**ПРИМЕР №11**

Задача: реализация 3D Построение изогнутой поверхонсти

**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));

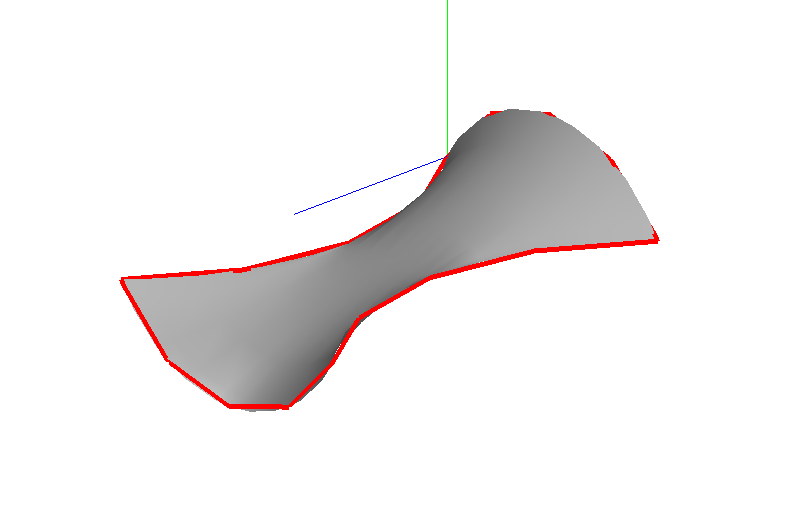
arrPnts.Add(MbCartPoint(70, 40));

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 MakeUserCommand5()

{

// Создание границ поверхности - четырех соединяющихся круговых

// дуг pCurve0, pCurve1, pCurve2, pCurve3.

// Дуги строятся по трем точкам (используются начальная, средняя и

// конечная точка дуги).

// При задании точек надо соблюдать направление дуг, чтобы они стыковались

// в одном направлении обхода.

// Дуга в плоскости XY

MbCartPoint3D pc01(0, 0, 0), pc02(5, 3, 0), pc03(10, 0, 0);

MbArc3D\* pCurve0 = new MbArc3D(pc01, pc02, pc03, 1, false);

// Дуга в плоскости ZX, начинающаяся в конечной точке pc01 кривой pCurve0

MbCartPoint3D pc11 = pc03, pc12(7, 0, 7.5), pc13(10, 0, 15);

MbArc3D\* pCurve1 = new MbArc3D(pc11, pc12, pc13, 1, false);

// Дуга в плоскости YX (располагается ниже плоскости ZX), начинающаяся

// в конечной точке pCurve2.

MbCartPoint3D pc21 = pc13, pc22(5, -3, 15), pc23(0, 0, 15);

MbArc3D\* pCurve2 = new MbArc3D(pc21, pc22, pc23, 1, false);

// Дуга в плоскости ZX, соединяющаяся конечную точку pCurve2 с начальной

// точкой pCurve0. (Дуга pCurve3 зеркально симметрична для pCurve1)

MbCartPoint3D pc31 = pc23, pc32(3, 0, 7.5), pc33 = pc01;

MbArc3D\* pCurve3 = new MbArc3D(pc31, pc32, pc33, 1, false);

// Построение поверхности

MbCoverSurface\* pSurf = new MbCoverSurface(\*pCurve0, \*pCurve1, \*pCurve2, \*pCurve3);

// Отображение поверхности

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

// Отображение граничных кривых

viewManager->AddObject(Style(5, LIGHTRED), pCurve0);

viewManager->AddObject(Style(5, LIGHTRED), pCurve1);

viewManager->AddObject(Style(5, LIGHTRED), pCurve2);

viewManager->AddObject(Style(5, LIGHTRED), pCurve3);

// Уменьшение счетчика ссылок динамически созданных объектов ядра

::DeleteItem(pSurf);

::DeleteItem(pCurve0);

::DeleteItem(pCurve1);

::DeleteItem(pCurve2);

::DeleteItem(pCurve3);

}