**Пример №1**

Задача: реализация 2D эскиза. Создание контура с использованием отрезков, дуг, скругления и фаски.

**Этапное построение эскиза.**

1. Создаем локальную СК:

MbPlacement3Dpl;

1. 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. Создаемфаску  
     
   ChamferPolyContour(pPolyline, 5,5, false, false, arrPnts[4]);
2. Создаемконтур  
     
   MbContour\* pContour = newMbContour(\*pPolyline, true);
3. Создаемскругление.   
     
   FilletPolyContour(pPolyline, 5, false, arrPnts[2], pContour);
4. Создаемдугу (Арку):  
     
   MbCartPointarcCenter(50, 40);   
   constdouble RADIUS = 20;  
   MbArc\* pArc = newMbArc(arcCenter, RADIUS, arrPnts[7], arrPnts[0], 1 /\*initSense\*/);
5. Добавляемаркувконтур:  
     
   pContour->AddSegment(pArc);
6. Добавляемконтурвсцену:  
     
   viewManager->AddObject(Style(1, RGB(0, 0, 255)), pContour, &pl);
7. Уменьшаемсчетчикссылокдинамическисозданныхобъектовядра:  
     
   ::DeleteItem(pPolyline);  
   ::DeleteItem(pArc);  
   ::DeleteItem(pContour);

**Кодпрограммы**

void MakeUserCommand0()

{

MbPlacement3Dpl; // ЛокальнаяСК (поумолчаниюсовпадаетсмировойСК)

// Вершиныломаной

// Создание массива точек

SArray<MbCartPoint>arrPnts(19);

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));

// ЛоманаялиниясвершинамиarrPnts

MbPolyline\* pPolyline = newMbPolyline(arrPnts, false/\* Флагнезамкнутойлинии \*/);

ChamferPolyContour(pPolyline, 5,5, false, false, arrPnts[4]);

MbContour\* pContour = newMbContour(\*pPolyline, true);

FilletPolyContour(pPolyline, 5, false, arrPnts[2], pContour);

// Дуга окружности для замыкания ломаной.

// При построении указывается центр, радиус, начальная и конечная точки и

// направление обхода дуги между этими точками (значение initSense>0 соответствует

// обходу против часовой стрелки, а initSense<0 – по часовой стрелке).

MbCartPointarcCenter(50, 40);

Constdouble RADIUS = 20;

MbArc\* pArc = newMbArc(arcCenter, RADIUS, arrPnts[7], arrPnts[0], 1 /\*initSense\*/);

// Контуриздвухсегментов

pContour->AddSegment(pArc);

// Отображениеконтура

if (pContour)

viewManager->AddObject(Style(1, RGB(0, 0, 255)), pContour, &pl);

// Уменьшение счетчиков ссылок динамически созданных объектов ядра

::DeleteItem(pPolyline);

::DeleteItem(pArc);

::DeleteItem(pContour);

}

**Список используемых функций(определить формат и привести к должному виду)**

MbCartPoint::Add(X, Y); - Создание двумерной точки. X, Y – координаты точек.

MbPolyline\* pPolyline = newMbPolyline(arrPnts, false); - Создание контура из массива точек.

ChamferPolyContour(pPolyline, 5.5, false, false, arrPnts[4]); - Созданиефаски

MbContour\* pContour = newMbContour(\*pPolyline, true); - Применениефаскинаконтуре

FilletPolyContour(pPolyline, 5, false, arrPnts[2], pContour); - Созданиесопряжениянаконтуре.

MbCartPointarcCenter; - создание центральной точки.

MbArc\* pArc = newMbArc(arcCenter, RADIUS, arrPnts[7], arrPnts[0], 1 /\*initSense\*/); - Создаемаркунаэскизе.

pContour->AddSegment(pArc); - отображениеаркинаэскизе.

viewManager->AddObject(Style(1, RGB(0, 0, 255)), pContour, &pl); - отображениеконтура.

::DeleteItem(); - Данная функция уменьшает счетчик ссылок динамически созданных объектов ядра.