LIITE A: LUVUN 6.2 CAD PUOLEN KOODI

```
import math
import NXOpen
import NXOpen.Annotations
import NXOpen.Features
import NXOpen.GeometricUtilities
import NXOpen.Preferences

def main() :

    theSession  = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work

    #Sylinteri 1

    cylinderbuilder1 = workPart.Features.CreateCylinderBuilder(NXOpen.Fea-
    tures.Cylinder.Null)

    cylinderbuilder1.Diameter.RightHandSide = "100"
    cylinderbuilder1.Height.RightHandSide = "30.0"
    cylinderbuilder1.Origin = NXOpen.Point3d(60.0, 30.0, 0.0)
    cylinderbuilder1.Direction = NXOpen.Vector3d(0.0, 1.0, 0.0)
    cylinderbuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.Bool-
    eanOperation.BooleanType.Create

    cylinderbuilder1.Commit()
    cylinderbuilder1.Destroy()

    #Sylinteri 2

    cylinderbuilder2 = workPart.Features.CreateCylinderBuilder(NXOpen.Fea-
    tures.Cylinder.Null)

    cylinderbuilder2.Diameter.RightHandSide = "50"
    cylinderbuilder2.Height.RightHandSide = "100"
    cylinderbuilder2.Origin = NXOpen.Point3d(60.0, 60.0 , 0.0)
    cylinderbuilder2.Direction = NXOpen.Vector3d(0.0, 1.0, 0.0)
    cylinderbuilder2.BooleanOption.Type = NXOpen.GeometricUtilities.Bool-
    eanOperation.BooleanType.Create

    cylinderbuilder2.Commit()
    cylinderbuilder2.Destroy()

    #Kartio

    conebuilder = workPart.Features.CreateConeBuilder(NXOpen.Fea-
    tures.Cone.Null)

    conebuilder.BaseDiameter.RightHandSide = "100"
    conebuilder.Height.RightHandSide = "50"

    origin1 = NXOpen.Point3d(60.0, 160.0, 0.0)
    vector1 = NXOpen.Vector3d(0.0, 1.0, 0.0)
    direction1 = workPart.Directions.CreateDirection(origin1, vector1,
    NXOpen.SmartObject.UpdateOption.WithinModeling)

    axis1 = conebuilder.Axis
```

```
point1 = workPart.Points.CreatePoint(origin1)

axis1.Point = point1
axis1.Direction = direction1

conebuilder.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOpe-
ration.BooleanType.Create

conebuilder.Commit()
conebuilder.Destroy()

#Kuutio

blockfeaturebuilder1 = workPart.Features.CreateBlockFeature-
Builder(NXOpen.Features.Block.Null)
blockfeaturebuilder1.Type = NXOpen.Features.BlockFeature-
Builder.Types.OriginAndEdgeLengths

origo = NXOpen.Point3d(0.0, -20.0, -60.0)
blockfeaturebuilder1.SetOriginAndLengths(origo, "120", "50", "120")
blockfeaturebuilder1.BooleanOption.Type = NXOpen.GeometricUtili-
ties.BooleanOperation.BooleanType.Create

blockfeaturebuilder1.Commit()
blockfeaturebuilder1.Destroy()

#Pallo

sphereBuilder1 = workPart.Features.CreateSphereBuilder(NXOpen.Fea-
tures.Sphere.Null)

piste = NXOpen.Point3d(60.0, 250.0, 0.0)
origo = workPart.Points.CreatePoint(piste)
sphereBuilder1.CenterPoint = origo
sphereBuilder1.Diameter.RightHandSide = "80"
sphereBuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.Bool-
eanOperation.BooleanType.Create

sphereBuilder1.Commit()
sphereBuilder1.Destroy()

#Putki

sketchInPlaceBuilder1 = workPart.Sketches.CreateSketchInPlace-
Builder2(NXOpen.Sketch.Null)

origin1 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal1 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane1 = workPart.Planes.CreatePlane(origin1, normal1,
NXOpen.SmartObject.UpdateOption.WithinModeling)

sketchInPlaceBuilder1.PlaneReference = plane1

nXObject1 = sketchInPlaceBuilder1.Commit()

sketch1 = nXObject1
feature1 = sketch1.Feature
```

```
    sketch1.Activate(NXOpen.Sketch.ViewReorient.TrueValue)

    sketchInPlaceBuilder1.Destroy()
    plane1.DestroyPlane()

    #Putken muotoviivan sketchin määrittely

    startPoint1 = NXOpen.Point3d(60.0, -120.0, 0.0)
    endPoint1 = NXOpen.Point3d(60.0, -20.0, 0.0)
    line1 = workPart.Curves.CreateLine(startPoint1, endPoint1)

    theSession.ActiveSketch.AddGeometry(line1, NXOpen.Sketch.InferCon-
    straintsOption.InferNoConstraints)

    theSession.ActiveSketch.Update()

    #Putken muotoviivan määrittely

    tubeBuilder1 = workPart.Features.CreateTubeBuilder(NXOpen.Features.Fea-
    ture.Null)

    tubeBuilder1.Tolerance = 0.01

    tubeBuilder1.OuterDiameter.RightHandSide = "60"
    tubeBuilder1.InnerDiameter.RightHandSide = "40"

    tubeBuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOp-
    eration.BooleanType.Create

    tubeBuilder1.PathSection.DistanceTolerance = 0.01
    tubeBuilder1.PathSection.ChainingTolerance = 0.01

    tubeBuilder1.PathSection.SetAllowedEntityTypes(NXOpen.Section.Allow-
    Types.OnlyCurves)

    features1 = [NXOpen.Features.Feature.Null] * 1
    sketchFeature1 = feature1 #sketchin viiva
    features1[0] = sketchFeature1
    curveFeatureRule1 = workPart.ScRuleFactory.CreateRuleCurveFeature(fea-
    tures1)

    tubeBuilder1.PathSection.AllowSelfIntersection(True)

    rules1 = [None] * 1
    rules1[0] = curveFeatureRule1
    helpPoint1 = NXOpen.Point3d(0.0, 0.0, 0.0)
    tubeBuilder1.PathSection.AddToSection(rules1, NXOpen.NXObject.Null,
    NXOpen.NXObject.Null, NXOpen.NXObject.Null, helpPoint1, NXOpen.Sec-
    tion.Mode.Create, False)

    #Putki määritelty

    nXObject2 = tubeBuilder1.Commit()

    tubeBuilder1.Destroy()

if __name__ == '__main__':
    main()
```

LIITE B: LUVUN 6.2 CAM PUOLEN KOODI

```
import math
import NXOpen
import NXOpen.CAM
import NXOpen.Features
def main() :

    theSession  = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work
    displayPart = theSession.Parts.Display

    #Luodaan pinnan jyrsintä halutulla työkalulla ja geometrialla

    nCGroup1 = workPart.CAMSetup.CAMGroupCollection.FindObject("NC_PRO-
    GRAM")
    method1 = workPart.CAMSetup.CAMGroupCollection.FindObject("METHOD")
    tool1 = workPart.CAMSetup.CAMGroupCollection.FindObject("UGT0212_002")
    orientGeometry1 = workPart.CAMSetup.CAMGroupCollection.FindOb-
    ject("MCS_SPINDLE")
    operation1 = workPart.CAMSetup.CAMOperationCollection.Create(nCGroup1,
    method1, tool1, orientGeometry1, "mill_planar", "FACE_MILLING",
    NXOpen.CAM.OperationCollection.UseDefaultName.TrueValue, "FACE_MILL-
    ING")

    faceMilling1 = operation1
    faceMillingBuilder1 = workPart.CAMSetup.CAMOperationCollection.Create-
    FaceMillingBuilder(faceMilling1)
    faceMillingBuilder1.PartGeometry.InitializeData(False)

    geometrySetList1 = faceMillingBuilder1.PartGeometry.GeometryList

    taggedObject1 = geometrySetList1.FindItem(0)
    geometrySet1 = taggedObject1

    #Määritellään kappale

    partLoadStatus1 = workPart.LoadThisPartFully()

    partLoadStatus1.Dispose()
    bodies1 = [NXOpen.Body.Null] * 1
    body1 = workPart.Bodies.FindObject("EXTRUDE(10)")
    bodies1[0] = body1
    bodyDumbRule1 = workPart.ScRuleFactory.CreateRuleBodyDumb(bodies1,
    True)

    scCollector1 = geometrySet1.ScCollector

    rules1 = [None] * 1
    rules1[0] = bodyDumbRule1
    scCollector1.ReplaceRules(rules1, False)

    nXObject1 = faceMillingBuilder1.Commit()
    faceMillingBuilder1.Destroy()

    faceMilling2 = nXObject1
    faceMillingBuilder2 = workPart.CAMSetup.CAMOperationCollection.Create-
    FaceMillingBuilder(faceMilling2)
```

```
#Määritellään aihio

origin1 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal1 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane1 = workPart.Planes.CreatePlane(origin1, normal1,
NXOpen.SmartObject.UpdateOption.AfterModeling)

unit1 = workPart.UnitCollection.FindObject("MilliMeter")

boundary1 = faceMillingBuilder2.BlankBoundary
boundarySetList1 = boundary1.BoundaryList

taggedObject2 = boundarySetList1.FindItem(0)

boundaryMillingSet1 = taggedObject2
origin2 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal2 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane2 = workPart.Planes.CreatePlane(origin2, normal2,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundaryMillingSet1.Plane = plane2

faces1 = [NXOpen.Face.Null] * 1
extrude1 = workPart.Features.FindObject("EXTRUDE(10)")
face1 = extrude1.FindObject("FACE 130 {(40,0,0) EXTRUDE(10)}")
faces1[0] = face1
faceDumbRule1 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces1)

scCollector2 = workPart.FindObject("ENTITY 113 1")
rules2 = [None] * 1
rules2[0] = faceDumbRule1
scCollector2.ReplaceRules(rules2, False)

bndsets1 = boundary1.AppendFaceBoundary(face1, True, False, False,
NXOpen.CAM.BoundarySet.ToolSideTypes.InsideOrLeft)

taggedObject3 = boundaryMillingSet1.BoundaryMemberList.FindItem(0)

boundaryMillingMemberSet1 = taggedObject3
postEventsCiBuilder1 = boundaryMillingMemberSet1.StartEventsBuilder
postEventsCiBuilder2 = boundaryMillingMemberSet1.EndEventsBuilder

plane1.DestroyPlane()
nXObject2 = faceMillingBuilder2.Commit()
faceMillingBuilder2.Destroy()

faceMilling3 = nXObject2
faceMillingBuilder3 = workPart.CAMSetup.CAMOperationCollection.Create-
FaceMillingBuilder(faceMilling3)

#Luodaan työstöradat

nXObject3 = faceMillingBuilder3.Commit()

objects1 = [NXOpen.CAM.CAMObject.Null] * 1
faceMilling4 = nXObject3
objects1[0] = faceMilling4
workPart.CAMSetup.GenerateToolPath(objects1)
```

```
faceMillingBuilder3.Destroy()
faceMillingBuilder4 = workPart.CAMSetup.CAMOperationCollection.Create-
FaceMillingBuilder(faceMilling4)
nXObject4 = faceMillingBuilder4.Commit()
faceMillingBuilder4.Destroy()
theSession.CleanUpFacetedFacesAndEdges()

#pinnan jyrsintä suoritettu
#Luodaan tason jyrsintä halutulla työkalulla ja geometrialla

tool2 = workPart.CAMSetup.CAMGroupCollection.FindObject("UGT0201_004")
operation2 = workPart.CAMSetup.CAMOperationCollection.Create(nCGroup1,
method1, tool2, orientGeometry1, "mill_planar", "PLANAR_MILL",
NXOpen.CAM.OperationCollection.UseDefaultName.TrueValue, "PLANAR_MILL")

planarMilling1 = operation2
planarMillingBuilder1 = workPart.CAMSetup.CAMOperationCollection.Cre-
atePlanarMillingBuilder(planarMilling1)

#Määritellään kappale

origin3 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal3 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane3 = workPart.Planes.CreatePlane(origin3, normal3,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundaryPlanarMill1 = planarMillingBuilder1.PartBoundary
boundarySetList2 = boundaryPlanarMill1.BoundaryList

taggedObject4 = boundarySetList2.FindItem(0)
boundarySetPlanarMill1 = taggedObject4
origin4 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal4 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane4 = workPart.Planes.CreatePlane(origin4, normal4,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundarySetPlanarMill1.Plane = plane4

faces2 = [NXOpen.Face.Null] * 1
faces2[0] = face1
faceDumbRule2 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces2)

scCollector3 = workPart.FindObject("ENTITY 113 7")
rules3 = [None] * 1
rules3[0] = faceDumbRule2
scCollector3.ReplaceRules(rules3, False)

plane5 = plane4.CopyPlane()

bndsets2 = boundaryPlanarMill1.AppendFaceBoundary(face1, False, True,
False, NXOpen.CAM.BoundarySet.ToolSideTypes.OutsideOrRight, 1, 1)

boundarySetPlanarMill2 = bndsets2[1]
boundarySetPlanarMill2.Plane = plane5

taggedObject5 = boundarySetPlanarMill1.BoundaryMemberList.FindItem(0)

plane3.DestroyPlane()
```

```
nXObject5 = planarMillingBuilder1.Commit()

planarMillingBuilder1.Destroy()

planarMilling2 = nXObject5
planarMillingBuilder2 = workPart.CAMSetup.CAMOperationCollection.Cre-
atePlanarMillingBuilder(planarMilling2)

#Määritellään aihio

origin5 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal5 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane6 = workPart.Planes.CreatePlane(origin5, normal5,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundaryPlanarMill2 = planarMillingBuilder2.BlankBoundary

boundarySetList3 = boundaryPlanarMill2.BoundaryList

taggedObject6 = boundarySetList3.FindItem(0)

boundarySetPlanarMill3 = taggedObject6
origin6 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal6 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane7 = workPart.Planes.CreatePlane(origin6, normal6,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundarySetPlanarMill3.Plane = plane7

faces3 = [NXOpen.Face.Null] * 1
faces3[0] = face1
faceDumbRule3 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces3)

scCollector4 = workPart.FindObject("ENTITY 113 5")
rules4 = [None] * 1
rules4[0] = faceDumbRule3
scCollector4.ReplaceRules(rules4, False)

bndsets3 = boundaryPlanarMill2.AppendFaceBoundary(face1, True, False,
False, NXOpen.CAM.BoundarySet.ToolSideTypes.InsideOrLeft, 1, 1)
taggedObject7 = boundarySetPlanarMill3.BoundaryMemberList.FindItem(0)
plane6.DestroyPlane()
nXObject6 = planarMillingBuilder2.Commit()
planarMillingBuilder2.Destroy()

planarMilling3 = nXObject6
planarMillingBuilder3 = workPart.CAMSetup.CAMOperationCollection.Cre-
atePlanarMillingBuilder(planarMilling3)

#Määritellään lattia

origin7 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal7 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane8 = workPart.Planes.CreatePlane(origin7, normal7,
NXOpen.SmartObject.UpdateOption.AfterModeling)

plane8.SetUpdateOption(NXOpen.SmartObject.UpdateOption.AfterModeling)
plane8.SetMethod(NXOpen.PlaneTypes.MethodType.Distance)
```

```
geom1 = [NXOpen.NXObject.Null] * 1
extrude2 = workPart.Features.FindObject("EXTRUDE(12)")
face2 = extrude2.FindObject("FACE 130 {(15,0,0) EXTRUDE(10)}")
geom1[0] = face2

plane8.SetGeometry(geom1)
plane8.SetFlip(False)
plane8.SetReverseSide(False)
plane8.SetAlternate(NXOpen.PlaneTypes.AlternateType.One)
plane8.Evaluate()
plane8.RemoveOffsetData()

planarMillingBuilder3.Geometry.FloorPlane = plane8

nXObject7 = planarMillingBuilder3.Commit()

#Luodaan työstöradat

objects2 = [NXOpen.CAM.CAMObject.Null] * 1
planarMilling4 = nXObject7
objects2[0] = planarMilling4
workPart.CAMSetup.GenerateToolPath(objects2)

planarMillingBuilder3.Destroy()

theSession.CleanUpFacetedFacesAndEdges()

#Pinnan jyrsintä suoritettu
#Luodaan ulkopuolen halkaisijan sorvaus halutulla työkalulla ja geometrialla

tool3 = workPart.CAMSetup.CAMGroupCollection.FindObject("UGT0101_011")
featureGeometry1 = workPart.CAMSetup.CAMGroupCollection.FindObject("TURNING_WORKPIECE")
operation3 = workPart.CAMSetup.CAMOperationCollection.Create(nCGroup1,
method1, tool3, featureGeometry1, "turning", "ROUGH_TURN_OD",
NXOpen.CAM.OperationCollection.UseDefaultName.TrueValue,
"ROUGH_TURN_OD")

roughTurning1 = operation3
roughTurningBuilder1 = workPart.CAMSetup.CAMOperationCollection.CreateRoughTurningBuilder(roughTurning1)

#Määritellään työstöalueen rajat

roughTurningBuilder1.TrimPoint1.Option = NXOpen.CAM.TrimPoint.Options.Point

scalar1 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.DimensionalityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)

extractFace1 = workPart.Features.FindObject("SpunOutline(59)")

edge1 = extractFace1.FindObject("EDGE * -2 * 1
{(0,32.4999999999986,0)(20,32.4999999999986,0)(40,32.4999999999986,0)
SpunOutline(59)}")
point1 = workPart.Points.CreatePoint(edge1, scalar1,
NXOpen.SmartObject.UpdateOption.AfterModeling)
```

```
    xform1, nXObject9 = workPart.Xforms.CreateExtractXform(edge1,
    NXOpen.SmartObject.UpdateOption.AfterModeling, False)

    scalar2 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.Dimensional-
    ityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)
    edge2 = nXObject9

    point2 = workPart.Points.CreatePoint(edge2, scalar2,
    NXOpen.SmartObject.UpdateOption.AfterModeling)
    point2.RemoveViewDependency()
    roughTurningBuilder1.TrimPoint1.Point = point2
    point2.RemoveViewDependency()
    roughTurningBuilder1.TrimPoint2.Option = NXOpen.CAM.TrimPoint.Op-
    tions.Point

    scalar3 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.Dimensional-
    ityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)

    edge3 = extractFace1.FindObject("EDGE * -1 * 1
    {(0,0,0)(0,16.2499999999993,0)(0,32.4999999999986,0) SpunOutline(59)}")
     oint3 = workPart.Points.CreatePoint(edge3, scalar3,
    NXOpen.SmartObject.UpdateOption.AfterModeling)

    xform2, nXObject10 = workPart.Xforms.CreateExtractXform(edge3,
    NXOpen.SmartObject.UpdateOption.AfterModeling, False)

    scalar4 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.Dimensional-
    ityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)

    edge4 = nXObject10
    point4 = workPart.Points.CreatePoint(edge4, scalar4,
    NXOpen.SmartObject.UpdateOption.AfterModeling)
    point4.RemoveViewDependency()
    roughTurningBuilder1.TrimPoint2.Point = point4
    point4.RemoveViewDependency()

    workPart.Points.DeletePoint(point1)
    workPart.Points.DeletePoint(point3)

    nXObject11 = roughTurningBuilder1.Commit()

    #Luodaan työstöradat

    objects3 = [NXOpen.CAM.CAMObject.Null] * 1
    roughTurning2 = nXObject11
    objects3[0] = roughTurning2
    workPart.CAMSetup.GenerateToolPath(objects3)

    roughTurningBuilder1.Destroy()
    theSession.CleanUpFacetedFacesAndEdges()

if __name__ == '__main__':
    main()
```

LIITE C: LUVUN 6.2 CMM PUOLEN KOODI

```
import math
import NXOpen
import NXOpen.Assemblies
import NXOpen.CAM
import NXOpen.Positioning
import NXOpen.Preferences
def main() :

    theSession  = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work
    displayPart = theSession.Parts.Display

    inspectionGroup1 = workPart.InspectionSetup.CmmInspectionGroupColle-
    tion.FindObject("GENERIC_MACHINE")
    machineGroupBuilder1 = workPart.InspectionSetup.CmmInspectionGroupCol-
    lection.CreateMachineGroupBuilder(inspectionGroup1)
    ncmctPartMountingBuilder1 = workPart.InspectionSetup.CreateNcmctPart-
    MountingBuilder("zeiss_prismo")
    ncmctPartMountingBuilder1.Positioning = NXOpen.CAM.NcmctPartMount-
    ingBuilder.PositioningTypes.UseAssemblyPositioning
    selectNXObjectList1 = ncmctPartMountingBuilder1.Geometry

    #Lisätään kappale

    selectNXObjectList2 = ncmctPartMountingBuilder1.Geometry
    component1 = workPart.ComponentAssembly.RootComponent.FindObject("COM-
    PONENT cam_tau 1")
    body1 = component1.FindObject("PROTO#.Bodies|EXTRUDE(3)")
    added1 = selectNXObjectList2.Add(body1)
    nXObject1 = ncmctPartMountingBuilder1.Commit()
    machineGroupBuilder1.RemoveMachine()

    #Lisätään mittauskone

    basePoint1 = NXOpen.Point3d(0.0, 0.0, 0.0)
    orientation1 = NXOpen.Matrix3x3()
    orientation1.Xx = 1.0
    orientation1.Xy = 0.0
    orientation1.Xz = 0.0
    orientation1.Yx = 0.0
    orientation1.Yy = 1.0
    orientation1.Yz = 0.0
    orientation1.Zx = 0.0
    orientation1.Zy = 0.0
    orientation1.Zz = 1.0
    component2, partLoadStatus1 = workPart.ComponentAssembly.AddCompo-
    nent("C:\\Apps\\Siemens\\NX12\\cmm_inspection\\resource\\library\\ma-
    chine\\installed_machines\\zeiss_prismo\\graphics\\Zeiss_Prismo.prt",
    "None", "ZEISS_PRISMO", basePoint1, orientation1, -1, True)

    partLoadStatus1.Dispose()
    objects1 = []
    nErrs2 = theSession.UpdateManager.AddObjectsToDeleteList(objects1)
    machineGroupBuilder1.MachinePartOccurrence = component2
    machineGroupBuilder1.UpdateCamSetup(NXOpen.CAM.MachineGroupBuilder.Re-
    trieveToolPocketInformation.Yes, ncmctPartMountingBuilder1)
```

```
ncmctPartMountingBuilder1.Destroy()
machineGroupBuilder1.Destroy()

#Lisätään mittapää ja sen pidike

machineGroupBuilder2 = workPart.InspectionSetup.CmmInspectionGroupCol-
lection.CreateMachineGroupBuilder(inspectionGroup1)
inspectionGroup2 = workPart.InspectionSetup.Re-
trieveDevice("Zeiss_VAST")
inspectionTool1, success1 = workPart.InspectionSetup.Re-
trieveTool("TP20_STD_A-5003-0040-01-A")
nXObject2 = machineGroupBuilder2.Commit()
machineGroupBuilder2.Destroy()

#Perussuuntaus ja nollapiste

componentPositioner1 = workPart.ComponentAssembly.Positioner
componentPositioner1.ClearNetwork()

arrangement1 = workPart.ComponentAssembly.Arrangements.FindObject("Ar-
rangement 1")
componentPositioner1.PrimaryArrangement = arrangement1
componentPositioner1.BeginMoveComponent()
allowInterpartPositioning1 = theSession.Preferences.Assemblies.In-
terpartPositioning
expression1 = workPart.Expressions.CreateSystemExpression-
WithUnits("1.0", NXOpen.Unit.Null)
unit1 = workPart.UnitCollection.FindObject("MilliMeter")
unit2 = workPart.UnitCollection.FindObject("Degrees")
network1 = componentPositioner1.EstablishNetwork()
componentNetwork1 = network1
componentNetwork1.MoveObjectsState = True
componentNetwork1.DisplayComponent = NXOpen.Assemblies.Component.Null
componentNetwork1.NetworkArrangementsMode = NXOpen.Positioning.Compo-
nentNetwork.ArrangementsMode.Existing
componentNetwork1.RemoveAllConstraints()

movableObjects1 = [NXOpen.NXObject.Null] * 1
movableObjects1[0] = component1
componentNetwork1.SetMovingGroup(movableObjects1)
componentNetwork1.Solve()
loaded1 = componentNetwork1.IsReferencedGeometryLoaded()
componentNetwork1.BeginDrag()

translation1 = NXOpen.Vector3d(0.0, 0.0, 0.0)

rotation1 = NXOpen.Matrix3x3()
rotation1.Xx = 0.0
rotation1.Xy = 0.0
rotation1.Xz = -1.0
rotation1.Yx = 0.0
rotation1.Yy = 1.0
rotation1.Yz = 0.0
rotation1.Zx = 1.0
rotation1.Zy = 0.0
rotation1.Zz = 0.0

componentNetwork1.DragByTransform(translation1, rotation1)
```

```
    componentNetwork1.EndDrag()
    componentNetwork1.ResetDisplay()
    componentNetwork1.ApplyToModel()
    componentNetwork1.Solve()
    componentPositioner1.ClearNetwork()
    nErrs3 = theSession.UpdateManager.AddToDeleteList(componentNetwork1)
    componentPositioner1.DeleteNonPersistentConstraints()
    componentPositioner1.EndMoveComponent()
    componentPositioner1.PrimaryArrangement = NXOpen.Assemblies.Arrange-
    ment.Null

    #Liitetään PMI tieto ja luodaan mittausohjelma

    inspectionLinkPmiBuilder1 = workPart.InspectionSetup.CmmInspection-
    OperationCollection.CreateInspectionLinkPmiBuilder(NXOpen.CAM.CAMOb-
    ject.Null)

    inspectionLinkPmiBuilder1.WorkpieceString = "Cam Workpiece"
    inspectionLinkPmiBuilder1.ViewString = "All"
    inspectionLinkPmiBuilder1.ToolString = "Auto"
    inspectionLinkPmiBuilder1.TipString = "Auto"
    inspectionLinkPmiBuilder1.AngleString = "Auto"
    inspectionLinkPmiBuilder1.ProgramLocationString = "INSPECTION_PATHS"
    inspectionLinkPmiBuilder1.ProbeTipType = NXOpen.CAM.InspectionMove-
    Builder.ProbeTipTypes.Any
    inspectionLinkPmiBuilder1.CreatePathsEnum = NXOpen.CAM.Inspec-
    tionLinkPmiBuilder.CreatePathsOptions.Yes

    inspectionLinkPmiBuilder1.LinkToPmi()
    inspectionLinkPmiBuilder1.OutputResults(NXOpen.ListingWindow.Device-
    Type.Window, "")
    inspectionLinkPmiBuilder1.Destroy()

if __name__ == '__main__':
    main()
```