

Москва 2016

# Конфигуратор машиностроительно ГО изделия

Выполнила студентка группы 161-343: Уразаева В.А

Преподаватели: Джунковский А.В.

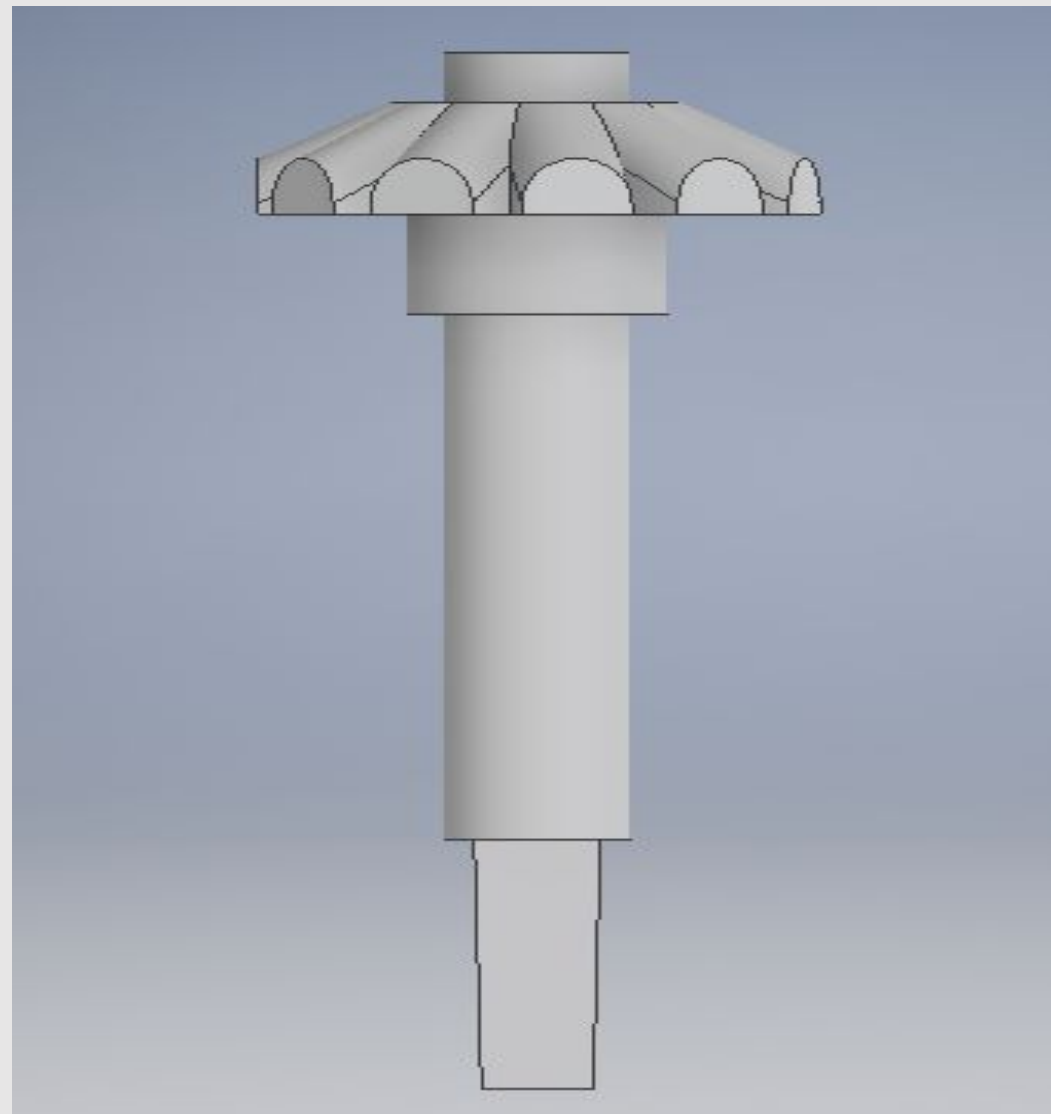
Холодов Д.А.

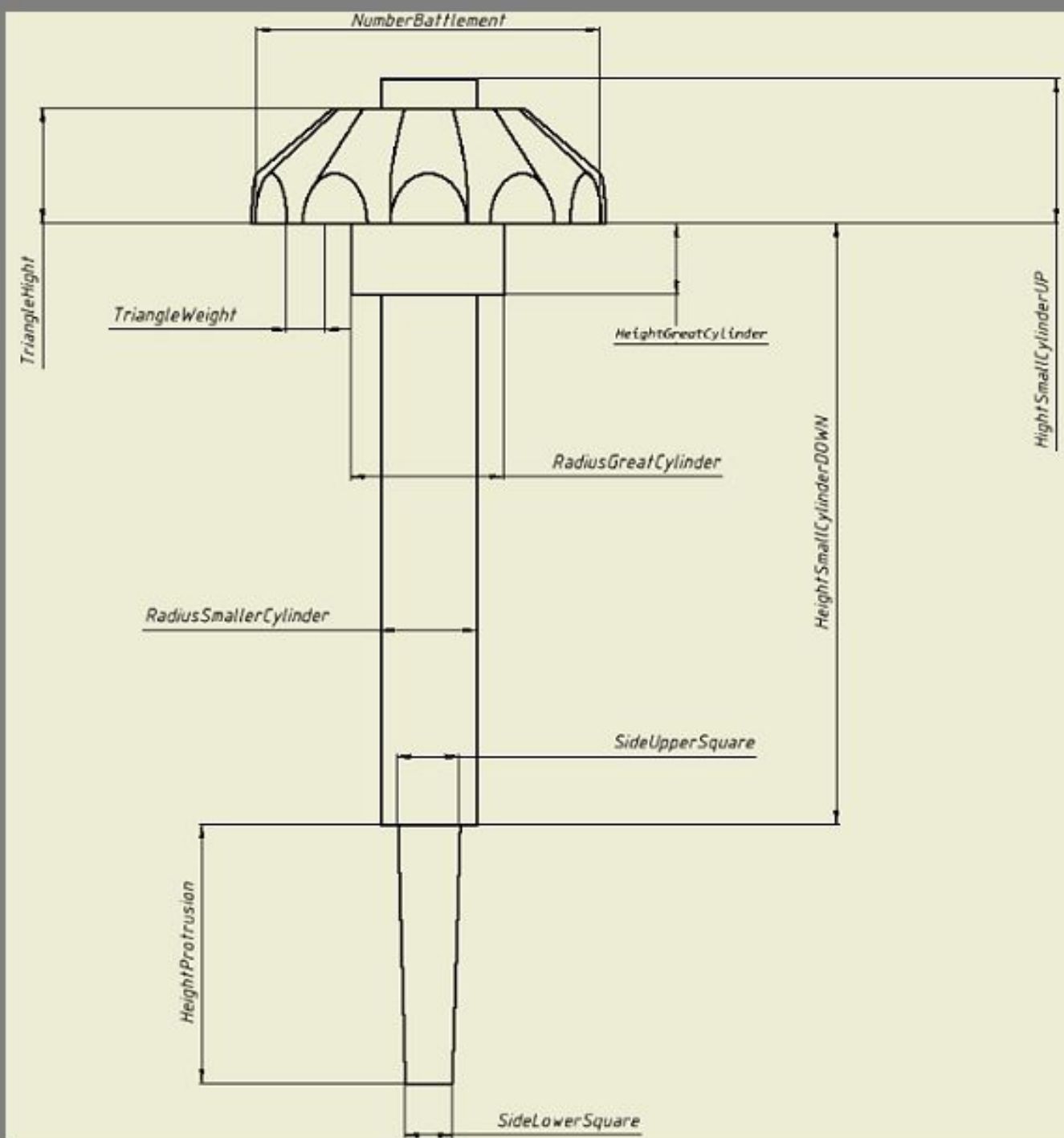
Куратор: Ражев В.В.

# Среда разработки

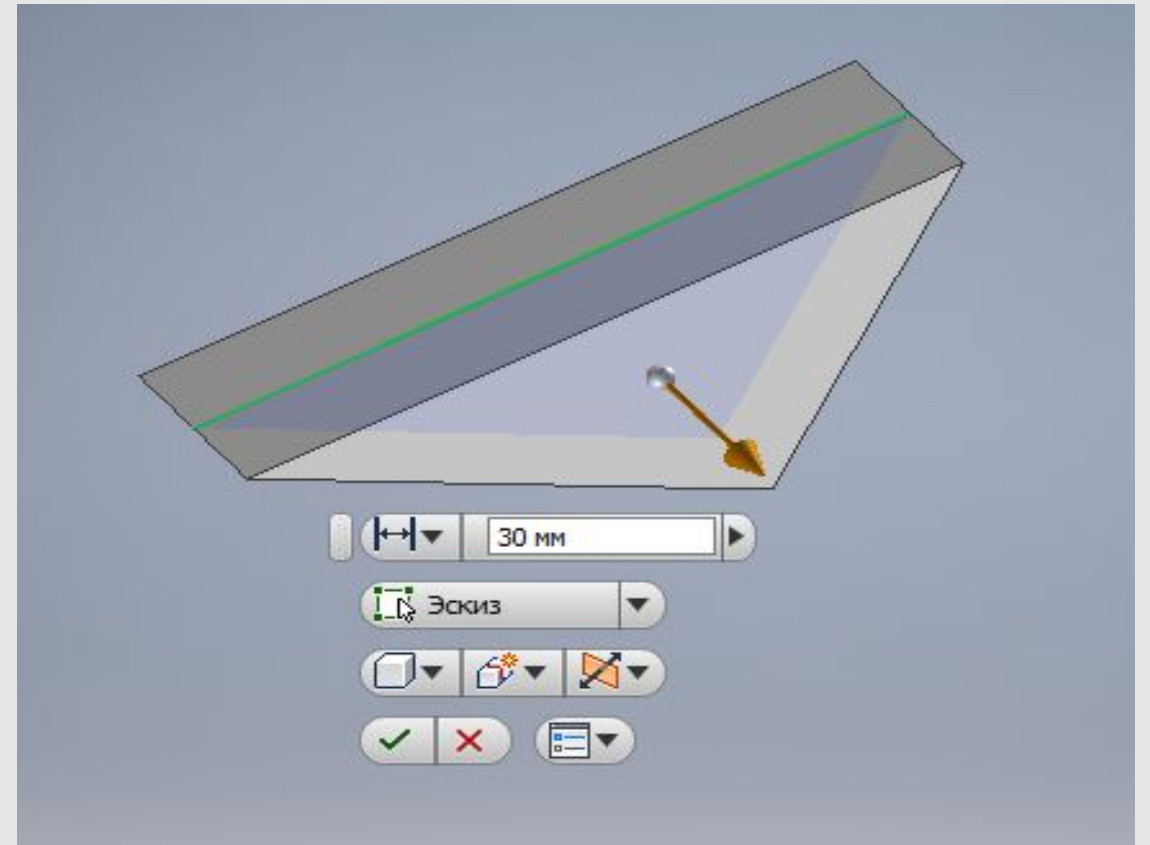
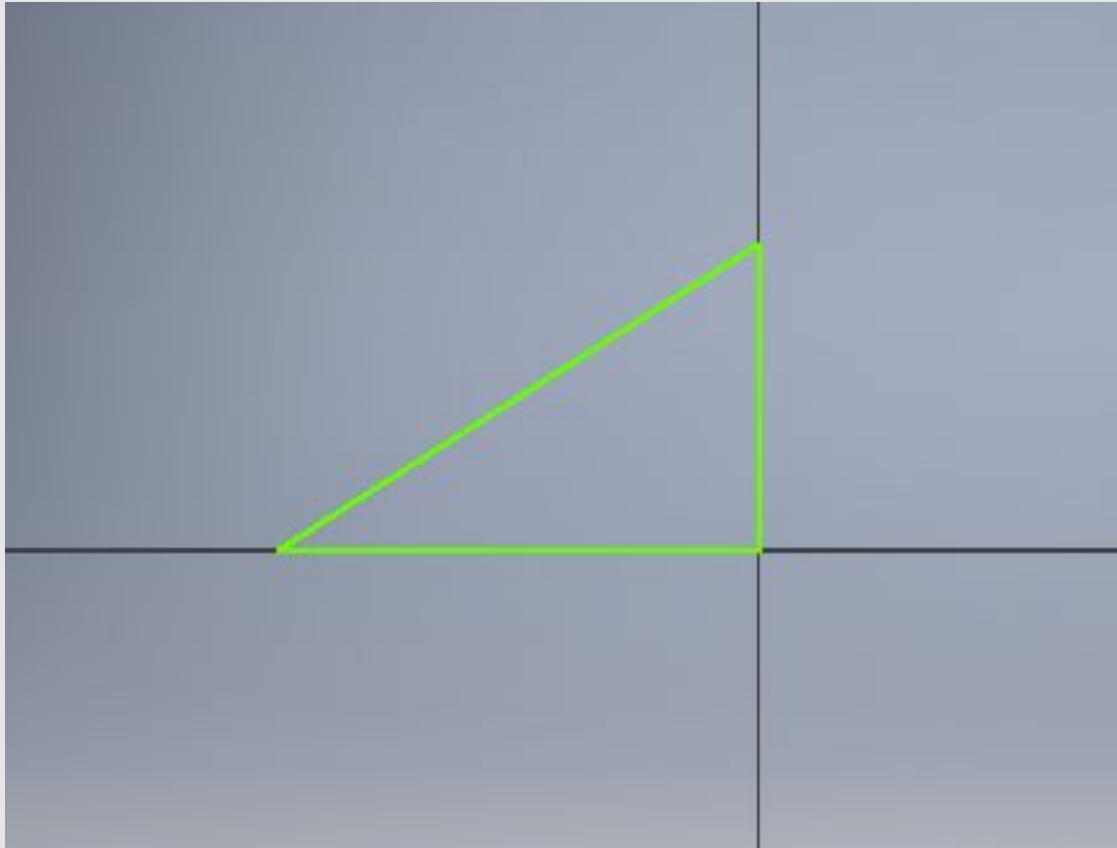


# Финальная конфигурация детали





# Алгоритм построения



```
/* Точки и линии треугольника*/
PlanarSketch *pSketch;
sketches->raw_Add(wp->GetItem(1), false, &pSketch);

SketchPointPtr point[3];
SketchLinePtr lines[3];

SketchPoints *skPoints;
SketchLines *skLines;
Profiles *skProfiles;

pSketch->get_SketchPoints(&skPoints);
pSketch->get_SketchLines(&skLines);
pSketch->get_Profiles(&skProfiles);

point[0] = skPoints->MethodAdd(pTransGeom->MethodCreatePoint2d(0, 0), false);
point[1] = skPoints->MethodAdd(pTransGeom->MethodCreatePoint2d(0, TriangleHeight), false);
point[2] = skPoints->MethodAdd(pTransGeom->MethodCreatePoint2d(-TriangleWeight, 0), false);

lines[0] = skLines->MethodAddByTwoPoints(point[0], point[1]);
lines[1] = skLines->MethodAddByTwoPoints(point[1], point[2]);
lines[2] = skLines->MethodAddByTwoPoints(point[2], point[0]);
```

```
/* Выдавливание треугольника */
Profile *pProfile;
skProfiles->raw__AddForSolid(&pProfile);

ExtrudeFeatures *ftExtrude;
ft->get_ExtrudeFeatures(&ftExtrude);

ExtrudeFeaturePtr oExtrudeDef = ftExtrude->MethodAddByDistanceExtent(pProfile, 3, kSymmetricExtentDirection, kJoinOperation);
```



```

/* Сопряжение треугольника 1 */
FilletFeatures *pFilletFt;
//FilletFeaturePtr *pFilletft2;
ft->get_FilletFeatures(&pFilletFt);

EdgeCollection *edgeColl;

pInvApp->TransientObjects->raw_CreateEdgeCollection(vtMissing, &edgeColl);

SurfaceBody *SurfBody;
SurfaceBodies *SurfBodies;

pPartComDef->get_SurfaceBodies(&SurfBodies);

SurfBodies->get_Item(1, &SurfBody);

// 1
Edges *edges;

SurfBody->get_Edges(&edges);

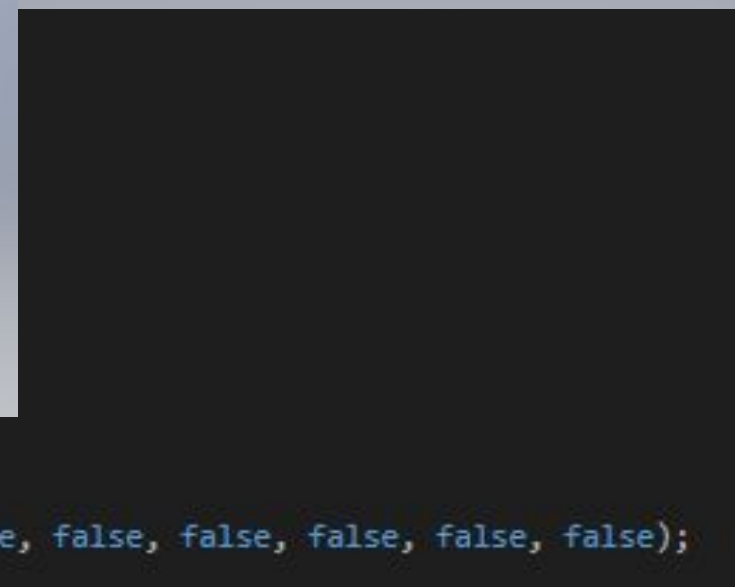
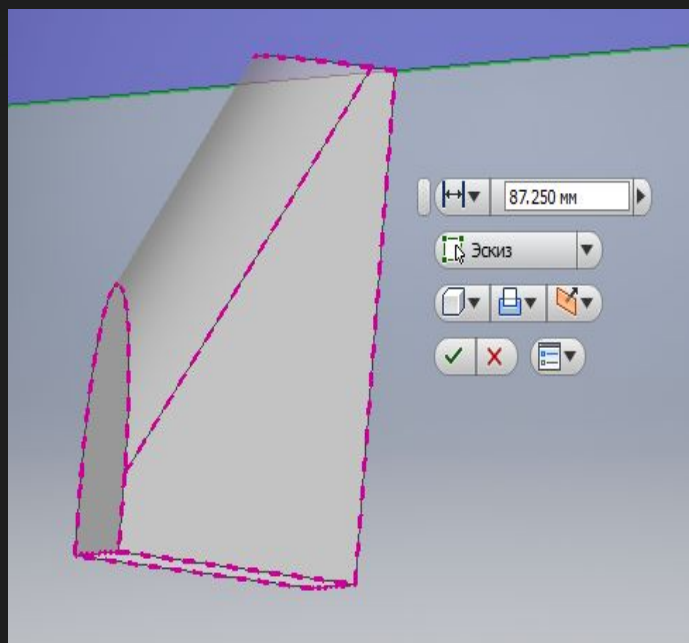
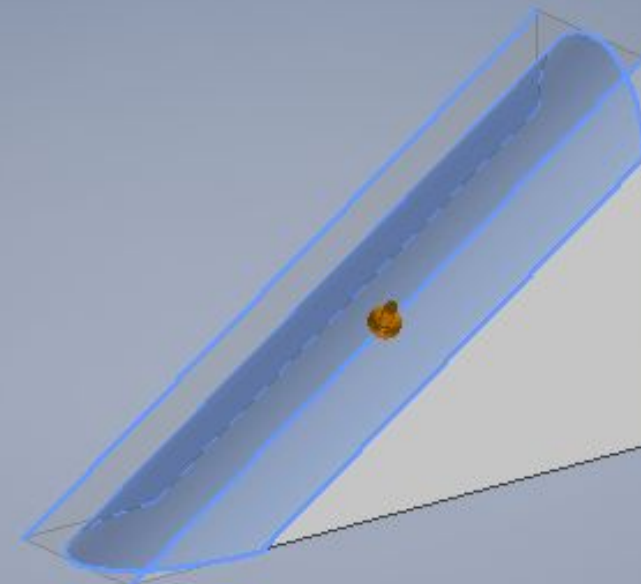
Edge *ed;
edges->get_Item(3, &ed);

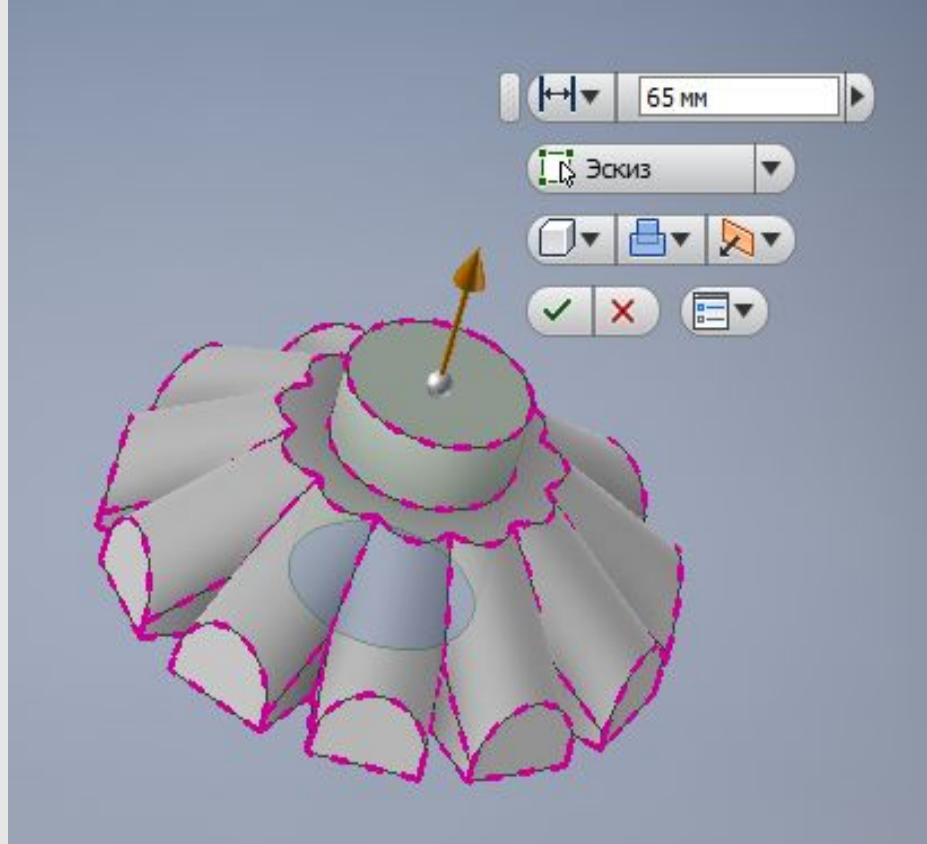
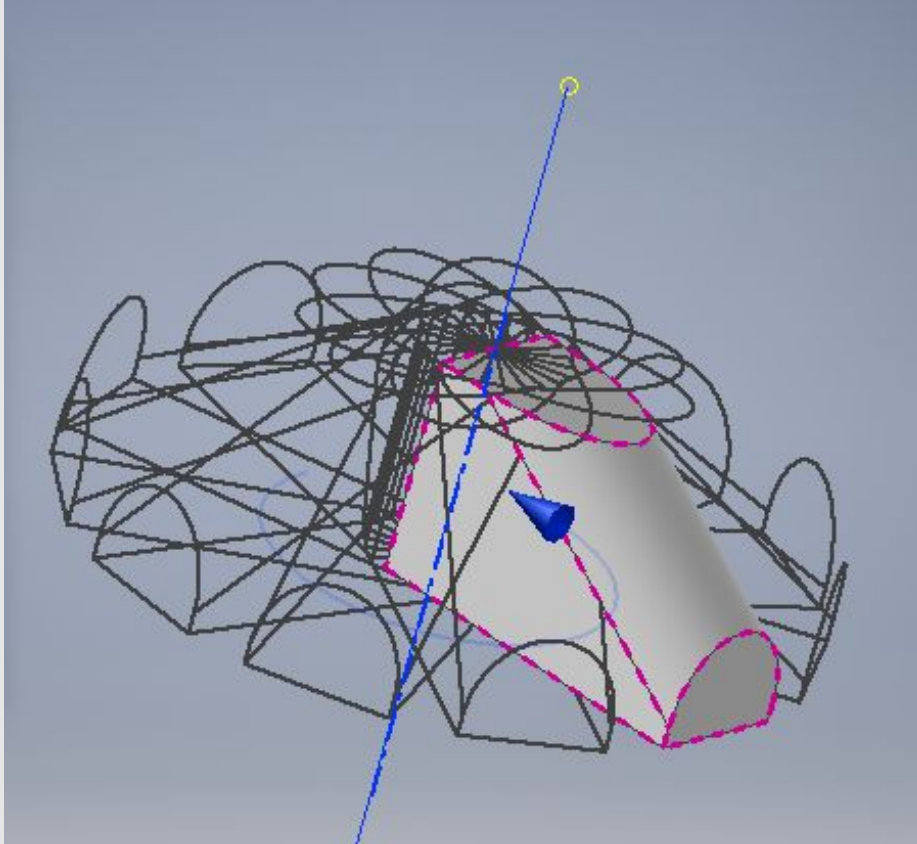
Edge *ed2;
edges->get_Item(7, &ed2);

edgeColl->MethodAdd(ed);
edgeColl->MethodAdd(ed2);

FilletFeaturePtr fillFeat = pFilletFt->MethodAddSimple(edgeColl, 1.5, false, false, false, false, false, false);

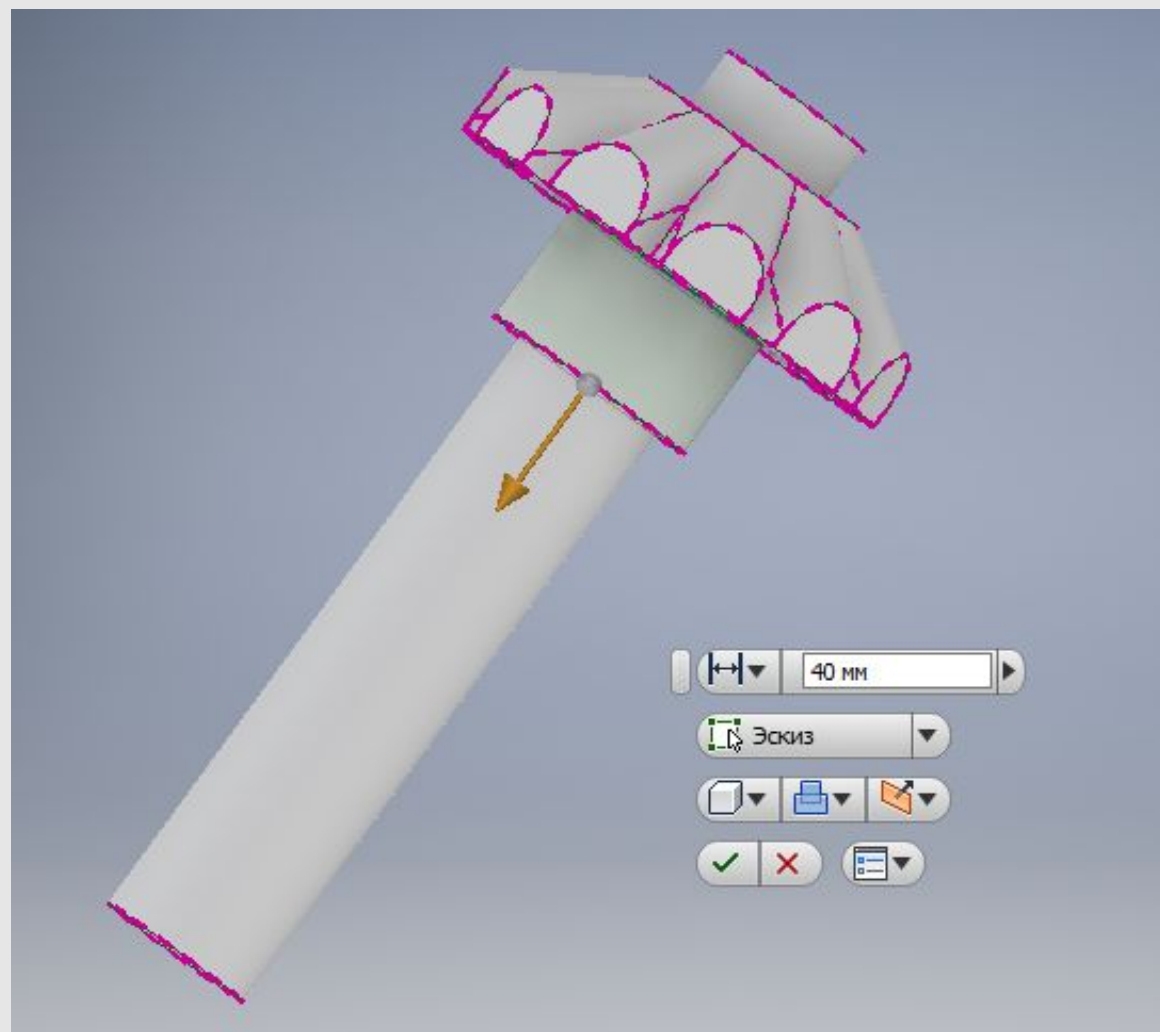
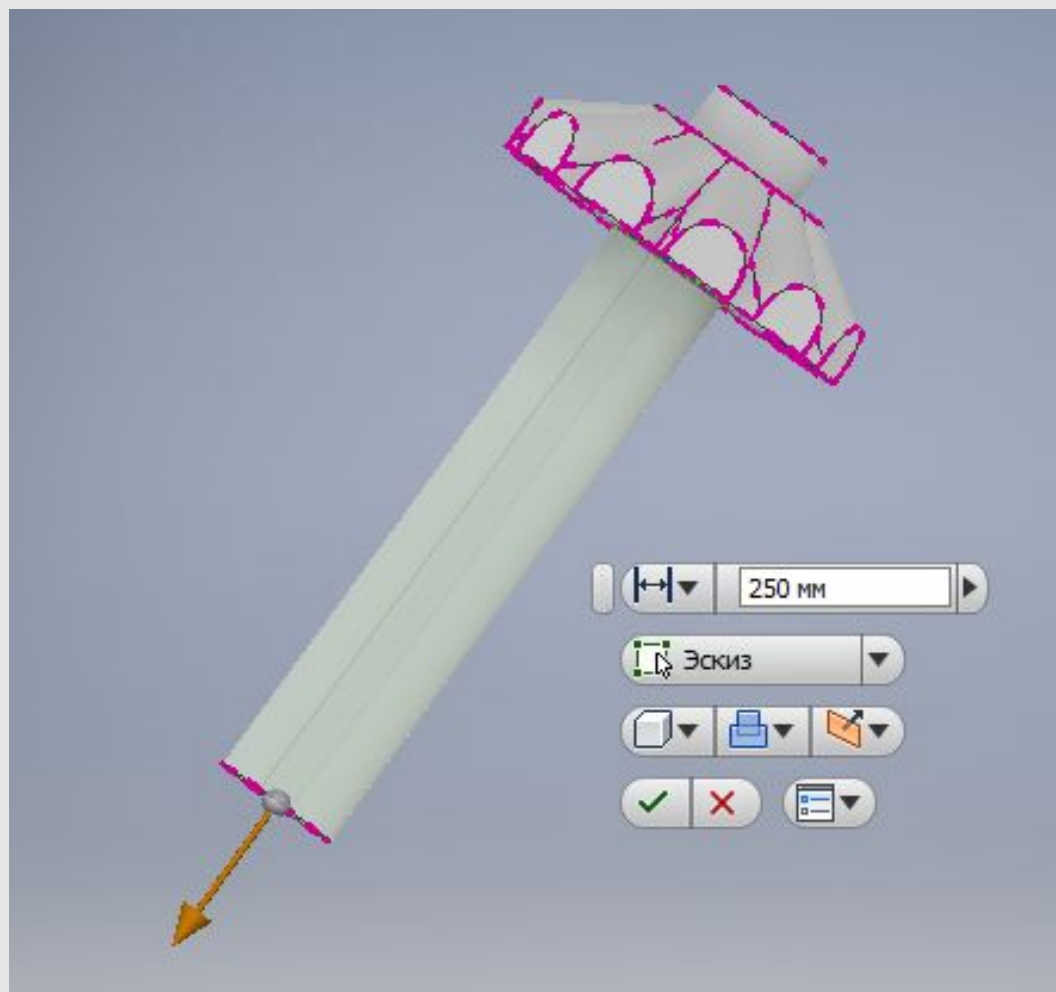
```

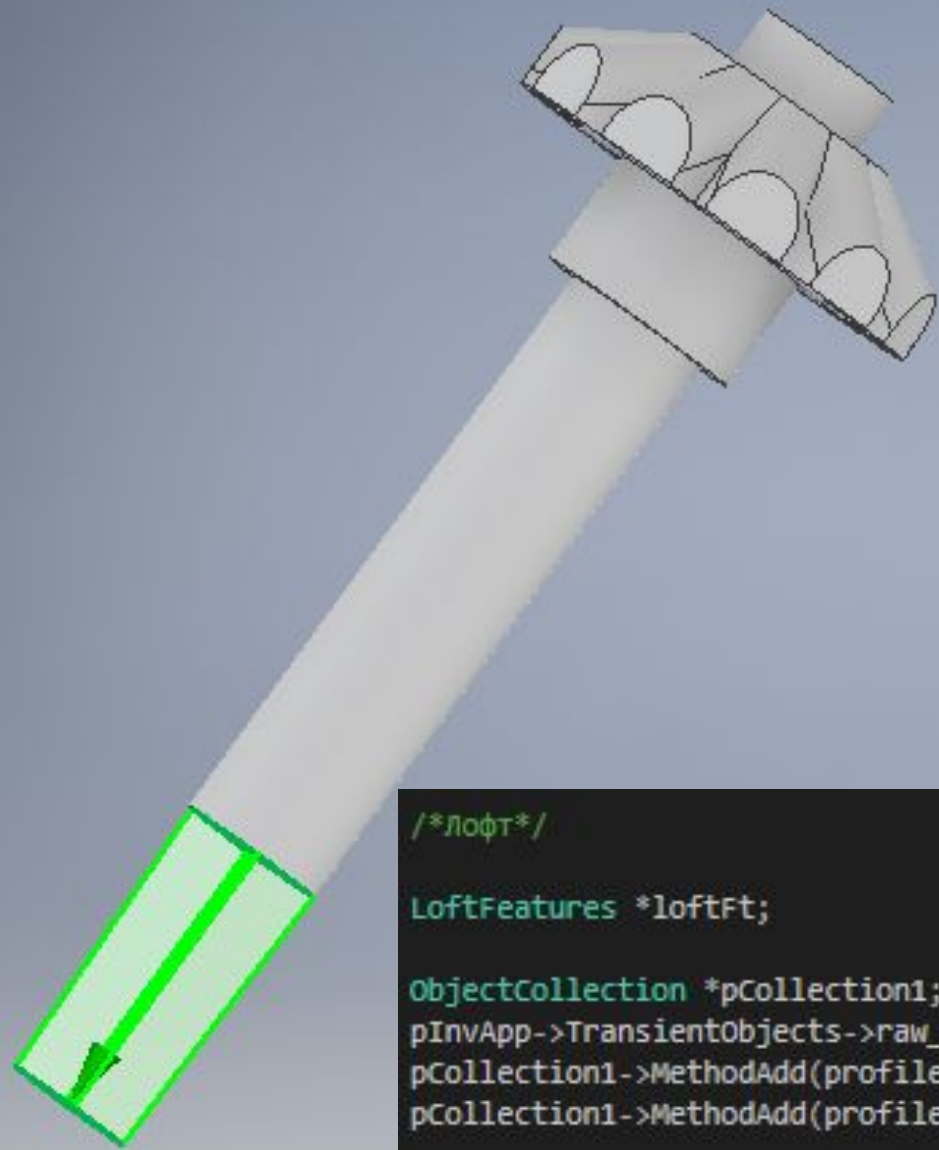




```
/*Круговой массив*/  
ObjectCollection *pCollection;  
  
pInvApp->TransientObjects->raw_CreateObjectCollection(vtMissing, &pCollection);  
  
pPartComDef->get_SurfaceBodies(&SurfBodies);  
  
SurfBodies->get_Item(1, &SurfBody);  
  
pCollection->MethodAdd(SurfBody);  
  
CircularPatternFeaturesPtr pCircPatFeat;  
  
ft->get_CircularPatternFeatures(&pCircPatFeat);  
  
CircularPatternFeature *circFeat = pCircPatFeat->MethodAdd(pCollection, wax->GetItem(3), true, NumberBattlement, "360 deg", true, kIdenticalCompute
```







```
/*ЛОФТ*/
```

```
LoftFeatures *loftFt;
```

```
ObjectCollection *pCollection1;
```

```
pInvApp->TransientObjects->raw_CreateObjectCollection(vtMissing, &pCollection1);
```

```
pCollection1->MethodAdd(profile7);
```

```
pCollection1->MethodAdd(profile6);
```

```
ft->get_LoфтFeatures(&loftFt);
```

```
LoftFeature *oLoфтDef8 = loftFt->MethodAdd(loftFt->MethodCreateLoфтDefinition(pCollection1, kNewBodyOperation));
```

# Проверка пределов вводимых значений

```
void CTestInventorCPPDlg::MinMax()  
{  
    UpdateData();  
    m_Min1 = 0;  
    m_Max1 = 35;  
    m_Min2 = 0;  
    m_Max2 = 55;  
    m_Min3 = 0;  
    m_Max3 = 55;  
    m_Min4 = 0;  
    m_Max4 = 12.5;  
    m_Min5 = 0;  
    m_Max5 = 17.5;  
    m_Min6 = 0;  
    m_Max6 = 20;  
    m_Min7 = 0;  
    m_Max7 = 32.5;  
    m_Min8 = 0;  
    m_Max8 = 125;  
    m_Min9 = 0;  
    m_Max9 = 8.7;  
    m_Min10 = 0;  
    m_Max10 = 8.5;  
    m_Min11 = 0;;  
    m_Max11 = 70;  
    UpdateData(false);  
}
```

```

// диалоговое окно CTestInventorCPPDlg
CTestInventorCPPDlg::CTestInventorCPPDlg(CWnd* pParent /*=NULL*/)
: CDialogEx(CTestInventorCPPDlg::IDD, pParent)
//Переменные по умолчанию
, TriangleHeight(7)
, TriangleWeight(11)
, NumberBattlement(11)
, RadiusSmallerCylinder(2.5)
, RadiusGreatCylinder(3.5)
, HeightGreatCylinder(4)
, HightSmallerCylinderUP(6.5)
, HightSmallerCylinderDOWN(25)
, SideUpperSquare(1.7)
, SideLowerSquare(1.5)
, HeightProtrusion(13)

, m_Min1(0)
, m_Min2(0)
, m_Min3(0)
, m_Min4(0)
, m_Min5(0)
, m_Min6(0)
, m_Min7(0)
, m_Min8(0)
, m_Min9(0)
, m_Min10(0)
, m_Max1(35)
, m_Max2(55)
, m_Max3(55)
, m_Max4(12.5)
, m_Max5(17.5)
, m_Max6(20)
, m_Max7(32.5)
, m_Max8(125)
, m_Max9(8.7)
, m_Max10(8.5)
, m_Min11(0)
, m_Max11(70)

{
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
};

```

```

void CTestInventorCPPDlg::DoDataExchange(CDataExchange* pDX)
{
CDialogEx::DoDataExchange(pDX);
DDX_Control(pDX, IDC_STATIC_PICTURE, m_ctrlPic);
DDX_Text(pDX, IDC_EDIT10, TriangleHeight);
DDX_Text(pDX, IDC_EDIT11, TriangleWeight);
DDX_Text(pDX, IDC_EDIT13, NumberBattlement);
DDX_Text(pDX, IDC_EDIT14, RadiusSmallerCylinder);
DDX_Text(pDX, IDC_EDIT2, RadiusGreatCylinder);
DDX_Text(pDX, IDC_EDIT3, HeightGreatCylinder);
DDX_Text(pDX, IDC_EDIT4, HightSmallerCylinderUP);
DDX_Text(pDX, IDC_EDIT6, HightSmallerCylinderDOWN);
DDX_Text(pDX, IDC_EDIT1, SideUpperSquare);
DDX_Text(pDX, IDC_EDIT5, SideLowerSquare);
DDX_Text(pDX, IDC_EDIT8, m_Min1);
DDX_Text(pDX, IDC_EDIT9, m_Min2);
DDX_Text(pDX, IDC_EDIT12, m_Min3);
DDX_Text(pDX, IDC_EDIT15, m_Min4);
DDX_Text(pDX, IDC_EDIT16, m_Min5);
DDX_Text(pDX, IDC_EDIT17, m_Min6);
DDX_Text(pDX, IDC_EDIT18, m_Min7);
DDX_Text(pDX, IDC_EDIT19, m_Min8);
DDX_Text(pDX, IDC_EDIT20, m_Min9);
DDX_Text(pDX, IDC_EDIT21, m_Min10);
DDX_Text(pDX, IDC_EDIT22, m_Max1);
DDX_Text(pDX, IDC_EDIT23, m_Max2);
DDX_Text(pDX, IDC_EDIT25, m_Max3);
DDX_Text(pDX, IDC_EDIT24, m_Max4);
DDX_Text(pDX, IDC_EDIT26, m_Max5);
DDX_Text(pDX, IDC_EDIT27, m_Max6);
DDX_Text(pDX, IDC_EDIT29, m_Max7);
DDX_Text(pDX, IDC_EDIT28, m_Max8);
DDX_Text(pDX, IDC_EDIT30, m_Max9);
DDX_Text(pDX, IDC_EDIT31, m_Max10);
DDX_Text(pDX, IDC_EDIT7, HeightProtrusion);
DDX_Text(pDX, IDC_EDIT32, m_Min11);
DDX_Text(pDX, IDC_EDIT33, m_Max11);
}

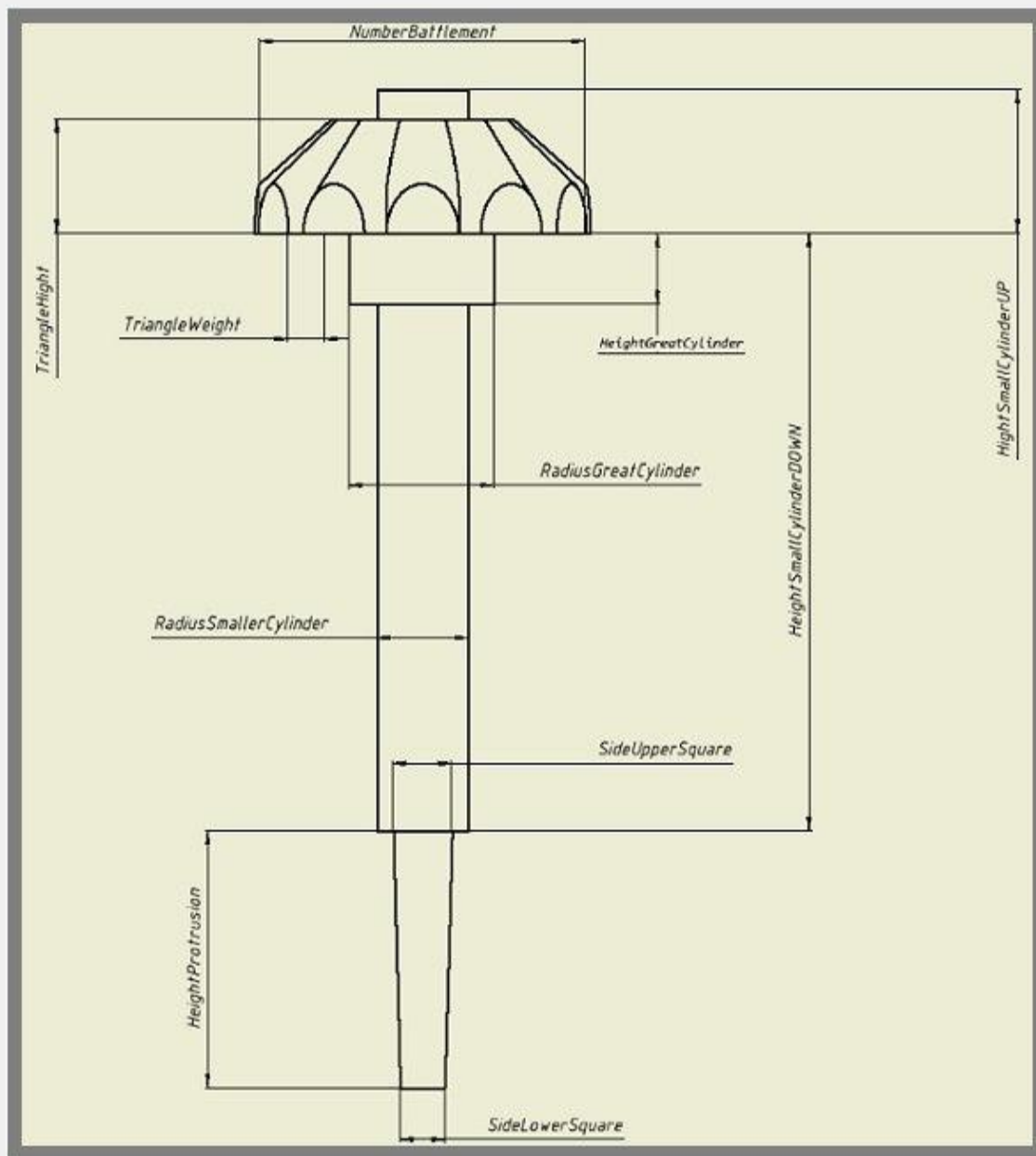
```



```
bool CTestInventorCPPDlg::CheckAllVariable()
{
    if ((TriangleHeight > 0 && TriangleHeight < 36) &&
        (TriangleWeight > 0 && TriangleWeight < 56) &&
        (NumberBattlement > 0 && NumberBattlement < 56) &&
        (RadiusSmallerCylinder > 0 && RadiusSmallerCylinder < 12.6) &&
        (RadiusGreatCylinder > 0 && RadiusGreatCylinder < 17.6) &&
        (HeightGreatCylinder > 0 && HeightGreatCylinder < 21) &&
        (HightSmallerCylinderUP > 0 && HightSmallerCylinderUP < 32.6) &&
        (HightSmallerCylinderDOWN > 0 && HightSmallerCylinderDOWN < 126) &&
        (SideUpperSquare > 0 && SideUpperSquare < 8.8) &&
        (SideLowerSquare > 0 && SideLowerSquare < 8.6) &&
        (HeightProtrusion > 0 && HeightProtrusion < 71))
        return true;
    else
        return false;
}
```



```
CString strerror;
strerror = _T("Неверные параметры:\n");
//это пример как писать границы параметра
if (!(TriangleHeight > 0 && TriangleHeight < 36))
    strerror += "TriangleHeight\n";
if (!(TriangleWeight > 0 && TriangleWeight < 56))
    strerror += "TriangleWeight\n";
if (!(NumberBattlement > 0 && NumberBattlement < 56))
    strerror += "NumberBattlement\n";
if (!(RadiusSmallerCylinder > 0 && RadiusSmallerCylinder < 12.6))
    strerror += "RadiusSmallerCylinder\n";
if (!(RadiusGreatCylinder > 0 && RadiusGreatCylinder < 17.6))
    strerror += "RadiusGreatCylinder\n";
if (!(HightSmallerCylinderUP > 0 && HightSmallerCylinderUP < 32.6))
    strerror += "HightSmallerCylinderUP\n";
if (!(HightSmallerCylinderDOWN > 0 && HightSmallerCylinderDOWN < 126))
    strerror += "HightSmallerCylinderDOWN\n";
if (!(SideUpperSquare > 0 && SideUpperSquare < 8.8))
    strerror += "SideUpperSquare\n";
if (!(SideLowerSquare > 0 && SideLowerSquare < 8.6))
    strerror += "SideLowerSquare\n";
if (!(HeightProtrusion > 0 && HeightProtrusion < 71))
    strerror += "HeightProtrusion\n";
//if (Сюда надо написать все границы своих параметров)
ErrorAll(strerror);
}
```



		Min	Max
TriangleHeight	<input type="text" value="7"/>	<input type="text" value="0"/>	<input type="text" value="35"/>
TriangleWeight	<input type="text" value="11"/>	<input type="text" value="0"/>	<input type="text" value="55"/>
NumberBattlement	<input type="text" value="11"/>	<input type="text" value="0"/>	<input type="text" value="55"/>
RadiusSmallerCylinder	<input type="text" value="2.5"/>	<input type="text" value="0"/>	<input type="text" value="12.5"/>
HightSmallerCylinderUP	<input type="text" value="6.5"/>	<input type="text" value="0"/>	<input type="text" value="32.5"/>
HightSmallerCylinderDOWN	<input type="text" value="25"/>	<input type="text" value="0"/>	<input type="text" value="125"/>
RadiusGreatCylinder	<input type="text" value="3.5"/>	<input type="text" value="0"/>	<input type="text" value="17.5"/>
HeightGreatCylinder	<input type="text" value="4"/>	<input type="text" value="0"/>	<input type="text" value="20"/>
SideUpperSquare	<input type="text" value="1.7"/>	<input type="text" value="0"/>	<input type="text" value="8.7"/>
SideLowerSquare	<input type="text" value="1.5"/>	<input type="text" value="0"/>	<input type="text" value="8.5"/>
HeightProtrusion	<input type="text" value="13"/>	<input type="text" value="0"/>	<input type="text" value="70"/>

Отмена

Создать

**Спасибо за внимание!**