МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Кафедра Программирования и информационных систем

09.03.04 Программная инженерия

Отчет по лабораторным работам

Создание 3D параллелепипеда средствами API функций Siemens NX

Зав. Кафедрой _____*д.т.н., профессор С.Д. Махортов*

Обучающиеся _____ ст. 3 курса оч. отд. *А.А. Бредихина*

Руководитель _____ _____*д.т.н., профессор М.И. Чижов*

Воронеж 2023

Содержание

Введение

Целью данных лабораторных работ является построение параллелепипеда средствами API функций Siemens NX на языке C# с использованием Visual Studio 2022.

Требуемый параллелепипеда выглядит следующим образом

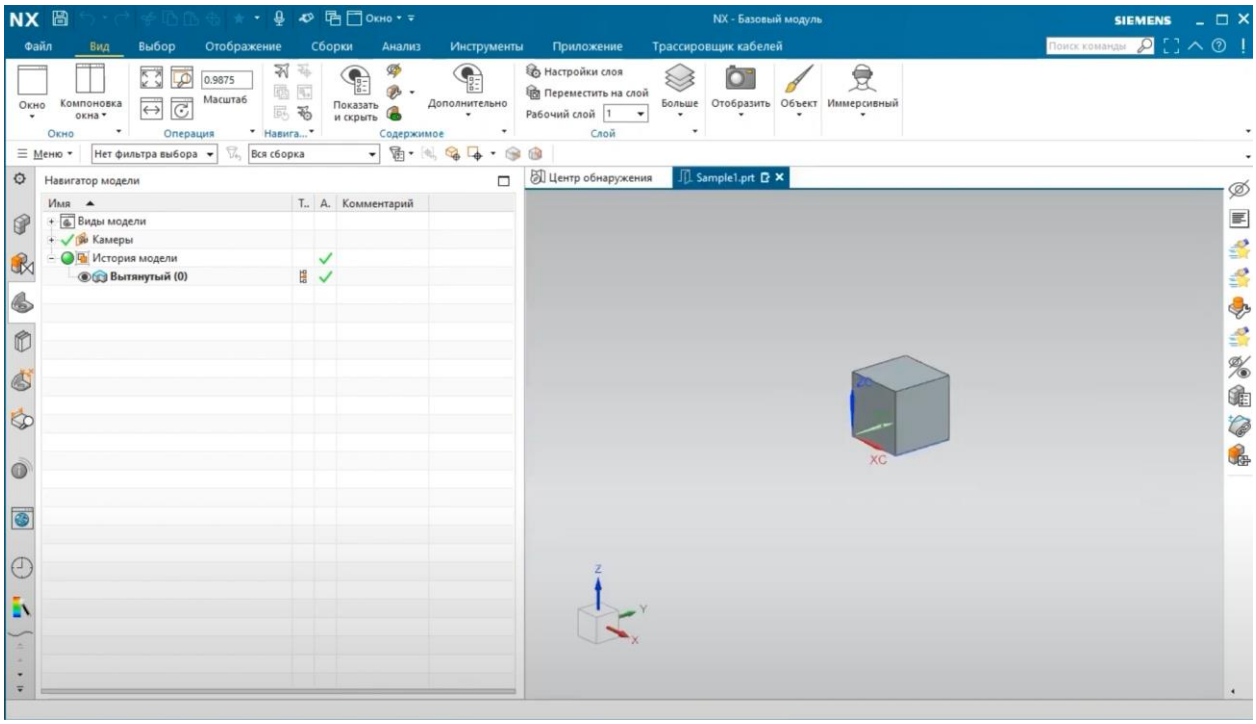

Рисунок 1 - Общий вид параллелепипеда

1 Создание и настройка проекта

Создадим пустой консольный проект, в котором будем выполнять NX Kompas v 21 нужно подключить библиотеки.

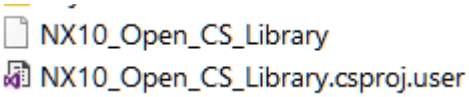

Рисунок 2 - Список подключаемых библиотек NX

2 Подключение нашего приложения к NX

Для того, чтобы подключить наше приложение к NX, выполним следующий код.

```
 try
        {
            theSession = Session.GetSession();
            theUI = UI.GetUI();
            theUFSession = UFSession.GetUFSession();
                        isDisposeCalled = false;
        }
        catch (NXOpen.NXException ex)
        {
            UI.GetUI().NXMessageBox.Show("Message", NXMessageBox.DialogType.Error,
ex.Message);
        }
```

3 Построение параллелепипеда

Процедура создания детали. Начинаем создавать объекты кривых, создаём линии.

```
Tag UFPart;

string part_name = "Sample1";
int units1 = 1;
theUFSession.Part.New(part_name, units1, out UFPart);
```

Начинаем создавать объекты кривых, создаём линии.

```
UFCurve.Line Line1 = new UFCurve.Line();
UFCurve.Line Line2 = new UFCurve.Line();
UFCurve.Line Line3 = new UFCurve.Line();
UFCurve.Line Line4 = new UFCurve.Line();


Line1.start_point = new double[3] { 0, 0, 0 };
Line1.end_point = new double[3] { 0, 20, 0 };
Line2.start_point = Line1.end_point;
Line2.end_point = new double[3] { 20, 20, 0 };
Line3.start_point = Line2.end_point;
Line3.end_point = new double[3] { 20, 0, 0 };
Line4.start_point = Line3.end_point;
Line4.end_point = new double[3] { 0, 0, 0 };


Tag[] SK1 = new Tag[4];

theUFSession.Curve.CreateLine(ref Line1, out SK1[0]);
theUFSession.Curve.CreateLine(ref Line2, out SK1[1]);
theUFSession.Curve.CreateLine(ref Line3, out SK1[2]);
theUFSession.Curve.CreateLine(ref Line4, out SK1[3]);
```

Вывод информационного окна.

```
NXOpen.Guide.InfoWrite("Построили элементы эскиза");
string[] Ext1_lim = { "0", "20" };
double[] point1 = new double[3];
double[] Ext1_dir = { 0, 0, 1 };


Tag[] Ext1;
theUFSession.Modl.CreateExtruded(SK1, "0", Ext1_lim, point1, Ext1_dir,
FeatureSigns.Nullsign, out Ext1);
NXOpen.Guide.InfoWrite("\n Построили элемент вытягивания");


theProgram.Dispose();
```

4 Заключение

В ходе выполнения данной лабораторной работы мы познакомились со средствами API функций NX. Создали динамически загружаемую библиотеку, которая может строить детали.