

Практична робота №8

Побудова 3D-моделі деталі типу «Пропелер» з використанням мови програмування C# і Solidworks API

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

Хід роботи

3D-модель, яку необхідно побудувати, – пропелер (рис. 1) з визначеними параметрами (кількістю лопастей, кроком спіралі лопасті та зовнішнім діаметром) за варіантом, який відповідає порядковому номеру у наведеній таблиці 1.

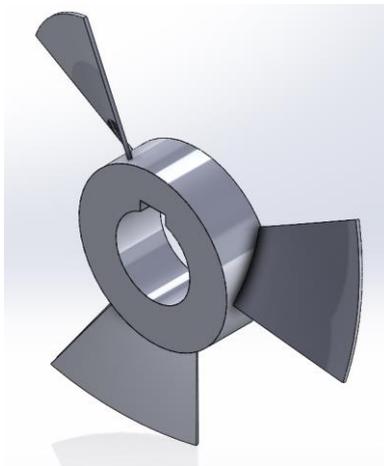


Рис. 1

Таблиця 1.

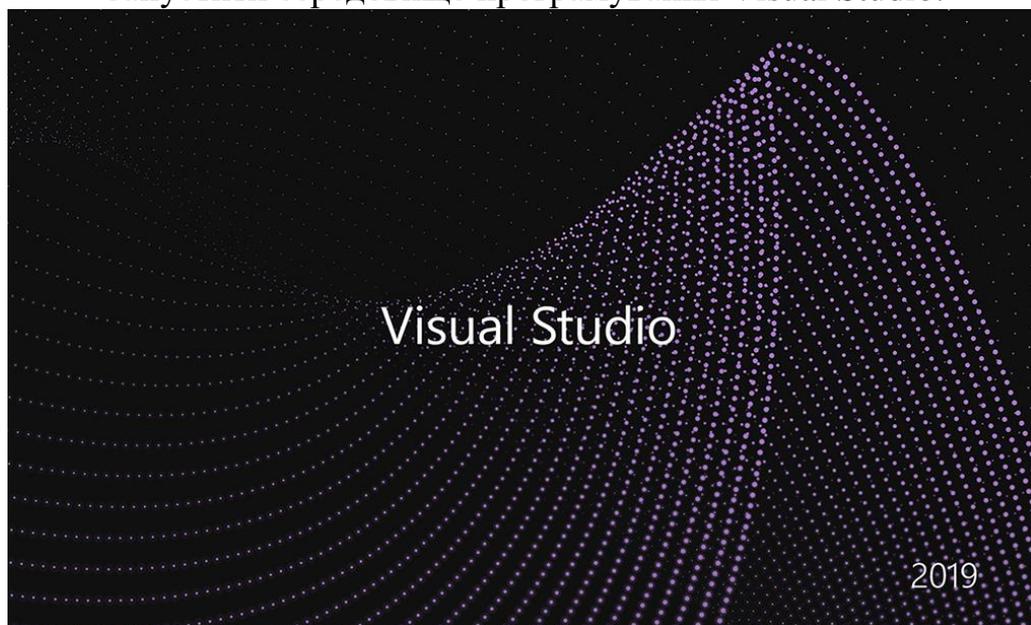
Варіант	Кількість лопастей N	Крок спіралі лопасті SpiralHeight	Зовнішній діаметр Dexternal
1	2	0,07	300
2	3	0,075	280
3	4	0,08	260
4	2	0,085	240
5	3	0,09	220
6	4	0,095	210
7	2	0,1	200
8	3	0,095	190
9	4	0,09	200
10	2	0,085	205
11	3	0,08	215
12	4	0,075	220

13	2	0,07	225
14	3	0,105	230
15	4	0,11	235
16	2	0,115	195
17	3	0,12	205
18	3	0,09	200
19	3	0,085	205
20	3	0,08	215
21	3	0,075	220
22	3	0,07	225
23	3	0,105	230
24	3	0,11	235
25	3	0,115	195

Ознайомитись з відеоінструкцією по практичній роботі за посиланням:
<https://youtu.be/FrYYjQBqBds>

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



Имя проекта - Propeller

Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name

Propeller

Location

D:\

Solution name ?

Propeller

Place solution and project in the same directory

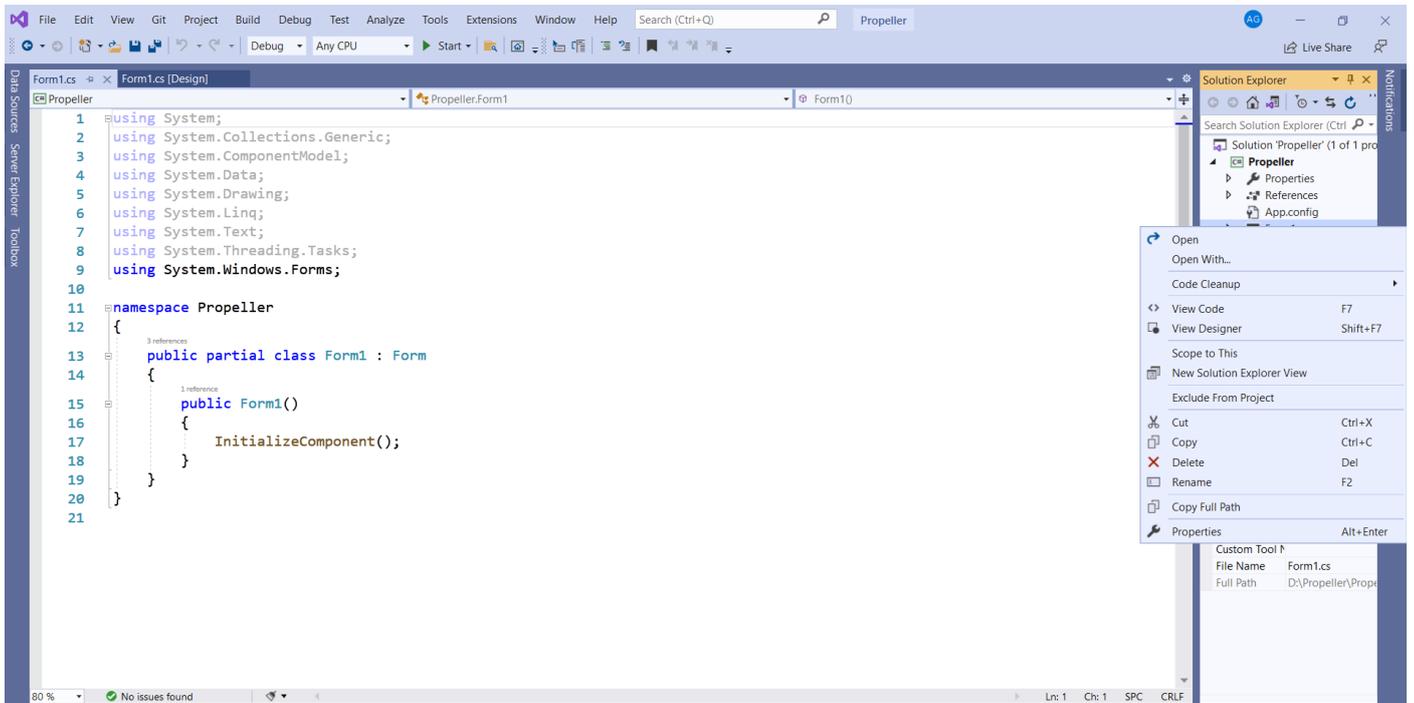
Framework

.NET Framework 4.7.2

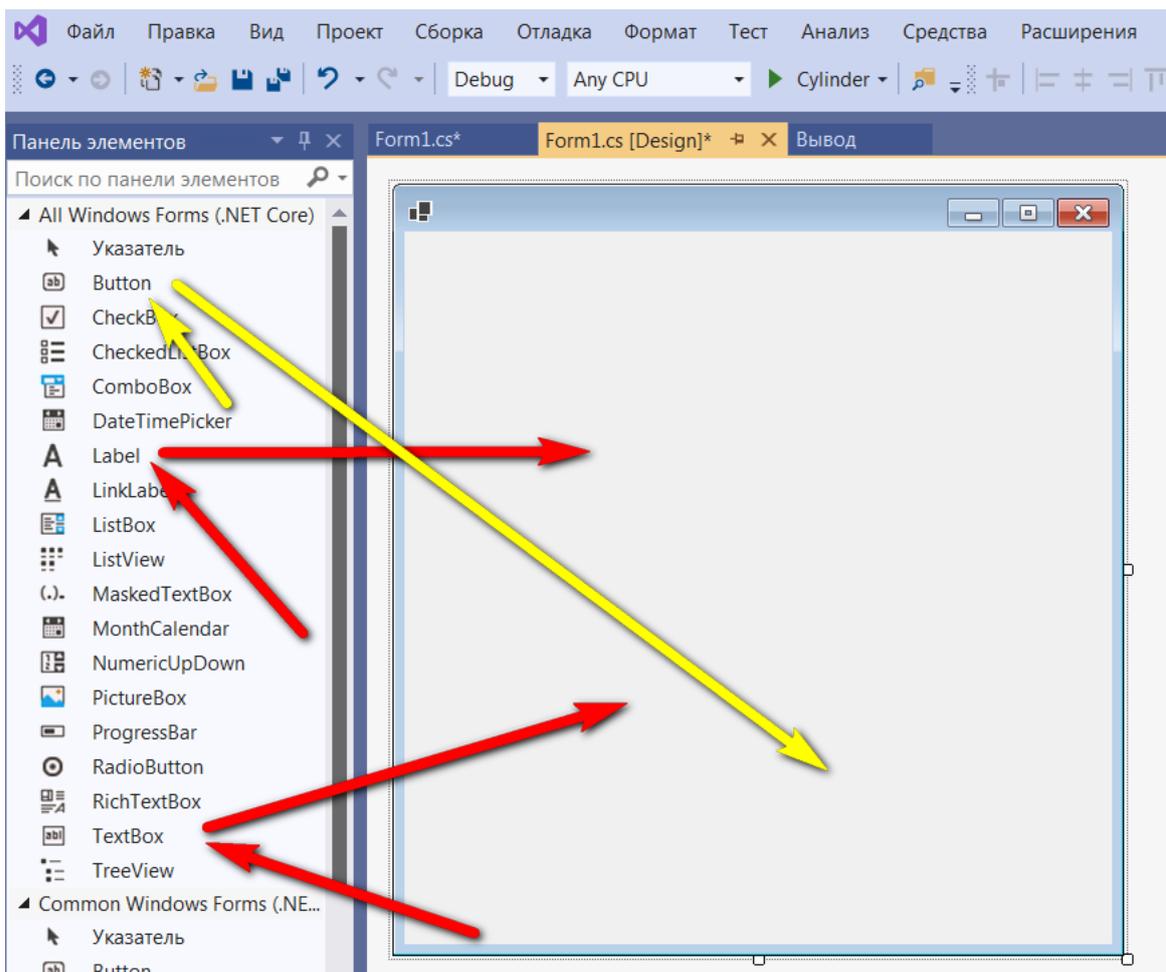
Back

Create

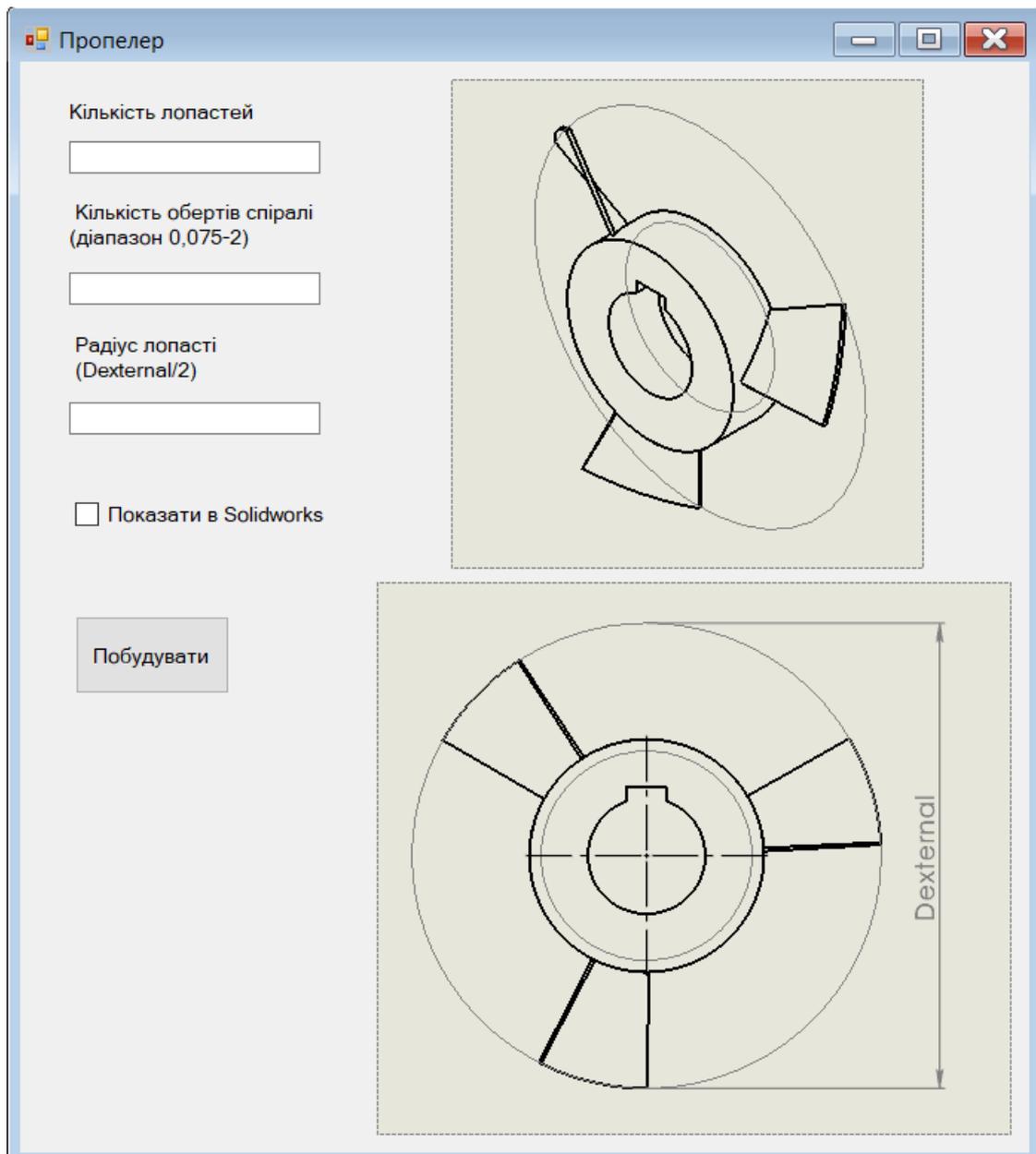
Відкрити в конструкторі:



З панелі елементів перетягнути 3 елементи Label, 3 елементи TextBox, 1 елемент Button, 1 елемент CheckBox, 2 елементи PictureBox



Форму відобразити наступним чином (в елементи PictureBox вставити додані до роботи 2 зображення):



Подвійним кліком в полі TextBox1 генеруємо процедуру обробки і присвоюємо змінній N (кількість лопастей) відповідне значення:

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

Подвійним кліком в полі TextBox2 генеруємо процедуру обробки і присвоюємо змінній SpiralHeight відповідне значення:

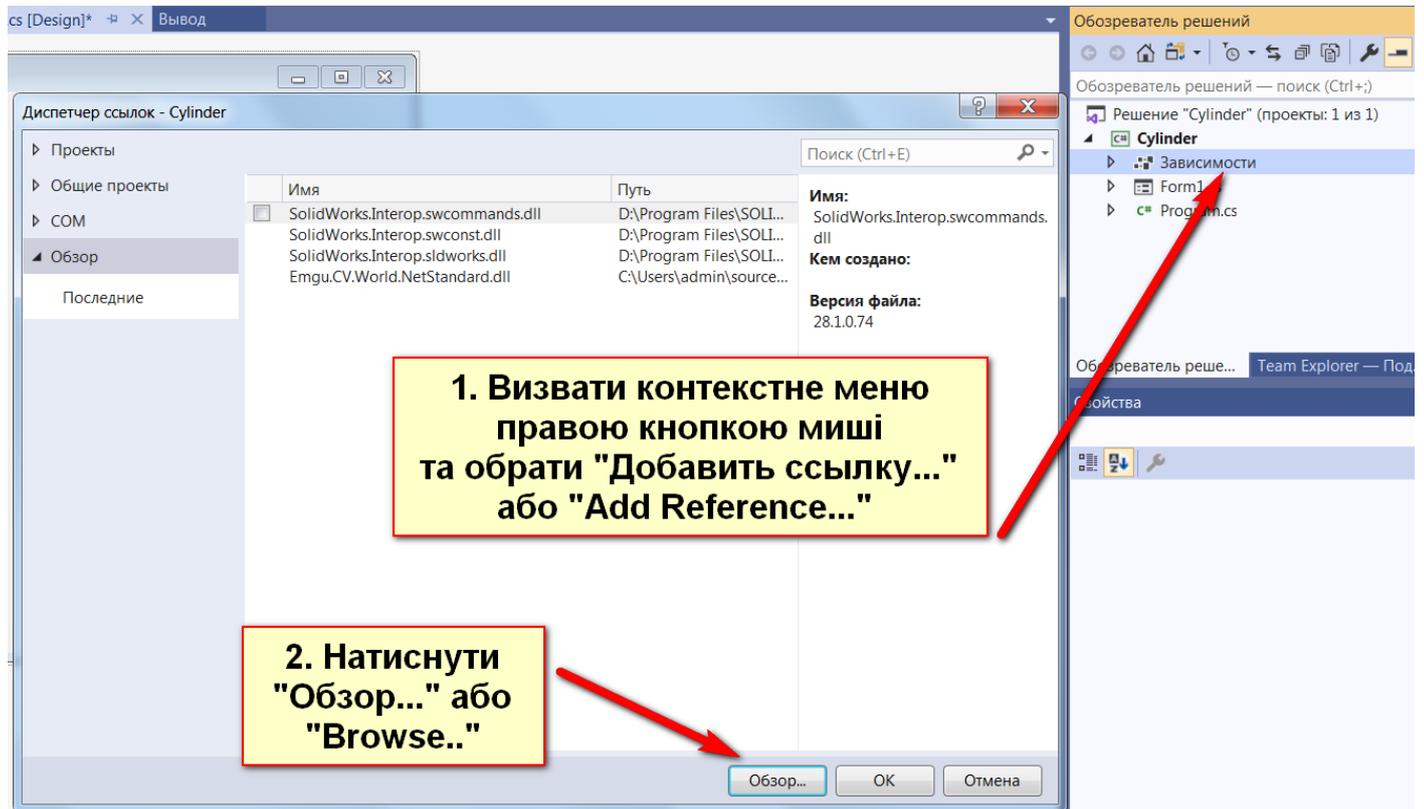
```
private void textBox2_TextChanged(object sender, EventArgs e)
{
    SpiralHeight = double.Parse(textBox2.Text);
}
```

Подвійним кліком в полі TextBox3 генеруємо процедуру обробки і присвоюємо змінній Dexternal відповідне значення:

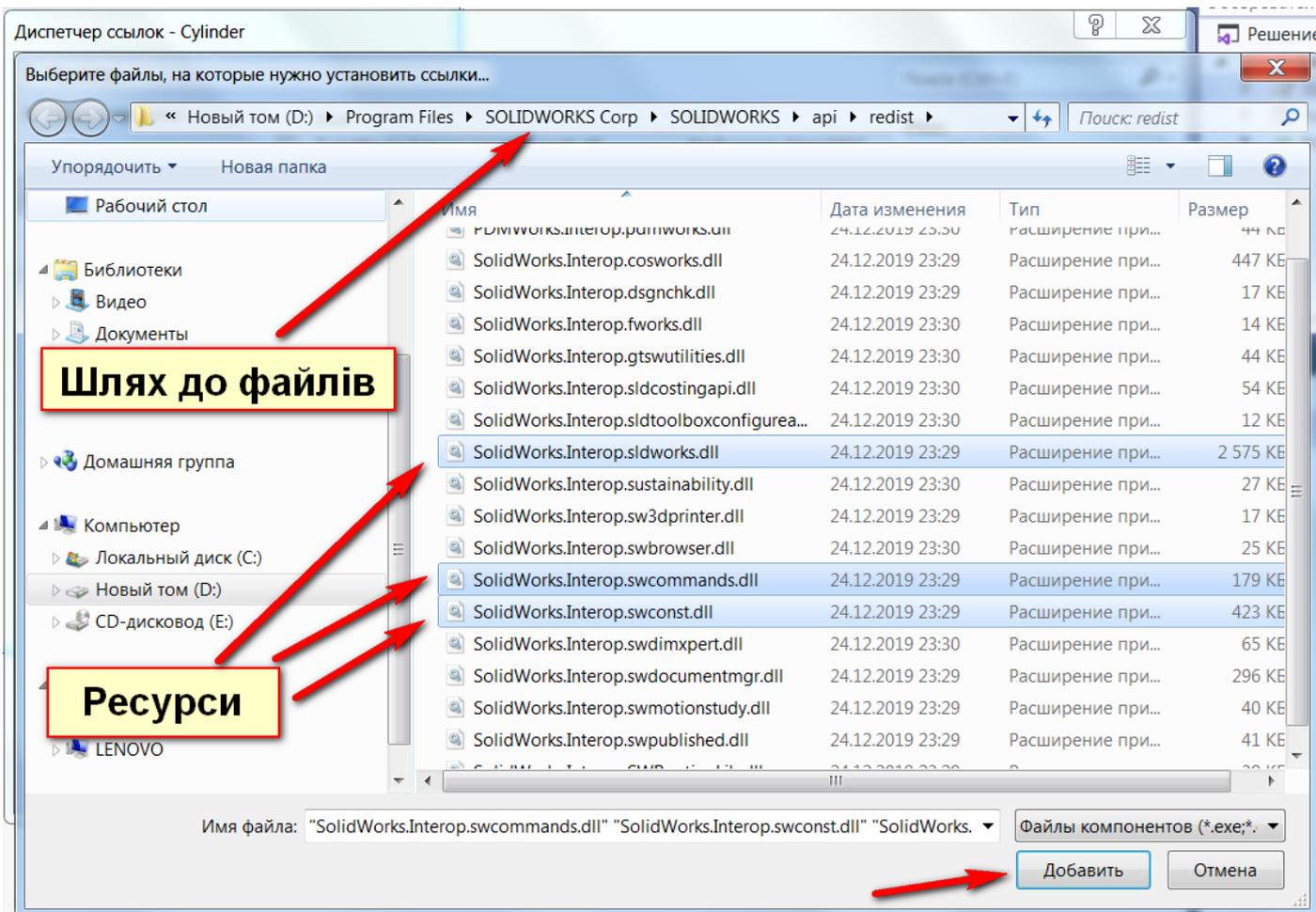
```
private void textBox3_TextChanged(object sender, EventArgs e)
{
    Dexternal = double.Parse(textBox3.Text) / 1000;
}
```

Створюємо змінні у кодї програми

```
public partial class Form1 : Form
{
    double SpiralHeight, Dexternal;
    int N;
    SldWorks swApp;
    IModelDoc2 swModel;
```



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 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

Побудуємо пропелер в Solidworks

Використаємо програмний код наведений в додатку 1.

Запускаємо програму і будуємо пропелер з наступними параметрами:

- 1) Кількість лопастей – N=4;
- 2) Крок спіралі лопасті SpiralHeight = 0,1;
- 3) Зовнішній діаметр Dexternal=250 мм.

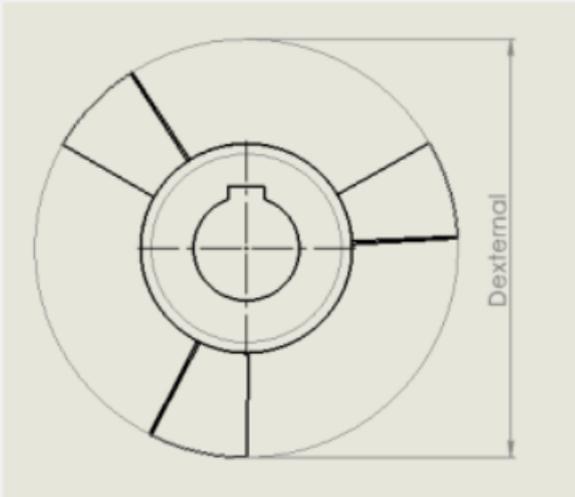
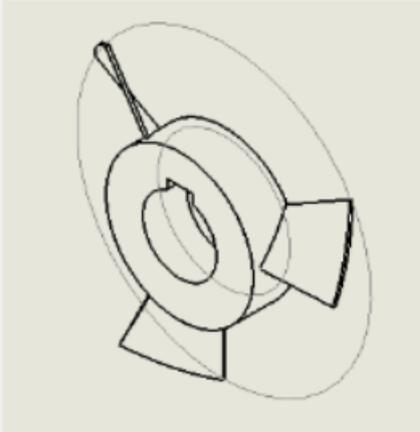
Кількість лопастей

Кількість обертів спіралі
(діапазон 0,075-2)

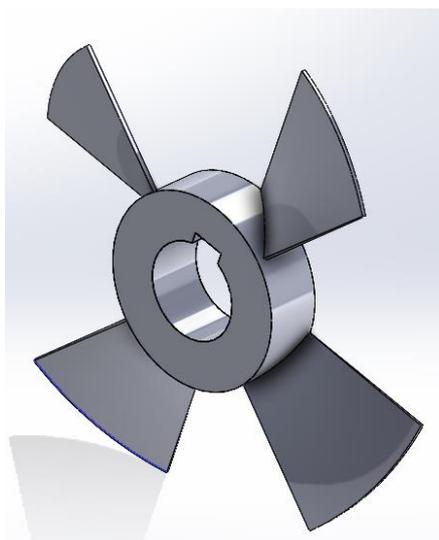
Радіус лопасті
(Dexternal/2)

Показати в Solidworks

Побудувати



Отримуємо результат:



Кінцевий файл [Form1.cs](#) в якості прикладу додається.

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

namespace CsharpSolidworks_Propeller
{
    public partial class Form1 : Form
    {
        double SpiralHeight, Dexternal;
        int N;
        SldWorks swApp;
        IModelDoc2 swModel;
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            N = int.Parse(textBox1.Text);
        }

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

        private void textBox3_TextChanged(object sender, EventArgs e)
        {
            Dexternal = double.Parse(textBox3.Text) / 1000;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            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;
if (checkBox1.Checked == false) { swApp.Visible = false; }
else
swApp.Visible = true;

swApp.NewPart();
swModel = swApp.IActiveDoc2;

ModelDoc2 swDoc = null;
bool boolstatus;

swDoc = ((ModelDoc2)(swApp.ActiveDoc));
// максимальній екран
ModelView myModelView = null;
myModelView = ((ModelView)(swDoc.ActiveView));
myModelView.FrameState = ((int)(swWindowState_e.swWindowMaximized));

boolstatus = swDoc.Extension.SelectByID2("Front Plane", "PLANE", 0, 0, 0, false, 0, null, 0);
swDoc.ClearSelection2(true);
SketchSegment skSegment = null;
skSegment = ((SketchSegment)(swDoc.SketchManager.CreateCircle(0.000000, 0.000000, 0.000000,
0, 0.1, 0.000000)));
swDoc.ClearSelection2(true);
skSegment = ((SketchSegment)(swDoc.SketchManager.CreateCircle(0.000000, 0.000000, 0.000000,
0, 0.05, 0.000000)));

swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Arc2", "SKETCHSEGMENT", -
0.06101280800821357, 0.020614681724845998, 0, false, 0, null, 0);

swDoc.ClearSelection2(true);
Dimension myDimension = null;
myDimension = ((Dimension)(swDoc.Parameter("D1@Sketch1")));
swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Arc1", "SKETCHSEGMENT", -
0.030090785420944566, 0.010639835728952773, 0, false, 0, null, 0);
swDoc.ClearSelection2(true);
myDimension = ((Dimension)(swDoc.Parameter("D2@Sketch1")));

swDoc.ClearSelection2(true);
swDoc.SketchManager.InsertSketch(true);
boolstatus = swDoc.Extension.SelectByID2("Arc2@Sketch1", "EXTSKETCHSEGMENT", -
0.041444410815605737, 0.027970713472260642, 0, false, 0, null, 0);
swDoc.ShowNamedView2("*Триметрия", 8);
swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Arc2@Sketch1", "EXTSKETCHSEGMENT", -
0.041444410815605737, 0.027970713472260642, 0, false, 0, null, 0);
Feature myFeature = null;
myFeature = ((Feature)(swDoc.FeatureManager.FeatureExtrusion2(true, false, false, 0, 0,
0.059999999999999998, 0.01, false, false, false, false, 0.017453292519943334, 0.017453292519943334,
false, false, false, false, true, true, true, 0, 0, false)));
swDoc.ISelectionManager.EnableContourSelection = false;

```

```

boolstatus = swDoc.Extension.SelectByID2("", "FACE", -0.018214823767152666,
0.024821100453984002, 0, false, 0, null, 0);

boolstatus = swDoc.Extension.SelectByID2("Front Plane", "PLANE", 0, 0, 0, true, 0, null, 0);

RefPlane myRefPlane = null;
myRefPlane = ((RefPlane)(swDoc.FeatureManager.InsertRefPlane(8, 0.005, 0, 0, 0, 0)));
swDoc.ClearSelection2(true);

boolstatus = swDoc.Extension.SelectByID2("Plane4", "PLANE", 0, 0, 0.005, false, 0, null, 0);
swDoc.ClearSelection2(false);

boolstatus = swDoc.Extension.SelectByID2("Front Plane", "PLANE", 0, 0, 0, false, 0, null, 0);
swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Plane4", "PLANE", 0, 0, 0, false, 0, null, 0);
skSegment = ((SketchSegment)(swDoc.CreateCircleByRadius2(0, 0, 0.005, 0.09)));
swDoc.ClearSelection2(true);

swDoc.ClearSelection2(true);
swDoc.SketchManager.InsertSketch(true);

swDoc.ClearSelection2(true);
myDimension = ((Dimension)(swDoc.Parameter("D1 @Sketch2")));
swDoc.ClearSelection2(true);
swDoc.SketchManager.InsertSketch(true);

boolstatus = swDoc.Extension.SelectByID2("Arc1 @Sketch2", "EXTSKETCHSEGMENT", -
0.0051561193844902927, -0.044703628855976382, 0, false, 0, null, 0);
swDoc.InsertHelix(false, true, false, true, 1, 0.050000000000000003, 0.20000000000000001,
SpiralHeight, 0, 4.7123889803847003);

boolstatus = swDoc.Extension.SelectByID2("Front Plane", "PLANE", 0, 0, 0, false, 0, null, 0);
swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Plane4", "PLANE", 0, 0, 0, false, 0, null, 0);
skSegment = ((SketchSegment)(swDoc.CreateCircleByRadius2(0, 0, 0.005, Dexternal)));
swDoc.ClearSelection2(true);

swDoc.ClearSelection2(true);
swDoc.SketchManager.InsertSketch(true);

boolstatus = swDoc.Extension.SelectByID2("Arc1", "SKETCHSEGMENT",
0.068232375085557836, 0.0033880903490759728, 0.0300000000000000016, false, 0, null, 0);
swDoc.ClearSelection2(true);
myDimension = ((Dimension)(swDoc.Parameter("D1 @Sketch3")));
swDoc.ClearSelection2(true);

myModelView.RotateAboutCenter(0.0042372155250729477, 0.0095337349314141336);
boolstatus = swDoc.Extension.SelectByID2("Sketch3", "SKETCH", 0, 0, 0, false, 0, null, 0);

myModelView.RotateAboutCenter(0, -0.0095337349314141336);
swDoc.InsertHelix(false, true, false, true, 1, 0.050000000000000003, Dexternal, SpiralHeight, 0,
4.7123889803847003);

```

```

swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Helix2", "REFERENCECURVES",
0.14704918567628056, -0.17354049988603037, 0.10539079158644427, true, 0, null, 0);
boolstatus = swDoc.Extension.SelectByID2("Helix1", "REFERENCECURVES",
0.11907217453148178, -0.1254258036739202, 0.10509121097157959, true, 0, null, 0);

swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Helix/Spiral2", "REFERENCECURVES",
0.10000000000000003, 0, 0.055000000000000007, false, 8193, null, 0);
boolstatus = swDoc.Extension.SelectByID2("Helix/Spiral1", "REFERENCECURVES",
0.044999999999999998, 0, 0.054999999999999993, true, 16385, null, 0);
myFeature = ((Feature)(swDoc.FeatureManager.SetNetBlendCurveData(0, 0, 0, 0, 1, true)));
myFeature = ((Feature)(swDoc.FeatureManager.SetNetBlendCurveData(0, 1, 0, 0, 1, true)));
myFeature = ((Feature)(swDoc.FeatureManager.SetNetBlendDirectionData(0, 32, 0, false, false)));
myFeature = ((Feature)(swDoc.FeatureManager.SetNetBlendDirectionData(1, 32, 0, false, false)));
myFeature = ((Feature)(swDoc.FeatureManager.InsertNetBlend(2, 2, 0, false, 0.0001, false, true,
true, true, false, -1, -1, false, -1, false, false, -1, false, -1, false)));

swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Boundary-Surface1", "BODYFEATURE", 0, 0, 0,
false, 1, null, 0);
myFeature = ((Feature)(swDoc.FeatureManager.FeatureBossThicken(0.002, 2, 71763003, false, true,
true, true)));

boolstatus = swDoc.DeSelectByID("Thicken1", "BODYFEATURE", 0, 0, 0);
boolstatus = swDoc.Extension.SelectByID2("Thicken1", "BODYFEATURE", 0, 0, 0, false, 0, null,
0);
boolstatus = swDoc.DeSelectByID("Thicken1", "BODYFEATURE", 0, 0, 0);
boolstatus = swDoc.Extension.SelectByID2("", "EDGE", -0.00042803633093058124, -
0.099989966108523731, 0.0061186712590313164, true, 1, null, 0);

swDoc = ((ModelDoc2)(swApp.ActiveDoc));

// Create an axis
boolstatus = swDoc.Extension.SelectByID2("", "FACE", 0, 0, 0, false, 0, null, 0);
boolstatus = swDoc.InsertAxis2(true);

boolstatus = swDoc.Extension.SelectByID2("Thicken1", "BODYFEATURE", 0, 0, 0, false, 0, null,
0);
boolstatus = swDoc.Extension.SelectByID2("Axis1", "AXIS", 0, 0, 0, true, 1, null, 0);
boolstatus = swDoc.Extension.SelectByID2("Thicken1", "SOLIDBODY", 0, 0, 0, true, 256, null, 0);

myFeature = ((Feature)swDoc.FeatureManager.FeatureCircularPattern3(N, 6.2831853071796, false,
"NULL", false, true));

swDoc.ClearSelection2(true);

swDoc = ((ModelDoc2)(swApp.ActiveDoc));
boolstatus = swDoc.Extension.SelectByID2("", "FACE", -0.065929145949858992, -
0.0054982073460507763, 0, false, 0, null, 0);
boolstatus = swDoc.Extension.SelectByID2("", "EDGE", -0.00080765673698740548,
0.050764068059945572, 0, false, 0, null, 0);
swDoc.SketchManager.InsertSketch(true);
swDoc.SetPickMode();

```

```

swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Plane4", "PLANE", -0.16246942817054327,
0.062041763732596802, 0, false, 0, null, 0);
swDoc.SketchManager.InsertSketch(true);
swDoc.ClearSelection2(true);
swDoc.SetPickMode();
swDoc.SetPickMode();
swDoc.SketchManager.InsertSketch(true);
swDoc.EditUndo2(1);
swDoc = ((ModelDoc2)(swApp.ActiveDoc));
swDoc.SketchManager.InsertSketch(true);
swDoc.ClearSelection2(true);
Array vSkLines = null;
vSkLines = ((Array)(swDoc.SketchManager.CreateCenterRectangle(0, 0.050000000000000003, 0, -
0.017026009532228381, 0.040330834077929673, 0)));
swDoc.ClearSelection2(true);
swDoc.SketchManager.InsertSketch(true);

swDoc.ClearSelection2(true);
boolstatus = swDoc.Extension.SelectByID2("Sketch5", "SKETCH", 0, 0, 0, false, 0, null, 0);
// Feature myFeature = null;
myFeature = ((Feature)(swDoc.FeatureManager.FeatureCut3(true, false, true, 1, 0, 0.01, 0.01, false,
false, false, false, 0.017453292519943334, 0.017453292519943334, false, false, false, false, true, true,
true, true, false, 0, 0, false)));
swDoc.ISelectionManager.EnableContourSelection = false;

Close();
if (checkBox1.Checked == false) { MessageBox.Show("Побудову моделі завершено"); }

// тут вкажіть реальні папку на диску для збереження файлу моделі
swModel.SaveAs("d:\\magistr2021_app.sldprt");
Application.Exit();
}
}
}

```