

## Практичне заняття 18.05.2020 р.

### Побудова 3D-моделі деталі з використанням мови програмування C# і Solidworks API

Ціль роботи – сформувати навички побудови 3D-моделі деталі в Solidworks з використанням C# Windows Forms.

#### Хід роботи

3D-модель, яку необхідно побудувати, – циліндр з визначеними розмірами (рис. 1) за варіантом, який відповідає порядковому номеру у наведеній таблиці 1.

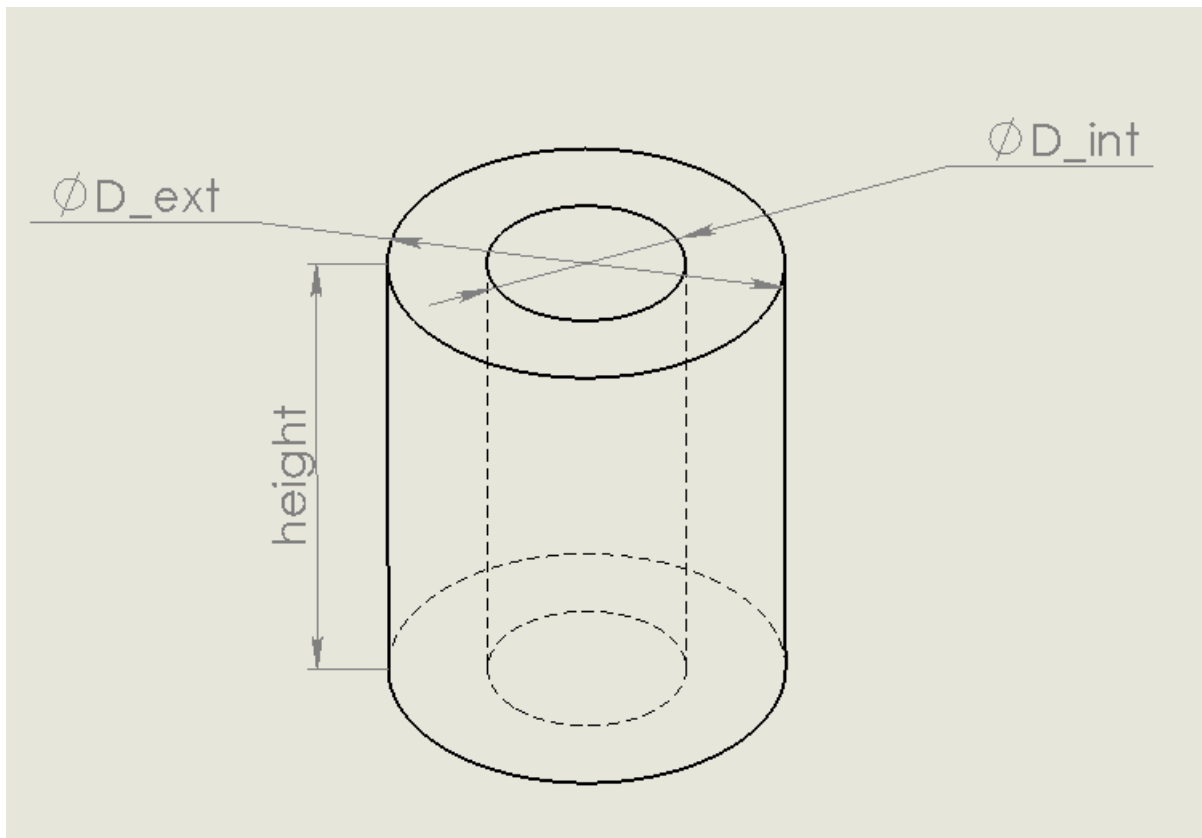


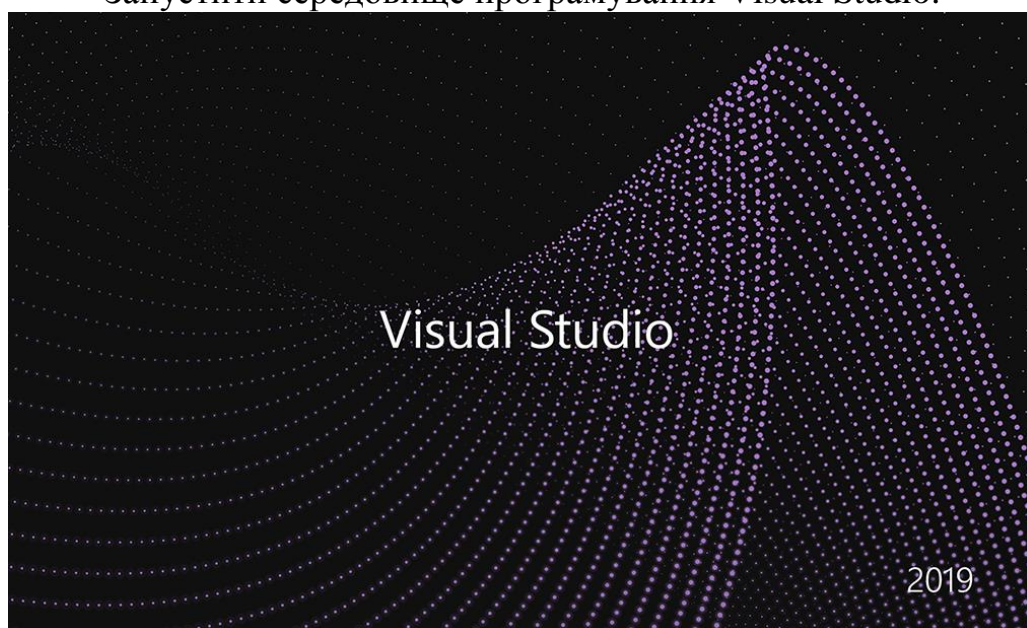
Рис. 1

Таблиця 1.

№	Прізвище	Ім'я	Група	Варіант	D_ext	D_int	height
1	Бовсунівський	Денис	Пі-58	1	160	107	120
2	Богданець	Віталій	Пі-58	2	159	106	122
3	Борейко	Юрій	Пі-58	3	158	105	124
4	Вознюк	Назарій	ПМ-138	4	157	104	126
5	Годлевський	Юрій	Пі-56	5	156	103	128
6	Горинович	Дмитро	Пі-58	6	155	102	130
7	Громський	Олександр	Пі-58	7	154	101	132

8	Козачинський	Дімітрій	КІ-3	8	153	100	134
9	Косачев	Антон	КБ-1	9	152	99	136
10	Костюк	Роман	Пі-58	10	151	98	138
11	Кравчук	Олександр	МБ-7	11	150	97	140
12	Крайнов	Владислав	МБ-7	12	149	96	142
13	Куліш	Владислав	Пі-58	13	148	95	144
14	Курачинська	Анастасія	КІ-3	14	147	94	146
15	Кучер	Богдан	Пі-58	15	146	93	148
16	Маліновський	Максим	МБ-7	16	145	94	150
17	Марковський	Артур	Пі-58	17	144	95	152
18	Осадчий	Сергій	Пі-58	18	143	96	154
19	Пастух	Богдан	МБ-7	19	142	97	156
20	Першко	Владислав	Пі-57	20	141	98	158
21	Петренко	Данііл	Пі-58	21	140	99	160
22	Примак	Артем	АТ-26	22	139	100	162
23	Проботюк	Андрій	КБ-1	23	138	101	164
24	Сафонюк	Ольга	Пі-58	24	137	102	166
25	Струць	Сергій	АТ-26	25	136	103	168
26	Тимчук	Володимир	МБ-7	26	135	104	170
27	Ткачук	Данііла	АТ-26	27	134	105	172
28	Труфаненко	Андрій	АТ-26	28	133	106	174
29	Халімончук	Олександр	МБ-7	29	132	107	176
30	Хоцький	Дмитро	Пі-58	30	131	108	178

Приклад.  
Запустити середовище програмування Visual Studio.



## Створити проект C# Windows Forms

### Создание проекта

#### Последние шаблоны проектов

Windows Forms App (.NET Core) C#

Поиск шаблонов (ALT+"B")

Очистить

C#

Windows

Все типы проектов



Консольное приложение (.NET Core)

Проект для создания приложения командной строки, которое может выполняться в среде .NET Core в Windows, Linux и Mac OS.

C# Linux macOS Windows Консоль



Библиотека классов (.NET Standard)

Проект для создания библиотеки классов, предназначенной для .NET Standard.

C# Android iOS Linux macOS Windows Библиотека



Проект тестов MSTest (.NET Core)

Проект, содержащий модульные тесты MSTest, которые можно выполнять на платформе .NET Core в Windows, Linux и MacOS.

C# Linux macOS Windows Тестирование



Тестовый проект NUnit (.NET Core)

Проект с тестами NUnit, которые могут выполняться на базе .NET Core в Windows, Linux и MacOS.

C# Linux macOS Windows Рабочий стол Тестирование Веб



Windows Forms App (.NET Core)

Проект для создания приложения с пользовательским интерфейсом Windows Forms (WinForms)

C# Windows Рабочий стол



## Им'я проекта - Cylinder

### Настроить новый проект

Windows Forms App (.NET Core) C# Windows Рабочий стол

Имя проекта

Cylinder

Расположение

C:\Users\admin\source\repos

Имя решения

Cylinder

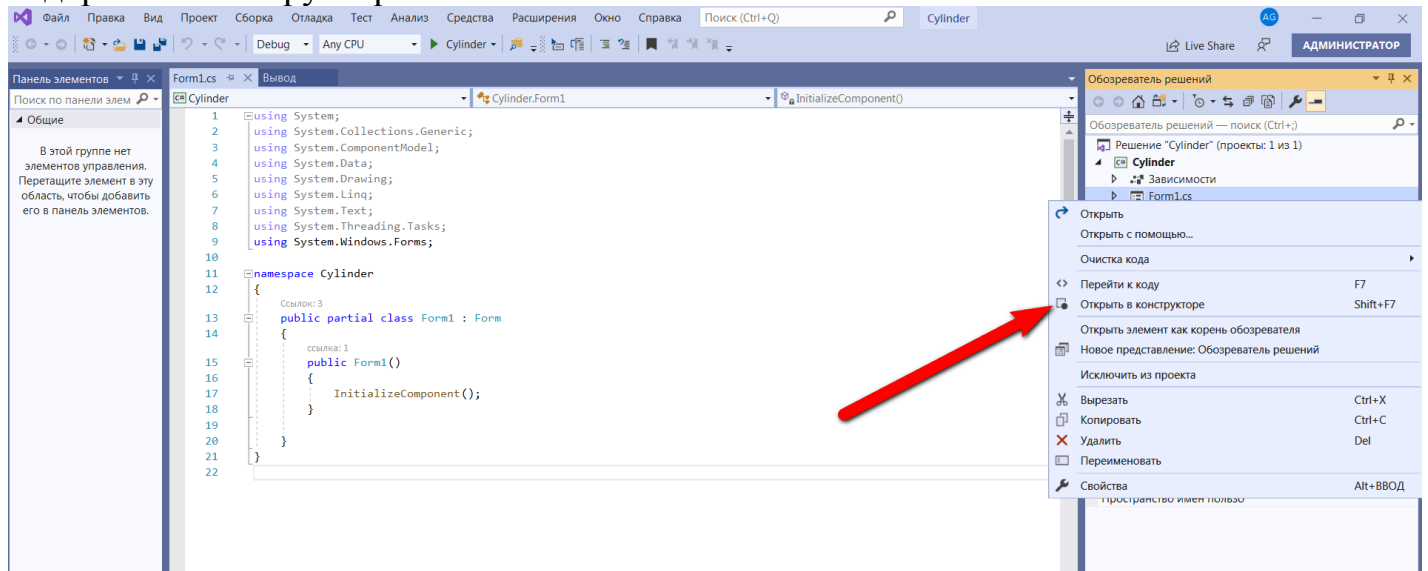
Поместить решение и проект в одном каталоге



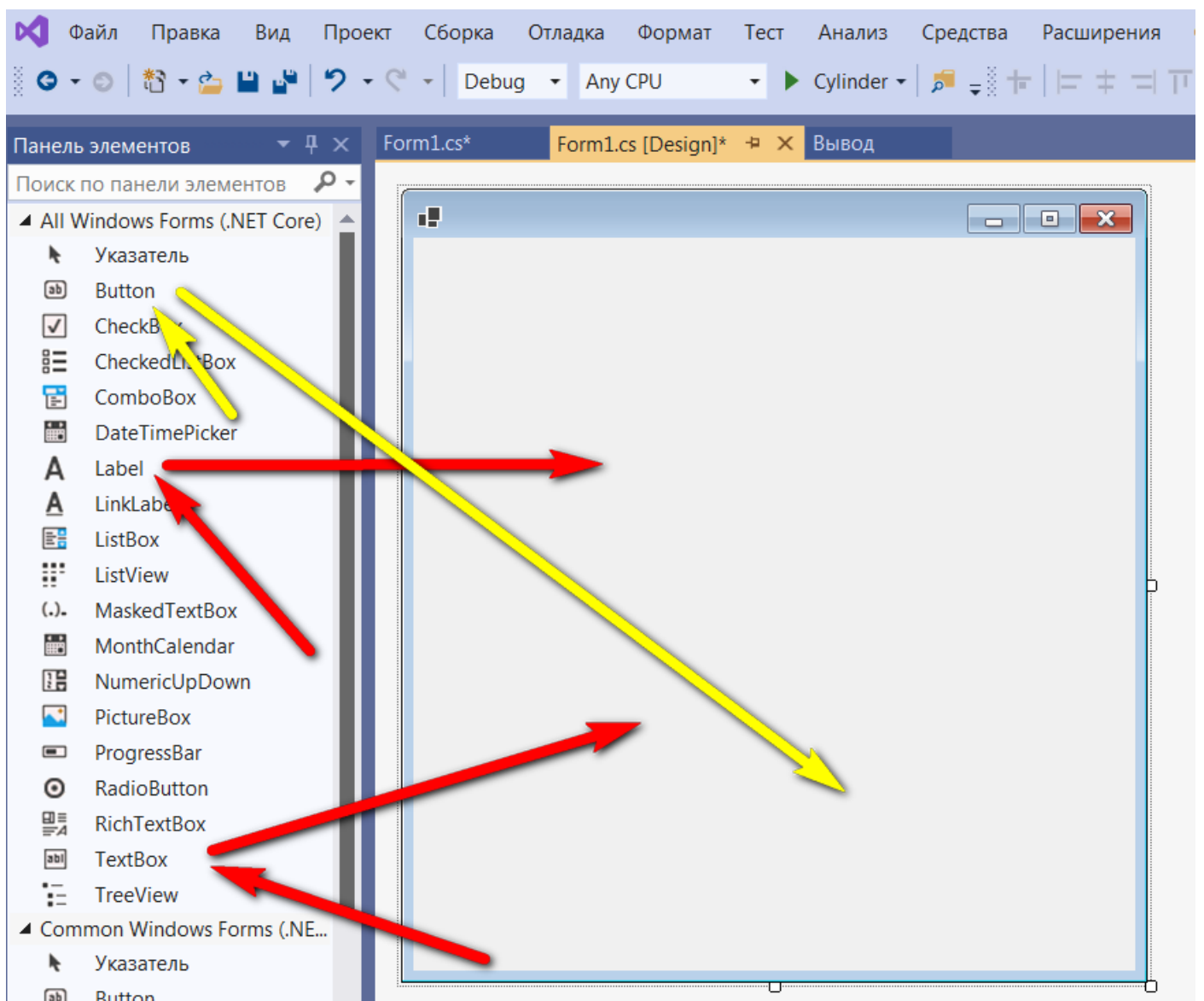
Назад

Создать

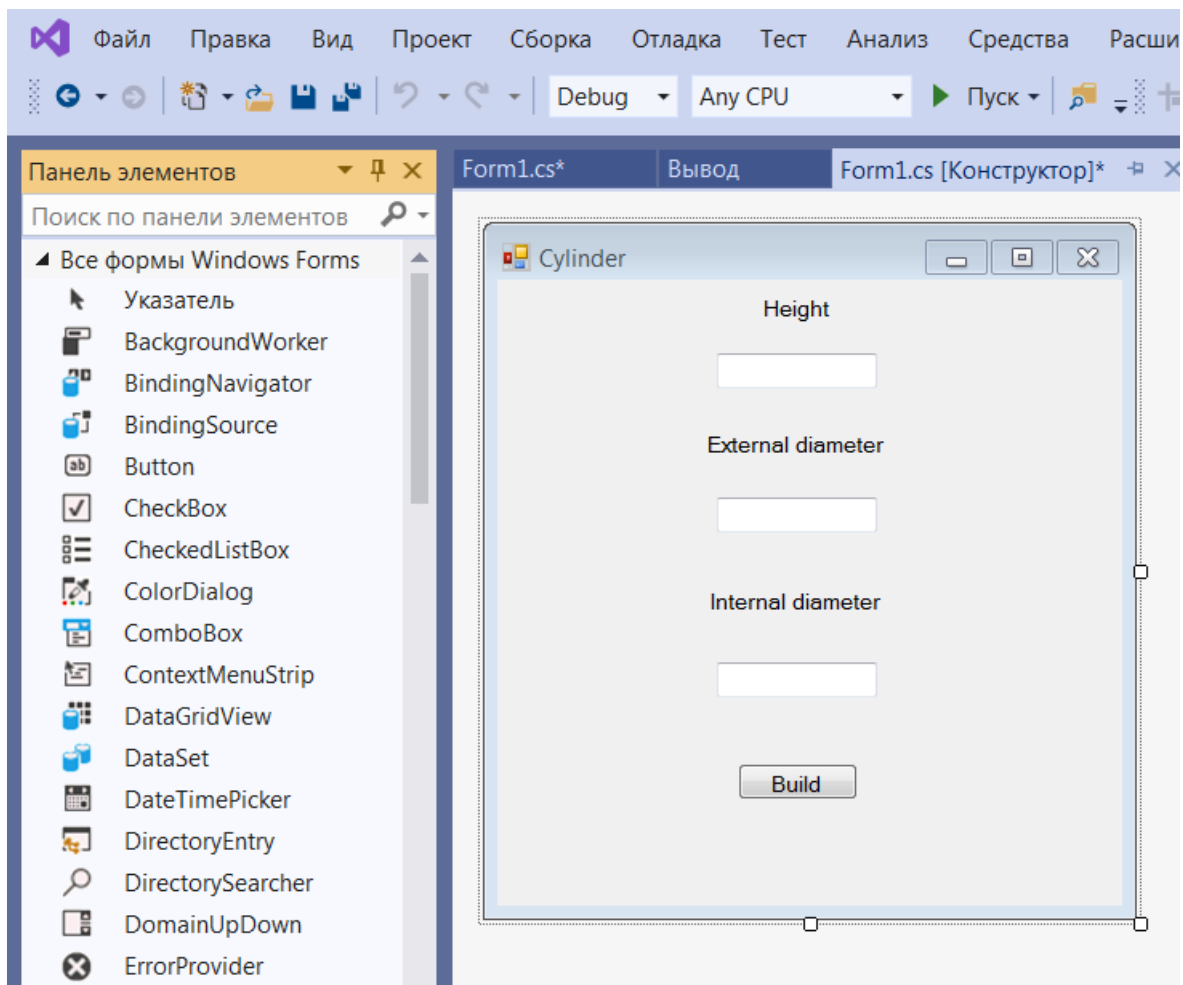
## Відкрити в конструкторі:



З панелі елементів перетягнути 3 елементи Label, 3 елементи TextBox, 1 елемент Button



Текст в элементах Label відобразити наступним чином:



Подвійним кліком в полі TextBox1 генеруємо процедуру обробки і присвоюємо змінній height відповідне значення:

```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    height = double.Parse(textBox1.Text);
}
```

Подвійним кліком в полі TextBox2 генеруємо процедуру обробки і присвоюємо змінній D\_ext відповідне значення:

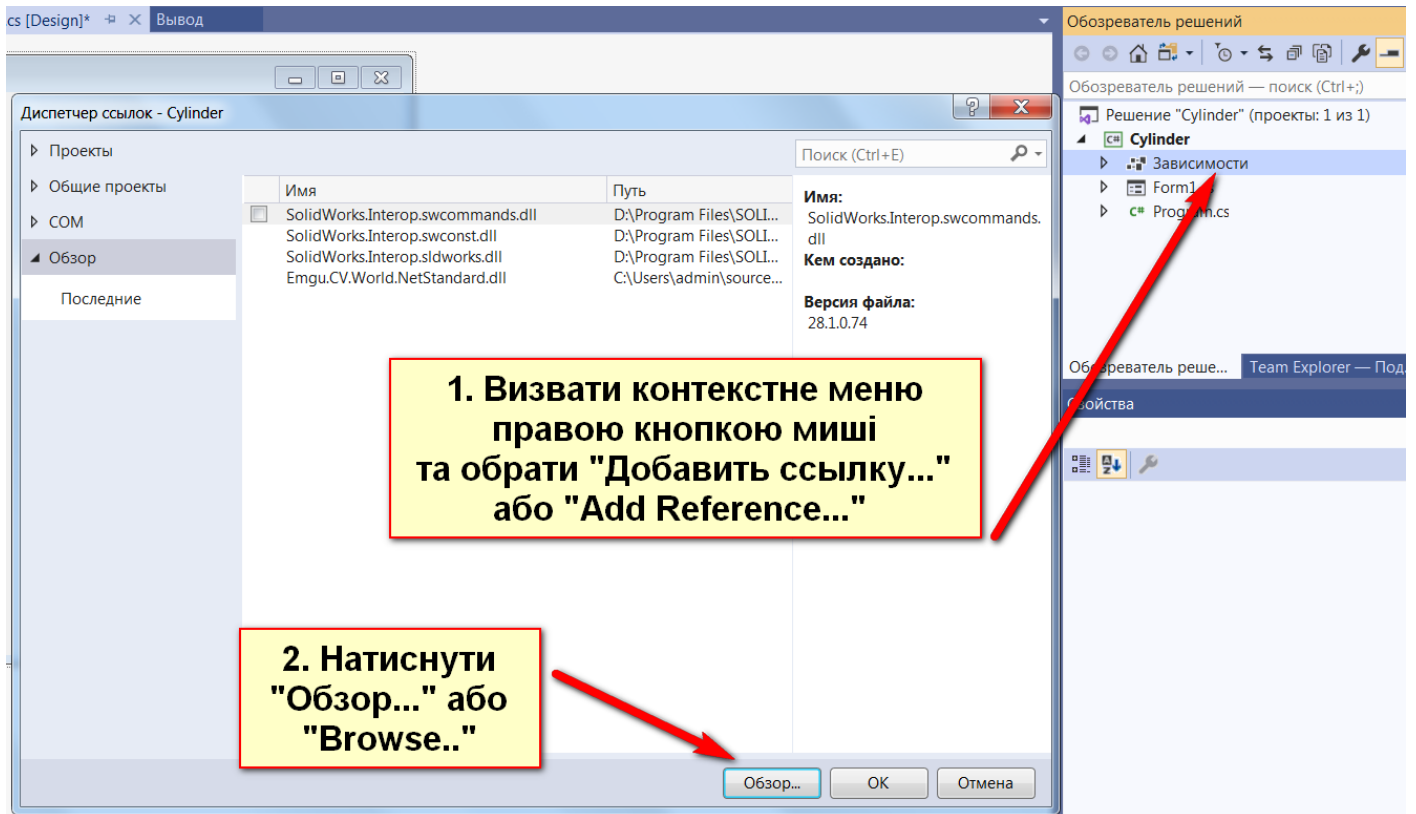
```
private void textBox2_TextChanged(object sender, EventArgs e)
{
    D_ext = double.Parse(textBox2.Text);
}
```

Подвійним кліком в полі TextBox3 генеруємо процедуру обробки і присвоюємо змінній D\_int відповідне значення:

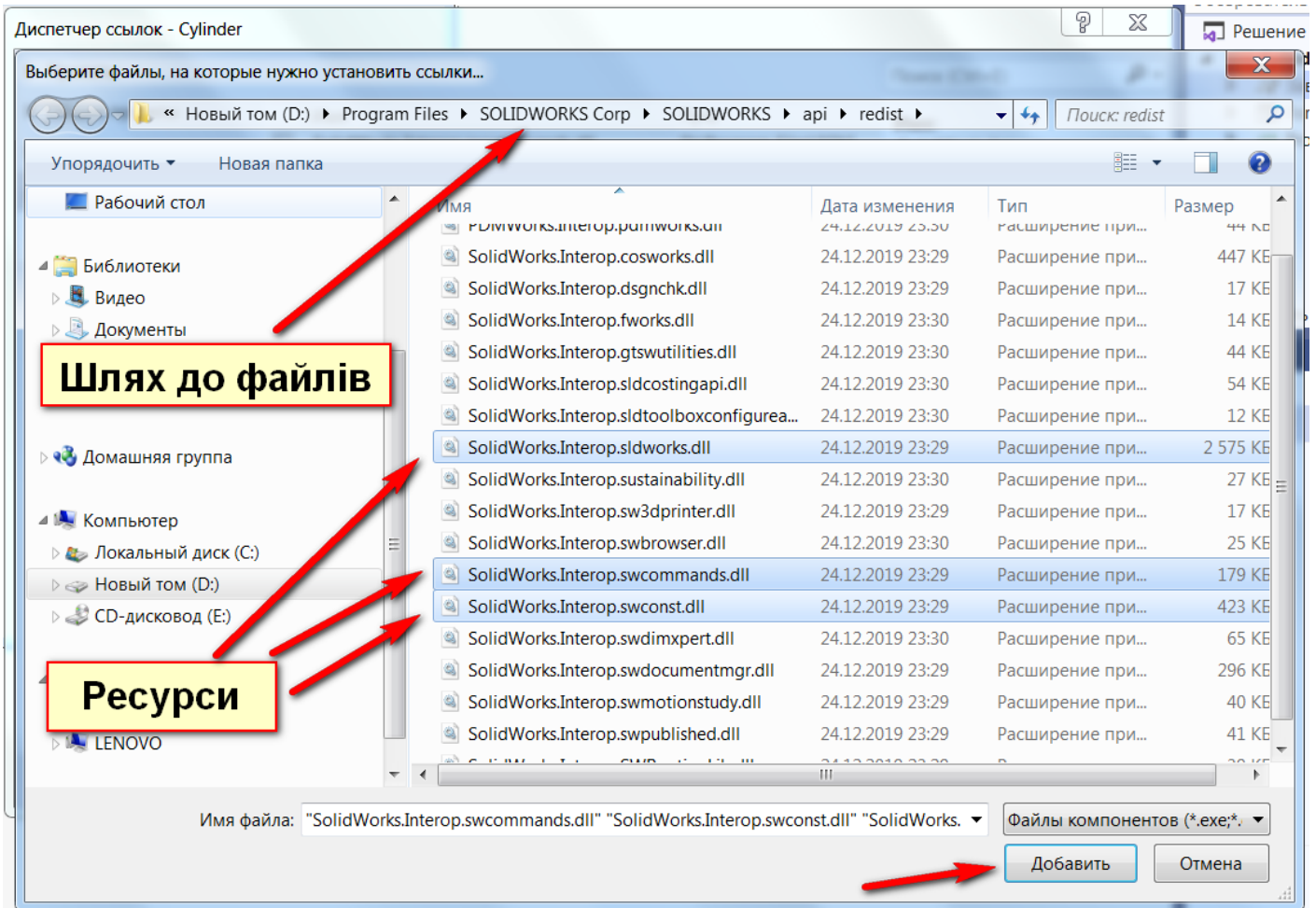
```
private void textBox3_TextChanged(object sender, EventArgs e)
{
    D_int = double.Parse(textBox3.Text);
}
```

Створюємо змінні у кодї програми

```
public partial class Form1 : Form
{
    double height, D_int, D_ext;
    ссылка: 1
    public Form1()
    {
        InitializeComponent();
    }
}
```



Додати необхідні dll-ресурси із папки Solidworks Corp\Solidworks\api\redist



Додати посилання на ресурси в програму

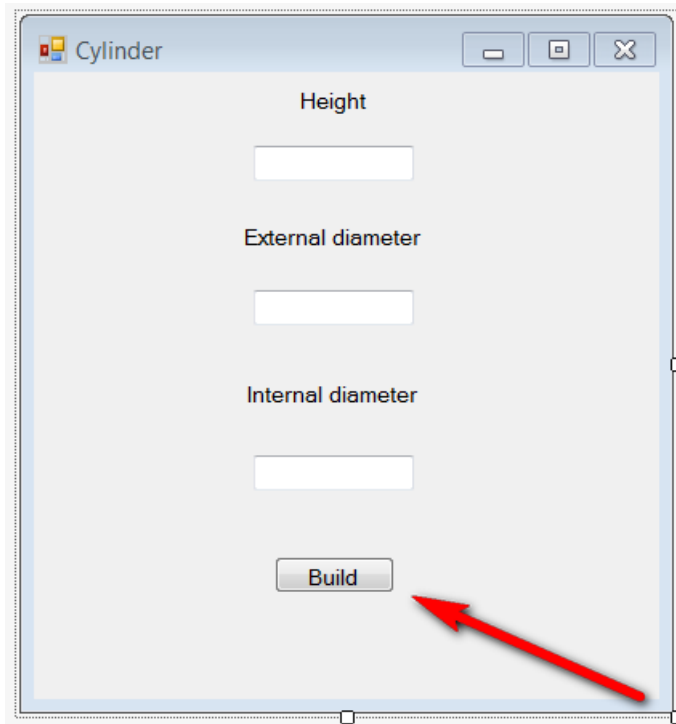
```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;
using SolidWorks.Interop.sldworks;
using SolidWorks.Interop.swcommands;
using SolidWorks.Interop.swconst;
using System.Runtime.InteropServices;

```

Подвійним кліком в конструкторі форми генеруємо процедуру обробки натискання Button:





Додаємо об'єкт Object App SW, закриваємо процеси Solidworks (якщо є), створюємо новий зі станом видимий

```
public partial class Form1 : Form
```

```
{
    double height, D_int, D_ext;
    ссылка: 1
    public Form1()
    {
        InitializeComponent();
    }
}
```

```
SldWorks SwApp; //Object App SW
IModelDoc2 swModel;
```

```
//Build button
```

```
ссылка: 1
private void button1_Click(object sender, EventArgs e)
{
```

```
{
    //Kill Process
    Process[] processes = Process.GetProcessesByName("SLDWORKS");

    foreach (Process process in processes)
    {
        process.CloseMainWindow();
        process.Kill();
    }
}
```

```
object processSW = System.Activator.CreateInstance(System.Type.GetTypeFromProgID("SldWorks.Application"));
```

```
SwApp = (SldWorks)processSW;
SwApp.Visible = true;
```

**Додати об'єкт Solidworks**

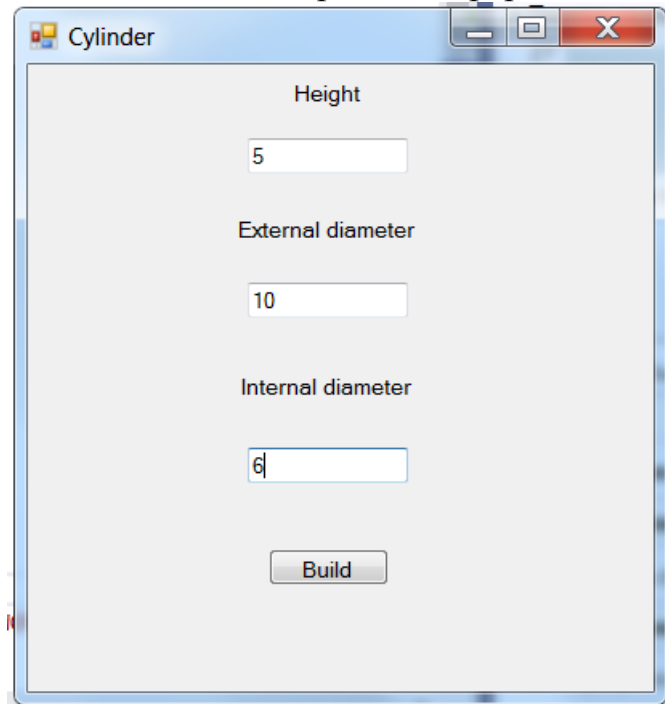
**закрити процес,  
якщо був в пам'яті**

**Створити новий процес Solidworks зі статусом Visible**

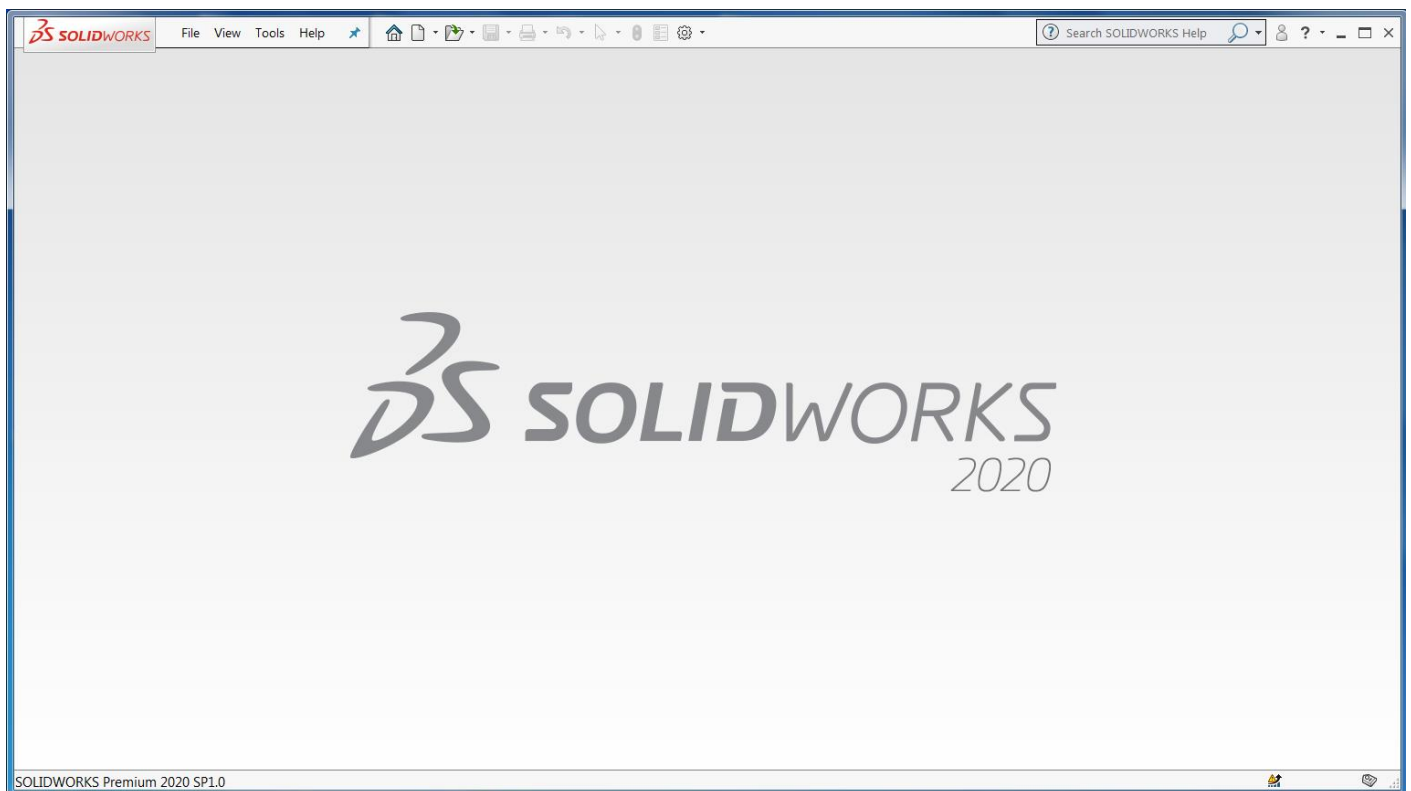


## Запускаємо програму

Результатом на поточний момент є відображення форми. Задаємо довільні значення.



Відбувається запуск програми Solidworks



Побудуємо циліндр в Solidworks

В процедурі обробки події Click Button додаємо наступний код:

```

SwApp.NewPart(); //Створити нову деталь
swModel = SwApp.IActiveDoc2; //Створити 3D документ

//Вибрати вид спереду
swModel.Extension.SelectByID2("Front Plane", "PLANE", 0, 0, 0, false, 0, null, 0);
//Створити ескіз на обраній площині
swModel.SketchManager.InsertSketch(true);

swModel.ClearSelection2(true);
//створити ескіз прямокутника для отримання циліндра
swModel.SketchManager.CreateLine(0, 0, 0, D_ext/2/1000, 0, 0); //Line_1
swModel.SketchManager.CreateLine(D_ext / 2 / 1000, 0, 0, D_ext / 2 / 1000, height/1000, 0); //Line_2
swModel.SketchManager.CreateLine(D_ext / 2 / 1000, height / 1000, 0, 0, height / 1000, 0); //Line_3
swModel.SketchManager.CreateLine(0, height / 1000, 0, 0, 0, 0); //Line_4

swModel.ClearSelection2(true);
// створити вісь обертання
swModel.SketchManager.CreateCenterLine(0, height / 1000, 0, 0, 0, 0); //Axis_Line
swModel.SketchManager.InsertSketch(true);

// побудувати циліндр обертанням
swModel.FeatureManager.FeatureRevolve2(true, true, false, false, false, false, 0, 0, 6.2831853071796,
    0, false, false, 0.01, 0.01, 0, 0, 0, true, true, true);

// створити допоміжну площину Plane1 паралельну площині "Спереди" ("Top Plane")
swModel.Extension.SelectByID2("Top Plane", "PLANE", 0, 0, 0, false, 0, null, 0);
swModel.FeatureManager.InsertRefPlane(8, height / 1000, 0, 0, 0, 0);

//Створити ескіз кола ("Sketch2") на допоміжній площині
swModel.Extension.SelectByID2("Plane1", "PLANE", 0, 0, 0, false, 0, null, 0);
swModel.SketchManager.CreateCircleByRadius(0, 0, height / 1000, D_int / 2 / 1000);
swModel.SketchManager.InsertSketch(true);

// побудувати отвір в циліндрі на повну його висоту
swModel.Extension.SelectByID2("Sketch2", "SKETCH", 0, 0, height / 1000, false, 0, null, 0);
swModel.FeatureManager.FeatureCut3(true, false, false, 0, 0, height / 1000, height / 1000, false, false, false,
    false, 0, 0, false, false, false, false, false, true, true, true, true, false, 0, 0, false);
swModel.ClearSelection2(true);
swModel.ViewZoomtofit();

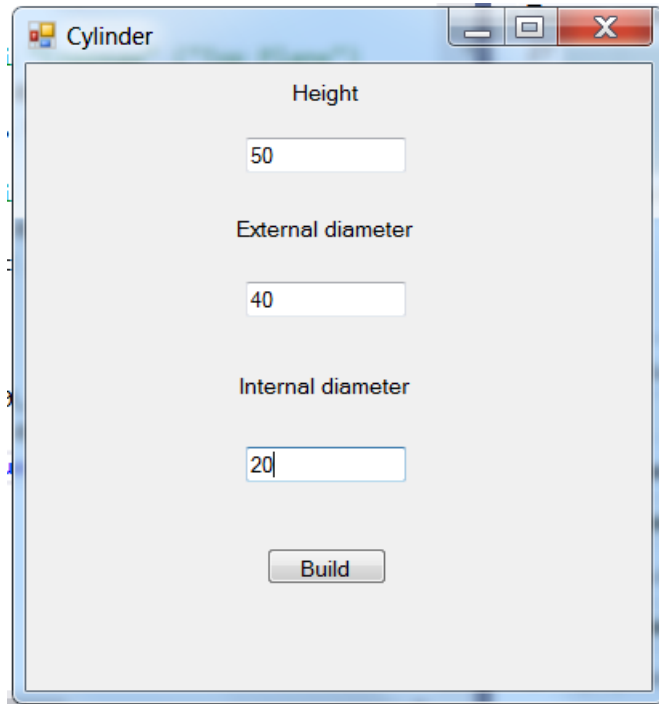
// зберігти файл 3D-моделі. Вказати необхідний шлях
// swModel.SaveAs("c:\\cs\\Cylinder.sldprt");

```

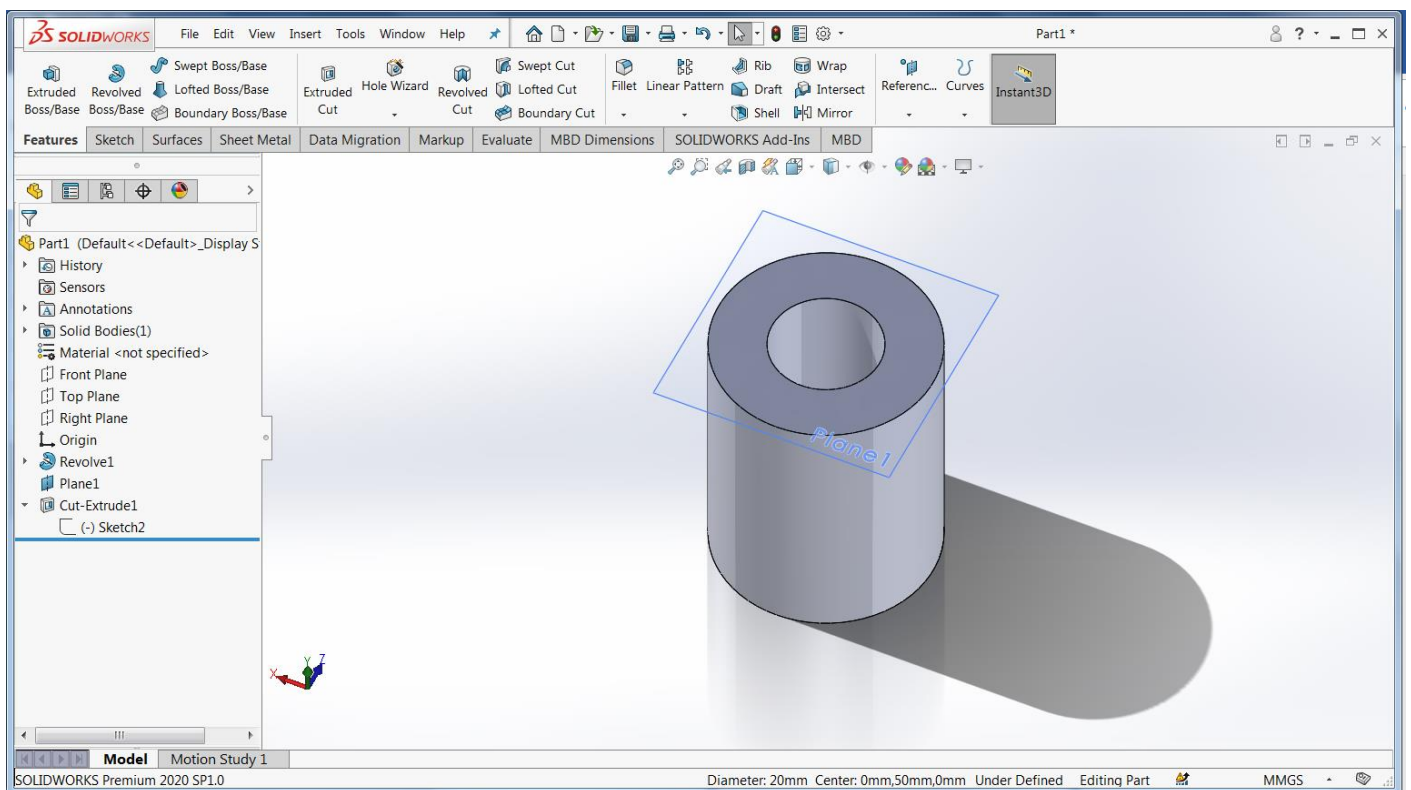
*При необхідності доповнити код програми обробкою виключень (Exception) з використанням try, catch*

Запускаємо програму і будуємо циліндр з наступними параметрами:

- 1) Висота – 50 мм;
- 2) Зовнішній діаметр – 40 мм;
- 3) Внутрішній діаметр – 20 мм



Отримуємо результат:



Кінцевий файл **Form1.cs** в якості прикладу додається.

Виконану 3D-модель та скріншот зображення моделі відіслати на [agromovoy@ztu.edu.ua](mailto:agromovoy@ztu.edu.ua) або в Telegram: [@agromovoy](https://www.instagram.com/agromovoy)