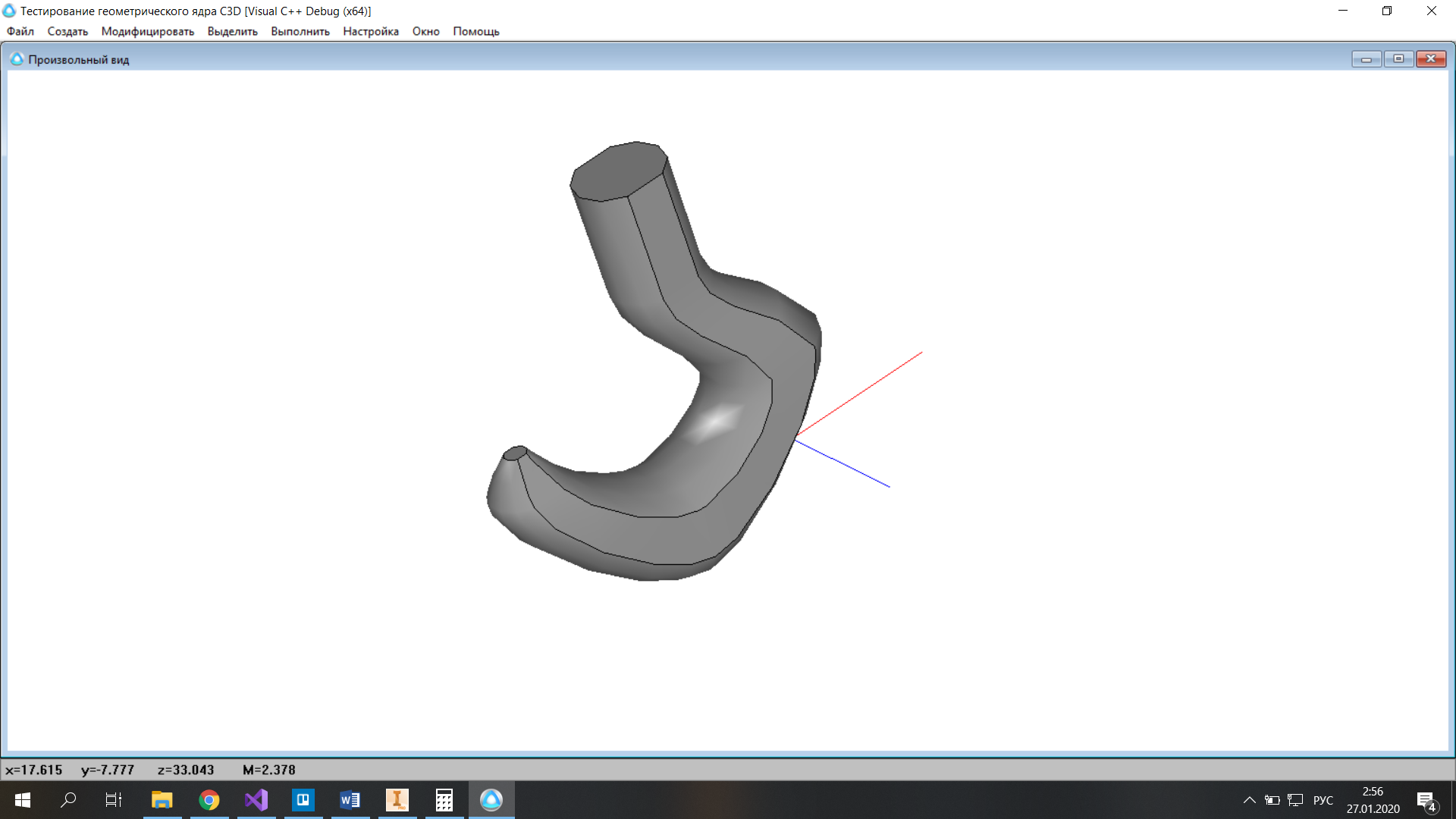
**Пример №8**

Задача: построение крюка (лофт).



**Поэтапное построение эскиза.**

1. Создаем функцию CreateSketch.

void CreateSketch(RPArray<MbSurface>& \_arrSurfaces, RPArray<MbContour>& \_arrContours, double vert,double hor, double OFS\_X, double OFS\_Y, double OFS\_Z, double d)

{

...

}

2. Затем описываем CreateSketch (Все функции, которые мы применили объяснены в комментариях).

double l = d \* 0.4472138576667362; //константа

SArray<MbCartPoint>arrPnts(4); //создание массива точек

arrPnts.Add(MbCartPoint(-l / 2, l));

arrPnts.Add(MbCartPoint(l / 2, l));

arrPnts.Add(MbCartPoint(-l / 2, -l));

arrPnts.Add(MbCartPoint(l / 2, -l));

MbLineSegment\* pS1 = new MbLineSegment(arrPnts[0], arrPnts[1]); //создание линии по двум точкам из массива

MbContour\* pContour = new MbContour(\*pS1, true); //создание контура, состоящего из линии pS1

MbArc\* pArc1 = new MbArc(MbCartPoint(0, 0), arrPnts[1], arrPnts[3], 0); //создание арки по центру и двум точкам из массива

pContour->AddSegment(pArc1); //добавление арки в контур

MbLineSegment\* pS2 = new MbLineSegment(arrPnts[3], arrPnts[2]);

pContour->AddSegment(pS2);

MbArc\* pArc2 = new MbArc(MbCartPoint(0, 0), arrPnts[2], arrPnts[0], 0);

pContour->AddSegment(pArc2);

MbPlacement3D plSurf(MbCartPoint3D(0 + OFS\_X, 0 + OFS\_Y, 0 + OFS\_Z), //создание локальной системы координат

MbCartPoint3D(hor + OFS\_X, vert + OFS\_Y, 0 + OFS\_Z), //создание точки в трехмерном пространстве

MbCartPoint3D(0 + OFS\_X, 0 + OFS\_Y, 1 + OFS\_Z));

MbSurface\* pSurf1 = new MbPlane(plSurf); //создание поверхности в локальной системе координат

\_arrSurfaces.push\_back(pSurf1);

\_arrContours.push\_back(pContour);

3. Переходим к описанию основной функции.

void MakeUserCommand0()

{

...

}

4. Первое что мы делам, создаем набор скетчей, они будут выступать образующими .

// Получение образующей в виде массивов поверхностей и контуров на них с помощью

// вспомогательной функции.

RPArray<MbSurface> arrSurfaces;

RPArray<MbContour> arrContours;

CreateSketch(arrSurfaces, arrContours, 0, 1, 30, 55, 0, 12);

CreateSketch(arrSurfaces, arrContours, 0, 1, 30, 40, 0, 12);

CreateSketch(arrSurfaces, arrContours, 0, 1, 40, 23, 0, 15);

CreateSketch(arrSurfaces, arrContours, 1, 0, 22, 10, 0, 15);

CreateSketch(arrSurfaces, arrContours, 0, 1, 5, 25, 0, 10);

CreateSketch(arrSurfaces, arrContours, 0, 1, 5, 30, 0, 3);

5. Создаем объект для операции лофта.

// Объект с параметрами операции заметания.

LoftedValues params;

6. Создаем объекты для наименование наших тел в тестовом приложении.

// Объекты для именования элементов модели твердого тела.

MbSNameMaker names(ct\_CurveLoftedSolid, MbSNameMaker::i\_SideNone, 0);

PArray<MbSNameMaker> contourNames(0, 1, false);

7. Создаем твердотельную модель при помощи лофта.

// Построение твердого тела заметания

MbSolid\* pSolid = NULL;

MbResultType res = ::LoftedSolid(arrSurfaces, arrContours, NULL, params, NULL, NULL, names, contourNames, pSolid);

8. Отображение построенного тела.

// Отображение построенного тела

if (res == rt\_Success)

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

9.Уменьшения счетчика ссылок.

// Уменьшение счетчика ссылок тела

::DeleteItem(pSolid);

**Код программы**

void CreateSketch (RPArray<MbSurface>& \_arrSurfaces, RPArray<MbContour>& \_arrContours, double vert,double hor, double OFS\_X, double OFS\_Y, double OFS\_Z, double d)

{

double l = d \* 0.4472138576667362;

SArray<MbCartPoint>arrPnts(4);

arrPnts.Add(MbCartPoint(-l / 2, l));

arrPnts.Add(MbCartPoint(l / 2, l));

arrPnts.Add(MbCartPoint(-l / 2, -l));

arrPnts.Add(MbCartPoint(l / 2, -l));

MbLineSegment\* pS1 = new MbLineSegment(arrPnts[0], arrPnts[1]);

MbContour\* pContour = new MbContour(\*pS1, true);

MbArc\* pArc1 = new MbArc(MbCartPoint(0, 0), arrPnts[1], arrPnts[3], 0);

pContour->AddSegment(pArc1);

MbLineSegment\* pS2 = new MbLineSegment(arrPnts[3], arrPnts[2]);

pContour->AddSegment(pS2);

MbArc\* pArc2 = new MbArc(MbCartPoint(0, 0), arrPnts[2], arrPnts[0], 0);

pContour->AddSegment(pArc2);

MbPlacement3D plSurf(MbCartPoint3D(0 + OFS\_X, 0 + OFS\_Y, 0 + OFS\_Z),

MbCartPoint3D(hor + OFS\_X, vert + OFS\_Y, 0 + OFS\_Z),

MbCartPoint3D(0 + OFS\_X, 0 + OFS\_Y, 1 + OFS\_Z));

MbSurface\* pSurf1 = new MbPlane(plSurf);

\_arrSurfaces.push\_back(pSurf1);

\_arrContours.push\_back(pContour);

}

void MakeUserCommand0()

{

// Получение образующей в виде массивов поверхностей и контуров на них с помощью

// вспомогательной функции.

RPArray<MbSurface> arrSurfaces;

RPArray<MbContour> arrContours;

CreateSketch(arrSurfaces, arrContours, 0, 1, 30, 55, 0, 12);

CreateSketch(arrSurfaces, arrContours, 0, 1, 30, 40, 0, 12);

CreateSketch(arrSurfaces, arrContours, 0, 1, 40, 23, 0, 15);

CreateSketch(arrSurfaces, arrContours, 1, 0, 22, 10, 0, 15);

CreateSketch(arrSurfaces, arrContours, 0, 1, 5, 25, 0, 10);

CreateSketch(arrSurfaces, arrContours, 0, 1, 5, 30, 0, 3);

// Объект с параметрами операции заметания.

LoftedValues params;

// Объекты для именования элементов модели твердого тела.

MbSNameMaker names(ct\_CurveLoftedSolid, MbSNameMaker::i\_SideNone, 0);

PArray<MbSNameMaker> contourNames(0, 1, false);

// Построение твердого тела заметания

MbSolid\* pSolid = NULL;

MbResultType res = ::LoftedSolid(arrSurfaces, arrContours, NULL, params,

NULL, NULL, names, contourNames, pSolid);

// Отображение построенного тела

if (res == rt\_Success)

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

// Уменьшение счетчика ссылок тела

::DeleteItem(pSolid);

}

**Список используемых функций**

CreateSketch – пользовательская функция для создание эскиза.

MbResultType::LoftedSolid – функция для построения тела заметания.

::DeleteItem – функция удаления ссылки на тело.

**Список используемых классов**

MbLineSegment - класс прямой в двумерном пространстве.

MbContour – класс контура в двумерном пространстве.

MbArc – класс дуги эллипса в двумерном пространстве.

MbPlacement3D – класс локальной системы координат в трехмерном пространстве.

MbCartPoint3D – класс трехмерной точки.

MbSurface – класс поверхности в трехмерном пространстве.

MbSolid – класс твердого тела.

MbSNameMaker – класс генератора имен с добавками к имени.