**ПРИМЕР №5**

Задача: реализация 3D модели вазы

**Поэтапное построение эскиза.**

1. Создаем локальную СК:  
   MbPlacement3Dpl;
2. Затем создаем массив точек, вершин ломаной, и объявляем их.

Создаем массив точек:

SArray<MbCartPoint>arrPnts(19);(возможно, тут надо описать, какое слово что значит)

Добавляем в массив точки:

arrPnts.Add(MbCartPoint(30, 40));

arrPnts.Add(MbCartPoint(30, 20));

arrPnts.Add(MbCartPoint(0, 20));

arrPnts.Add(MbCartPoint(0, 0));

arrPnts.Add(MbCartPoint(100, 0));

arrPnts.Add(MbCartPoint(100, 20));

arrPnts.Add(MbCartPoint(70, 20));

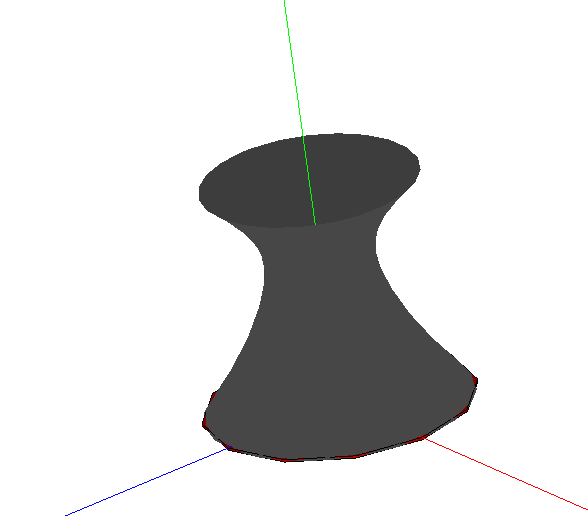
arrPnts.Add(MbCartPoint(70, 40));

1. Создаем ломаную:

MbPolyline\* pPolyline = newMbPolyline(arrPnts, false/\* Флаг незамкнутой линии \*/);

1. Создаем фаску:  
   ChamferPolyContour(pPolyline, 5,5, false, false, arrPnts[4]);
2. Создаем контур:  
   MbContour\* pContour = newMbContour(\*pPolyline, true);
3. Создаем скругление:  
   FilletPolyContour(pPolyline, 5, false, arrPnts[2], pContour);
4. Создаем дугу (Арку):  
   MbCartPointarcCenter(50, 40);   
   const double RADIUS = 20;  
   MbArc\* pArc = new MbArc(arcCenter, RADIUS, arrPnts[7], arrPnts[0], 1 /\*initSense\*/);
5. Добавляем арку в контур:  
   pContour->AddSegment(pArc);
6. Добавляем контур в сцену:  
   viewManager->AddObject(Style(1, RGB(0, 0, 255)), pContour, &pl);
7. Уменьшаем счетчик ссылок динамически созданных объектов ядра:  
   ::DeleteItem(pPolyline);  
   ::DeleteItem(pArc);  
   ::DeleteItem(pContour);

**Результат построения.**



**Код программы.**

void MakeUserCommand8()

{

// Базовая поверхность вращения: построение по образующей, оси вращения и углу

SArray<MbCartPoint3D> arrGenPnts(7);

arrGenPnts.Add(MbCartPoint3D(0, -4.5, 0));

arrGenPnts.Add(MbCartPoint3D(4, 10, 0));

arrGenPnts.Add(MbCartPoint3D(0, 15, 0));

MbBezier3D\* pGenCurve = new MbBezier3D(arrGenPnts, false);

MbAxis3D axRev(MbCartPoint3D(0, 0, 0), MbVector3D(0, 1, 0));

MbRevolutionSurface\* pBaseSurf = new MbRevolutionSurface(

\*pGenCurve, axRev, 2 \* M\_PI, false);

// Построение эквидистантной поверхности

MbOffsetSurface\* pOffSurf = new MbOffsetSurface(\*pBaseSurf, 7, false);

// Построение цилиндрической поверхности

double height\_Cyl = -0.1;

double radius\_Cyl = 5.5;

MbCartPoint3D baseCenter1(0, 0, 0); // Центр первого основания

MbCartPoint3D baseCenter2(0, height\_Cyl, 0); // Центр второго основания

// Точка на втором основании для указания радиуса цилиндра

MbCartPoint3D pntOnBase2(radius\_Cyl, height\_Cyl, 0);

// Вызов функции ядра для создания элементарной поверхности

MbSurface\* pCylSurf = NULL;

MbResultType resCylSurf = ::ElementarySurface(baseCenter1, baseCenter2, pntOnBase2,

st\_CylinderSurface, pCylSurf);

// Построение цилиндрического тела

MbSolid\* pCyl = NULL;

if (resCylSurf == rt\_Success)

{

// Вспомогательный объект для именования составных элементов твердого тела

MbSNameMaker namesCyl(ct\_ElementarySolid, MbSNameMaker::i\_SideNone, 0);

// Вызов функции ядра для построения тела на основе элементарной поверхности

MbResultType resSolid = ::ElementarySolid(\*pCylSurf, namesCyl, pCyl);

if (resSolid == rt\_Success)

viewManager->AddObject(Style(1, LIGHTRED), pCyl);

}

viewManager->AddObject(Style(1, LIGHTGRAY), pOffSurf);

// Уменьшение счетчика ссылок динамически созданных объектов ядра

::DeleteItem(pGenCurve);

::DeleteItem(pBaseSurf);

::DeleteItem(pOffSurf);

::DeleteItem(pCylSurf);

::DeleteItem(pCyl);

}