWPolyline、Polyline）の各頂点の座標を指定
- 指定: LWPolyline、Polylineは作成済みとする。 / Dim points(5) As Double / points(3) = 5 / points(4) = 5 / points(5) = 0 / ＊＊＊Obj.Coordinates = points　'2 番目（配列No.は3,4,5）の頂点座標(X,Y,Z)を指定
- 取得: 'ポリラインのすべての頂点座標を宣言　バリアントであることに注意！！！ / Dim retCoord As Variant

オブジェクト	バリアント
LightweightPolyline	OCS の 2D 点の配列
Polyline	3D 点の配列(X と Y 座標は OCS で、Z 座標は無視)

'ポリラインのすべての頂点座標を取得 / retCoord = plineObj.Coordinates / '2番目の頂点の座標を表示 / MsgBox "2番目の頂点の座標は次のとおりです。:" & _ points(3) & "," & points(4) & "," & points(5)

7.StartAngle プロパティ（Arc、Ellipse）
- 説明: 円弧または楕円の開始角度（ラジアン表示）を指定
- 指定: Arc、Ellipseは作成済みとする。 / '開始角度45度、終了角度270度に指定（注.ラジアン表示である） / ***Obj.startAngle = 45 * (3.14 / 180) / ***Obj.endAngle = 270 * (3.14 / 180)
- 取得: '***Objの開始角度、終了角度を取得 / MsgBox "***Objの開始角度は" & ***Obj.startAngle * (180 / 3.14) & _ " ***Objの終了角度は" & ***Obj.endAngle * (180 / 3.14)

8.EndAngle プロパティ（Arc、Ellipse）
- 説明: 円弧または楕円の終了角度（ラジアン表示）を指定
- 指定: Arc、Ellipseは作成済みとする。 / '開始角度45度、終了角度270度に指定（注.ラジアン表示である） / ***Obj.startAngle = 45 * (3.14 / 180) / ***Obj.endAngle = 270 * (3.14 / 180)
- 取得: '***Objの開始角度、終了角度を取得 / MsgBox "***Objの開始角度は" & ***Obj.startAngle * (180 / 3.14) & _ " ***Objの終了角度は" & ***Obj.endAngle * (180 / 3.14)
- 例: "左の例に含まれるので省略。"

9.DirectionVector プロパティ（Ray、XLine）
- 説明: 放射線、構築線の方向ベクトルを指定
- 指定: Ray、Xlineは作成済みとする。 / '方向ベクトルの宣言 / Dim newDirectionVec(0 To 2) As Double / '方向ベクトルを設定 / newDirectionVec(0) = 3#: newDirectionVec(1) = 1#: newDirectionVec(2) = 0# / '放射線の方向ベクトルを指定 / rayObj.DirectionVector = newDirectionVec
- 取得: '方向ベクトルの宣言　バリアントであることに注意！！！ / Dim retDir As Variant / '放射線の新しい方向ベクトルを取得 / retDir = rayObj.DirectionVector / ThisDrawing.Regen True / MsgBox "放射線の新しい方向ベクトルは：" & Example"_ rayObj.DirectionVector(0) & "," & rayObj.DirectionVector(1) & "," & rayObj.DirectionVector(2)

10.Closed プロパティ（LWPolyline、Polyline、Spline）
- 説明: 3D ポリライン、ライトウェイト ポリライン、ポリライン、スプラインの開閉を指定 / True: 閉じたオブジェクト / False: 開いたオブジェクト (既定)
- 指定: オブジェクトは作成済みとする。 / Dim plineObj As AcadPolyline / plineObj.Closed = True　'Close Polyline / OR / plineObj.Closed = False　'Open Polyline
- 取得: 'Closed状態を示す文字変数 / Dim closedState As String / 'Closedプロパティを取得 / closedState = IIf(plineObj.Closed, "Closed", "Open")　'IIf(expr,truepart,falsepart)構文 / 'Closed状態を表示 / MsgBox "ポリラインのClosed状態は:" & closedState

11.ConstantWidth プロパティ（LWPolyline、Polyline）
- 説明: ポリラインのすべてのセグメントに対する全体的な幅を指定 / 注意：このプロパティは、すべてのセグメントの開始幅と終了幅を同一の値に設定します。 / 参考：個々のセグメントの幅を指定するにはSetWidth および GetWidth メソッドを使用します。
- 指定: オブジェクトは作成済みとする。 / Dim plineObj As AcadLWPolyline / 'すべてのセグメントの幅を同一にします / plineObj.ConstantWidth = 2.5
- 取得: 'セグメント幅の宣言 / Dim CWidth As Double / 'セグメント幅を取得 / CWidth = plineObj.ConstantWidth

12、Origin プロパティ（Block、Hatch、UCS）
- 説明: UCS座標の原点を指定
- 指定: オブジェクトは作成済みとする。 / '原点配列の宣言 / Dim origin(0 To 2) As Double / '原点配列の設定 / origin(0) = 2: origin(1) = 2: origin(2) = 0 / 'オブジェクトの原点を指定 / ***Obj.origin = origin
- 取得: 'モデル空間に図形オブジェクト***Objが作成されているとする / Dim origin(0 To 2) As Double　'原点配列の宣言 / origin = ***Obj.origin　'オブジェクトの原点を取得 / MsgBox "原点は:" & ***Obj.origin(0) & "," & ***Obj.origin(1) & "," & ***Obj.origin(2) "

13.PatternName プロパティ（Hatch）
- 説明: ハッチング パターンの名前を指定
- 指定: オブジェクトは作成済みとする。 / Dim hatchObj As AcadHatch / 'ハッチングのパターン名を指定 / patternName = "ANSI31"
- 取得: オブジェクトは作成済みとする。 / Dim hatchObj As AcadHatch / 'ハッチングのパターン名を宣言 / Dim patternName As String / 'ハッチングパターンの名前を取得 / patternName = hatchObj.patternName

14.PatternAngle プロパティ（Hatch）
- 説明: ハッチング パターンの角度を指定（ラジアンで表した角度有効範囲は 0～2π）
- 指定: オブジェクトは作成済みとする。 / Dim hatchObj As AcadHatch / 'ハッチングパターンの角度を指定 / hatchObj.patternAngle = 5.25
- 取得: オブジェクトは作成済みとする。 / Dim hatchObj As AcadHatch / 'ハッチングパターンの角度を宣言 / Dim patternAngle As Double / 'ハッチングパターンの角度を取得 / patternAngle = hatchObj.patternAngle

15.PatternScale プロパティ（Hatch）
- 説明: ハッチング パターンの尺度を指定（0 以外の値）
- 指定: オブジェクトは作成済みとする。 / Dim hatchObj As AcadHatch / 'ハッチングパターンのスケールを指定 / hatchObj.PatternScale = 1.5
- 取得: オブジェクトは作成済みとする。 / Dim hatchObj As AcadHatch / 'ハッチングパターンのスケールを宣言 / Dim PatternScale As Double / 'ハッチングパターンのスケールを取得 / PatternScale = hatchObj.PatternScale

例　実行例

(C)取得プロパティ

1、Area プロパティ（平面上で閉じた領域）
- 説明: 面積取得（円弧、円、楕円、ハッチング、ライトウェイト ポリライン、ポリライン、リージョン 平面上で閉じたスプラインの囲まれた面積を取得） / 注意：マルチラインを閉じて面積を求めることは出来ない。
- 宣言: '面積の宣言 / Dim plineArea As Double
- 取得: 'ポリラインの面積を取得 / plineArea = plineObj.Area / '（ポリラインオブジェクトを作成する）
- 表示: 'ポリラインの面積を表示 / MsgBox "ポリラインの面積は:" & plineArea

2、Length プロパティ（Line、3DPolyline）
- 説明: 長さ取得（オブジェクトの長さを取得） / 注意：直線が 3Dであっても長さを取得することができる。
- 構文: '線オブジェクトを宣言 / Dim lineObj As AcadLine / '（線オブジェクトを作成する） / '線の長さを取得 / MsgBox "線の長さは:" & lineObj.length

3、Angle プロパティ（Line）
- 説明: 角度取得（線分の角度を取得 ラジアン単位） / 注意　角度は X 軸から反時計回りに計測します。
- 構文: '線オブジェクトを宣言 / Dim lineObj As AcadLine / '（線オブジェクトを作成する） / '線の角度を取得 / MsgBox "線の角度は:" & lineObj.angle
- 例: 2.3562＊（180/3.1415）=135.00度

4、Delta プロパティ（Line）
- 説明: 線分のデルタを取得（線分の deltaX、deltaY、DeltaZ の値を表す 3 要素の配列を取得します。） / 注意1.終点座標が始点座標よりマイナスであればデルタもマイナス値になる。絶対値（プラス）ではない。 / 注意2.直線が 3Dであっても Deltaを取得することができる。
- 構文: '線のデルタを宣言 / Dim lineDelta As Variant　'バリアントであることに注意！！！ / '線のデルタを取得 / lineDelta = lineObj.Delta / '線のデルタ（増分）の成分（X,Y,Z）を表示 / MsgBox "線のデルタ（増分）は:" & vbCrLf & _ "DeltaX:" & lineDelta(0) & vbCrLf & _ "DeltaY:" & lineDelta(1) & vbCrLf & _ "DeltaZ:" & lineDelta(2)

5、Centroid プロパティ（Region）
- 説明: リージョンの図心（2D 座標）を取得（図心はバリアント型で定義された X、Y 方向の値として返されます。） / 注意.平面図形の場合 線分を閉じた（closed=true)だけでは Centroid は求まらない。必ずRegionにしなければいけない。
- 構文: 'リージョンの重心を宣言 / Dim Centroid As Variant　'バリアントであることに注意！！！ / 'リージョンの重心を取得 / Centroid = boxObj.Centroid / 'リージョンの重心を表示 / MsgBox "リージョンの重心は" & Centroid(0) & "," & Centroid(1)

6、MomentOfInertia プロパティ（Region）
- 説明: リージョンの慣性モーメントを取得（慣性モーメントは、バリアント型で定義された X、Y 方向の値として返されます。） / 注意.平面図形の場合 線分を閉じた（closed=true)だけでは MomentOfInertia は求まらない。必ずRegionにしなければいけない。
- 構文: 'リージョンの慣性モーメントを宣言 / Dim momentOfInertia As Variant　'バリアントであることに注意！！！ / 'リージョンの慣性モーメントを取得 / momentOfInertia = boxObj.momentOfInertia / 'リージョンの慣性モーメントを表示 / MsgBox "リージョンの慣性モーメントは:" & momentOfInertia(0) & "," & momentOfInertia(1)

7、Circumference プロパティ（Circle）
- 説明: 円の円周を取得
- 構文: '円オブジェクトを宣言 / Dim circleObj As AcadCircle / '（円オブジェクトを作成する） / '円オブジェクトの円周を表示 / MsgBox "円の円周は:" & circleObj.Circumference

8、ArcLength プロパティ（Arc）
- 説明: 円弧の長さを取得
- 構文: '円弧オブジェクトを宣言 / Dim arcObj As AcadArc / '（円弧オブジェクトを作成する） / '円弧の長さを取得 / MsgBox "円弧の長さは:" & arcObj.ArcLength

9、TotalAngle プロパティ（Arc）
- 説明: 円弧の全角度を取得（ラジアン）
- 構文: '円弧オブジェクトを宣言 / Dim arcObj As AcadArc / '（円弧オブジェクトを作成する） / '円弧の全角度を取得 / MsgBox "円弧の全角度は:" & arcObj.TotalAngle
- 例: 円弧の角度　230-10=220 度 / 求めたラジアンを度に変換すると / 3.8397＊（180/3.1415）=220 度

10、StartPoint プロパティ（Arc、Ellipse、Line）
（Arc、Ellipse、Line のプロパティだが 指定は Lineのみ可能）
- 説明: 線の始点を指定 / Line オブジェクトのみ指定可能（Arc、Ellipseは指定できない）
- 構文: '線の始点座標を宣言　取得の場合はバリアントであることに注意！！！ / Dim startPoint As Variant / '線の始点座標を取得 / startPoint= lineObj.startPoint / '線の始点座標を表示 / MsgBox "線の始点座標は " & _ startPoint(0) & "," & startPoint(1) & "," & startPoint(2)

11、EndPoint プロパティ（Arc、Ellipse、Line）
（Arc、Ellipse、Line のプロパティだが 指定は Lineのみ可能）
- 説明: 線の終点を指定 / Line オブジェクトのみ指定可能（Arc、Ellipseは指定できない）
- 構文: '線の終点座標を宣言　取得の場合はバリアントであることに注意！！！ / Dim endPoint As Variant / '線の終点座標を取得 / endPoint= lineObj.endPoint / '線の終点座標を表示 / MsgBox "線の終点座標は " & _ endPoint(0) & "," & endPoint(1) & "," & endPoint(2)

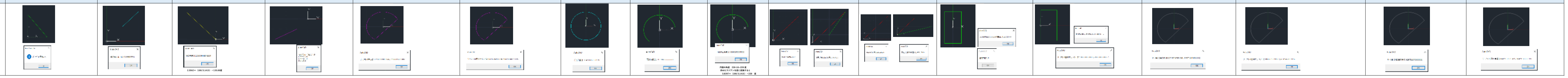
12.PrincipalDirections プロパティ（Region）
- 説明: リージョンの主方向を取得（主方向は バリアント型で定義された X、Y 方向の値として返されます。） / 注.線分を閉じた（closed=true)だけでは Centroid は求まらない、Regionにする必要がある。
- 構文: Dim ＊＊＊Obj As AcadRegion / '（＊＊＊Objを作成する、オブジェクトはリージョン） / '主方向を取得 / MsgBox "主方向は:" & ＊＊＊Obj.PrincipalDirections

13.PrincipalMoments プロパティ（Region）
- 説明: リージョンの主慣性モーメントを取得（主慣性モーメントは、バリアント型で定義された X、Y 方向の値として返されます。） / 注.線分を閉じた（closed=true)だけでは Centroid は求まらない、Regionにする必要がある。
- 構文: Dim ＊＊＊Obj As AcadRegion / '（＊＊＊Objを作成する、オブジェクトはリージョン） / '主慣性モーメントを取得 / MsgBox "主慣性モーメント は:" & ＊＊＊Obj.PrincipalMoments

12.PrincipalDirections プロパティ（Region）
- 説明: リージョンの主方向を取得（主方向は バリアント型で定義された X、Y 方向の値として返されます。） / 注.線分を閉じた（closed=true)だけでは Centroid は求まらない、Regionにする必要がある。
- 構文: Dim ＊＊＊Obj As AcadRegion / '（＊＊＊Objを作成する、オブジェクトはリージョン） / '主方向を取得 / MsgBox "主方向は:" & ＊＊＊Obj.PrincipalDirections

13.PrincipalMoments プロパティ（Region）
- 説明: リージョンの主慣性モーメントを取得（主慣性モーメントは、バリアント型で定義された X、Y 方向の値として返されます。） / 注.線分を閉じた（closed=true)だけでは Centroid は求まらない、Regionにする必要がある。
- 構文: Dim ＊＊＊Obj As AcadRegion / '（＊＊＊Objを作成する、オブジェクトはリージョン） / '主慣性モーメントを取得 / MsgBox "主慣性モーメント は:" & ＊＊＊Obj.PrincipalMoments

14.ProductOfInertia プロパティ（Region）
- 説明: リージョンの慣性乗積を取得（慣性乗積は、バリアント型で定義された X、Y 方向の値として返されます。） / 注.線分を閉じた（closed=true)だけでは Centroid は求まらない、Regionにする必要がある。
- 構文: Dim ＊＊＊Obj As AcadRegion / '（＊＊＊Objを作成する、オブジェクトはリージョン） / '慣性乗積を取得 / MsgBox "慣性乗積 は:" & ＊＊＊Obj.ProductOfInertia

15.RadiiOfGyration プロパティ（Region）
- 説明: リージョンの回転半径を取得（回転半径は、バリアント型で定義された X、Y 方向の値として返されます。） / 注.線分を閉じた（closed=true)だけでは Centroid は求まらない、Regionにする必要がある。
- 構文: Dim ＊＊＊Obj As AcadRegion / '（＊＊＊Objを作成する、オブジェクトはリージョン） / '回転半径を取得 / MsgBox "主慣性モーメント は:" & ＊＊＊Obj.RadiiOfGyration

例　実行例