**ПРИМЕР №10**

Задача: реализация 3D модели кинематическая поверхность (колено)

**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));

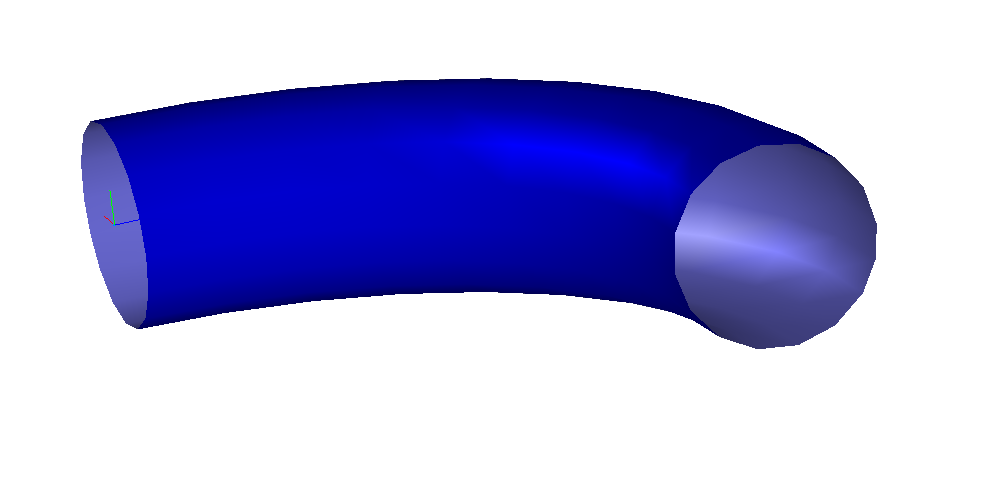
arrPnts.Add(MbCartPoint(70, 40));

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 MakeUserCommand3()

{

const double DEG\_TO\_RAD = M\_PI / 180.0;

MbPlacement3D plArc; // СК для построения первой образующей (совпадает с мировой)

MbPlacement3D plCurve; // СК для построения направляющей (вычисляется далее)

// Поворот локальной СК направляющей кривой из плоскости XY в плоскость XZ

// мировой системы координат

plCurve.Rotate(MbAxis3D(MbVector3D(MbCartPoint3D(0, 0, 0), MbCartPoint3D(1, 0, 0))),

90 \* DEG\_TO\_RAD);

// Построение двумерной образующей кривой - окружности

const double RAD = 10;

const MbCartPoint arcCenter(0, 0);

// Построение окружности на плоскости по центру и радиусу

MbArc\* pArc2D = new MbArc(arcCenter, RAD);

// Построение окружности в трехмерном пространстве

MbArc3D\* pArc = new MbArc3D(\*pArc2D, plArc);

// Построение направляющей кривой - дуги окружности

// Сначала строится дуга двумерной окружности по центру окружности, радиусу,

// начальной и конечной точкам

MbArc\* pCurve2D = new MbArc(MbCartPoint(-50, 0), 50, MbCartPoint(0, 0),

MbCartPoint(-50, 50), 1);

// Построение дуги трехмерной окружности

MbArc3D\* pCurve = new MbArc3D(\*pCurve2D, plCurve);

// Вызов функции построения кинематической поверхности

MbSurface\* pSurface = NULL;

::EvolutionSurface(\*pArc, \*pCurve, pSurface);

// Отображение построенной поверхности

viewManager->AddObject(Style(1, RGB(0, 0, 255)), pSurface);

::DeleteItem(pSurface);

::DeleteItem(pCurve2D);

::DeleteItem(pCurve);

::DeleteItem(pArc2D);

::DeleteItem(pArc);

}