VS Codeで作るSolidWorksの自動化

1. VS Codeのダウンロード

<https://www602.math.ryukoku.ac.jp/Prog1/vscode-win.html>の通りにダウンロードする。

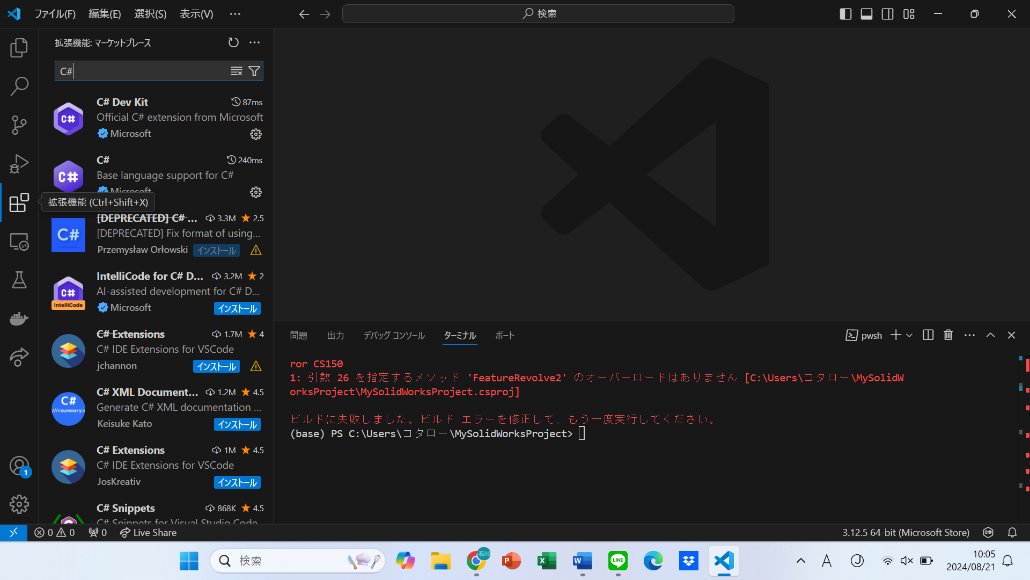
1. SolidWorksのダウンロード

Dropboxにあるライセンスの資料から、ダウンロードする。

その後SolidWorksを開いて、新規から部品ファイルを選択して保存場所を指定して保存する。保存場所は「C:\SW\」でも可。

1. VS Codeの拡張機能のダウンロード

「Extensions」ボタンを クリックして、「Extensions Marketplace」を表示します。



その後以下の拡張機能をダウンロードする。

* **C#** (Microsoft製)
* .NET Install Tool for Extension Authors
* .NET Core Test Explorer
* C# Extensions
* NuGet Package Manager
* Debugger for Unity

1. .NET SDKのインストール

SolidWorksAPIを使うために.NET SDKを[https://dotnet.microsoft.com/ja-jp/download](VS%20Codeで作るSolidWorksの自動化.docx)をからダウンロードする。

.NET 8.0をダウンロードする。

1. SolidWorksAPIを使用するための準備

SolidWorksがインストールされているディレクトリから以下の物をコピーします。

場所はC\Program Files\SOLIDWORKS Corp\SOLIDWORKS内にある

* SolidWorks.Interop.sldworks.dll
* SolidWorks.Interop.swconst.dll

これらを後ほど説明する「libs」ファイルに入れる。

1. ディレクトリ構造の整理

ディレクトリ構造を以下のようにする。

C

└──SW

└──Part1.SLDPRT

└──MySolidWorksProject

├── bin

│ └── Debug

│ └── net8.0

├── libs

│ ├── SolidWorks.Interop.sldworks.dll

│ └── SolidWorks.Interop.swconst.dll

├── obj

│ └── Debug

│ └── net8.0

├── MySolidWorksProject.csproj

└── Program.cs

1. C#プロジェクトの作成

新規フォルダを作成する。今回は「My MySolidWorksProject」とする。

「Ctrl」＋「Shift」＋「@」を押して「新しいターミナル」を開く。

ターミナルから、以下のコードを実行します。

dotnet new console -n MySolidWorksProject

プロジェクトにSolidWorks APIのDLLを追加するために、MySolidWorksProject.csprojファイルを編集し、以下の内容に変更します。

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net8.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<Reference Include="SolidWorks.Interop.sldworks">

<HintPath>libs\SolidWorks.Interop.sldworks.dll</HintPath>

</Reference>

<Reference Include="SolidWorks.Interop.swconst">

<HintPath>libs\SolidWorks.Interop.swconst.dll</HintPath>

</Reference>

</ItemGroup>

</Project>

1. SolidWorksAPIの作成

Program.csのファイルをVS Codeの新規ファイルから作り、以下のサンプルコードを打つ。(ここに自身が自動化させたいものをプログラムする)

using SolidWorks.Interop.sldworks;

using SolidWorks.Interop.swconst;

using System;

namespace MySolidWorksProject

{

class Program

{

static void Main(string[] args)

{

// SolidWorksアプリケーションを起動

SldWorks swApp = new SldWorks();

ModelDoc2 swModel;

PartDoc swPart;

bool boolstatus;

int longstatus = 0;

// SolidWorksを起動し、ウィンドウを表示

swApp.Visible = true;

// 新しいパーツを作成

swModel = swApp.NewPart();

swPart = (PartDoc)swModel;

// スケッチの開始

swModel.SketchManager.InsertSketch(true);

// スケッチに長方形を描画

swModel.SketchManager.CreateCenterRectangle(0, 0, 0, 0.05, 0.05, 0);

// スケッチを終了

swModel.SketchManager.InsertSketch(true);

// スケッチを押し出し

FeatureManager swFeatureMgr = swModel.FeatureManager;

swFeatureMgr.FeatureExtrusion2(true, false, false, 0, 0, 0.01, 0.01, false, false, false, false, 0, 0, false, false, false, false, true, false, false, false, false, 0, 0, out longstatus);

// パーツを指定の場所に保存

string savePath = @"先ほど作った新規の部品ファイルの保存先を指定。同一ディレクトリにC:\SW\MySolidWorksPart.SLDPRTでも可 ";

boolstatus = swModel.SaveAs(savePath);

if (boolstatus)

{

Console.WriteLine("SolidWorksモデルが作成され、保存されました！");

}

else

{

Console.WriteLine("モデルの保存に失敗しました。");

}

Console.ReadKey();

}

}

}

1. プログラムのbuildとrun

ターミナルで以下のコマンドを実行する

dotnet build MySolidWorksProject.csproj

dotnet run

＊プロジェクトのディレクトリに実行しない場合や途中から実行する場合は以下のコマンドを使用する。

cd C:\MySolidWorksProject