**Пример №3**

Задача: построение параллелепипеда с отверстием при помощи инструмента выдавливания.

**Этапное построение эскиза.**

1. Создадим отдельную функцию для создания эскиза.  
   Создадим контур квадрата по массиву точек.

SArray<MbCartPoint> arrPnts(8);  
 arrPnts.Add(MbCartPoint(0, 0));  
 arrPnts.Add(MbCartPoint(50, 0));  
 arrPnts.Add(MbCartPoint(50, 50));  
 arrPnts.Add(MbCartPoint(0, 50));  
MbPolyline\* pPolyline = new MbPolyline(arrPnts, true);  
MbContour\* pContour = new MbContour(\*pPolyline, true);

Создадим контур круглого отверстия при помощи функции окружности.   
const MbCartPoint centerCircle(25, 25);  
const double RAD = 10;  
MbArc\* pCircle = new MbArc(centerCircle, RAD);  
MbContour\* pContourCircle = new MbContour([\*pCircle](https://vk.com/pcircle), true);

1. Создадим локальную систему координат.  
   MbPlacement3D pl;
2. Создадим контур для образующей.  
   RPArray<MbContour> arrContours;  
   CreateSketch(arrContours);  
     
   Сделаем отображение данного контура.  
   for (int i = 0; i < arrContours.size(); i++)  
   viewManager->AddObject(Style(1, RGB(0, 0, 255)), arrContours[i], &pl);
3. Создадим плоскость, в которой будет располагаться образующая.  
   MbPlane\* pPlaneXY = new MbPlane(MbCartPoint3D(0, 0, 0), MbCartPoint3D(1, 0, 0),  
   MbCartPoint3D(0, 1, 0));
4. Создадим объект для хранения параметров образующей.  
   MbSweptData sweptData(\*pPlaneXY, arrContours);
5. Определим параметры для операции выдавливания.  
   Создадим направляющий вектор.  
   MbVector3D dir(0, 0, -1);  
     
   Зададим параметры операции выдавливания.  
   const double HEIGHT\_FORWARD = 100.0, HEIGHT\_BACKWARD = 0.0;  
   ExtrusionValues extrusionParam(HEIGHT\_FORWARD, HEIGHT\_BACKWARD);
6. Создадим объект для именования элементов модели твердого тела.  
   MbSNameMaker operNames(ct\_CurveExtrusionSolid, MbSNameMaker::i\_SideNone, 0);  
   PArray<MbSNameMaker> cNames(0, 1, false);
7. Создадим твердое тело.  
   MbSolid\* pSolid = NULL;  
   MbResultType res = ::ExtrusionSolid(sweptData, dir, NULL, NULL, false,  
   extrusionParam, operNames, cNames, pSolid);
8. Отобразим модель твердого тела.  
   if (res == rt\_Success)  
    {  
    viewManager->AddObject(Style(1, LIGHTGRAY), pSolid);  
    }
9. Уменьшение счетчиков ссылок динамически созданных объектов ядра  
   ::DeleteItem(pSolid);  
   ::DeleteItem(pPlaneXY);  
   for (int i = 0; i < arrContours.size(); i++)  
    ::DeleteItem(arrContours[i]);

**Код программы**

void CreateSketch(RPArray<MbContour>& \_arrContours)  
{  
// Создание массива точек квадрата, к которому в дальнейшем добавятся скругления.  
// Размер массива - 8 точек для учета точек четырех сегментов скруглений.  
 SArray<MbCartPoint> arrPnts(8);  
 arrPnts.Add(MbCartPoint(0, 0));  
 arrPnts.Add(MbCartPoint(50, 0));  
 arrPnts.Add(MbCartPoint(50, 50));  
 arrPnts.Add(MbCartPoint(0, 50));  
// Построение единой ломаной внешнего контура по точкам  
 MbPolyline\* pPolyline = new MbPolyline(arrPnts, true);  
 MbContour\* pContour = new MbContour(\*pPolyline, true);  
// Центры и радиусы окружностей, дуги которых входят в контур  
 const MbCartPoint centerCircle(25, 25);  
 const double RAD = 10;  
 MbArc\* pCircle = new MbArc(centerCircle, RAD);  
 MbContour\* pContourCircle = new MbContour([\*pCircle](https://vk.com/pcircle), true);  
 \_arrContours.push\_back(pContour);  
 \_arrContours.push\_back(pContourCircle);  
}

void MakeUserCommand0()  
{  
 MbPlacement3D pl; // Локальная СК (по умолчанию совпадает с мировой СК)  
// СОЗДАНИЕ КОНТУРОВ ДЛЯ ОБРАЗУЮЩЕЙ  
 RPArray<MbContour> arrContours;  
 CreateSketch(arrContours);  
  
// Отображение образующей (в плоскости XY глобальной СК)  
 for (int i = 0; i < arrContours.size(); i++)  
 viewManager->AddObject(Style(1, RGB(0, 0, 255)), arrContours[i], &pl);  
// ПОСТРОЕНИЕ ТЕЛА ВЫДАВЛИВАНИЯ  
// Образующая размещается на плоскости XY глобальной СК.  
// Важное замечание: объект-плоскость должен создаваться динамически,  
// поскольку он продолжает использоваться в объекте-твердом теле после  
// выхода из данной функции.  
 MbPlane\* pPlaneXY = new MbPlane(MbCartPoint3D(0, 0, 0), MbCartPoint3D(1, 0, 0),  
 MbCartPoint3D(0, 1, 0));  
// Объект, хранящий параметры образующей  
 MbSweptData sweptData(\*pPlaneXY, arrContours);  
// Направляющий вектор для операции выдавливания  
 MbVector3D dir(0, 0, -1);  
// Параметры операции выдавливания, задающие свойства тела для построения:  
// расстояние выдавливания в прямом и в обратном направлении вдоль  
// направляющего вектора  
 const double HEIGHT\_FORWARD = 100.0, HEIGHT\_BACKWARD = 0.0;  
 ExtrusionValues extrusionParam(HEIGHT\_FORWARD, HEIGHT\_BACKWARD);  
// Служебный объект для именования элементов модели твердого тела  
 MbSNameMaker operNames(ct\_CurveExtrusionSolid, MbSNameMaker::i\_SideNone, 0);  
 PArray<MbSNameMaker> cNames(0, 1, false);  
// Построение твердого тела выдавливания  
 MbSolid\* pSolid = NULL;  
 MbResultType res = ::ExtrusionSolid(sweptData, dir, NULL, NULL, false,  
 extrusionParam, operNames, cNames, pSolid);  
// Отображение построенного тела  
 if (res == rt\_Success)  
 {  
// Смещение тела по оси Y, чтобы при отображении оно не накладывалось на образующую  
 pSolid->Move(MbVector3D(0, 80, 0));  
 viewManager->AddObject(Style(1, LIGHTGRAY), pSolid);  
 }  
// Уменьшение счетчиков ссылок динамически созданных объектов ядра  
 ::DeleteItem(pSolid);  
 ::DeleteItem(pPlaneXY);  
 for (int i = 0; i < arrContours.size(); i++)  
 ::DeleteItem(arrContours[i]);  
}

**Список используемых функций**

CreateSketch – пользовательская функция для создание эскиза.

MbResultType::ExtrusionSOlid – функция создание модели выдавливания.

**Список используемых классов**

MbPolyline – класс для создания замкнутой кривой.

MbContour – класс для создания контура.

MbArc – класс дуги.

MbPlane – класс плоскости.

MbCartPoint3D – класс точки в трехмерном пространстве.

MbVector3D – класс вектора в трехмерном пространстве.

MbSweptData – класс данных образующей.

MbSNameMaker – класс для генерирования имен.

MbSolid – класс твердого тела.

MbResultType – класс для описания конечного результата ( в данном случае твердое тело выдавливания)