

C#WindowsForm 作成

サンプル資料

## 目次

1	本書の目的 .....	1
2	開発環境構 .....	2
2.1	Visual Studio 2019 Community インストール手順 .....	2
3	WindowsForm 作成 .....	4
3.1	ソリューションとプロジェクトの作成 .....	4
3.2	WindowsForm へのコントロール(部品)配置と属性変更 .....	6
3.3	イベントハンドラの実装 .....	10
3.4	サンプルプログラムの実行 .....	12

# 1 本書の目的

本書は C# を利用し、独習で簡単な WindowsForm アプリの作成とコントロールの属性変更などを行える様になる事を目的としています。

対象者 IT スキルは下記を行える事を目安にしています。

- PC の基本操作
- ソフトウェアのインストール、アンインストール
- インターネットサービスのアカウント新規登録等

## 2 開発環境構

### 2.1 Visual Studio 2019 Community インストール手順

1. Visual Studio Community 2019 - Free IDE and Developer Tools (microsoft.com)を下記 URL からダウンロードする。

<https://visualstudio.microsoft.com/ja/vs/community/>



図 1 ダウンロードリンク

2. ダウンロードしたインストーラーを起動する  
任意のフォルダにダウンロードした Visual Studio 2019 Community のインストーラーを起動する。



図 2 ダウンロードしたインストーラーファイル

(※インストーラーの数字の部分は同一の必要はありません)

3. インストールウィザードからインストールを行う  
「このアプリがデバイスに変更を加えることを許可しますか」  
上記のメッセージウィンドウが表示されたら、「はい」をクリックする。

「作業を開始する前に、インストールを構成するためにいくつかの点を設定する必要があります。」

上記のメッセージウィンドウが表示されたら、「続行」をクリックする。



図 3 インストール内容の選択

上図、図 3 と同じく「NET デスクトップ開発」、「C++によるデスクトップ開発」にチェックを入れる。「インストール(I)」をクリックする。

インストール処理が始まり、進捗が表示されるので、インストール終了まで待つ。

インストールウィザードが閉じ、下図の画面が表示され、インストール終了

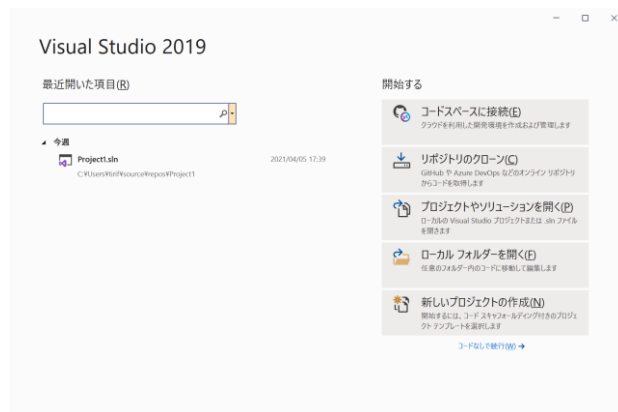


図 4 インストール終了後に自動的に起動した Visual Studio の画面

## 3 WindowsForm 作成

### 3.1 ソリューションとプロジェクトの作成

インストールした Visual Studio を起動し、  
WindowsFrom アプリを開発する為の ソリューション、プロジェクトを作成します。

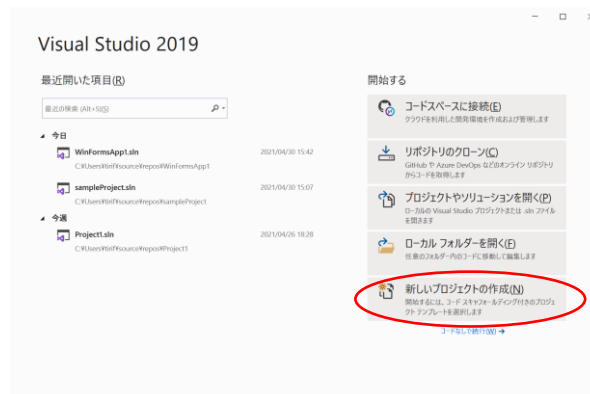


図 5 プロジェクト作成メニュー

対象言語を C#、プラットフォームを Windows に設定し下に表示される  
Windows フォーム アプリを選択します。

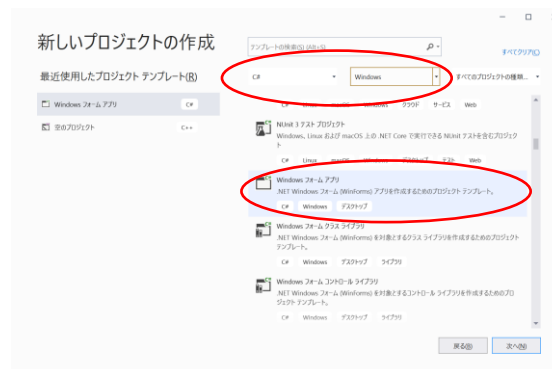


図 6 対象言語、プラットフォームの選択と

ソリューション、プロジェクト名、保存先の設定をします。



図 7 ソリューション、プロジェクト名、保存先の設定

ターゲットフレームワークを設定します。(今回はデフォルトの.NET 5.0 に設定)



図 8 ターゲットフレームワークの設定

## 3.2 WindowsForm へのコントロール(部品)配置と属性変更

デザイナー編集画面が表示されるので、左のツールボックスから配置する部品を選択します。

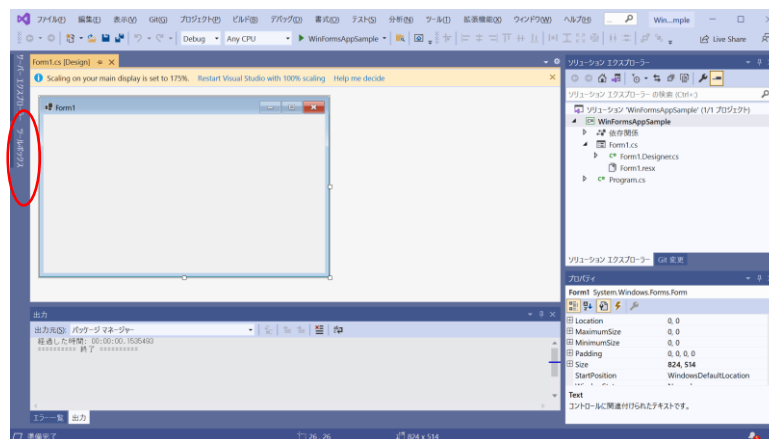
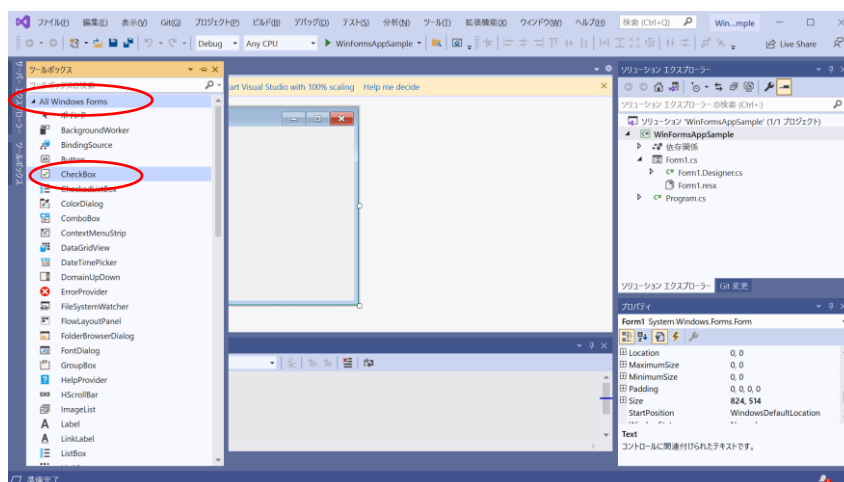


図 9 デザイナー編集画面とツールボックス

ツールボックスの「All Windows Forms」または「すべてのコントロール」からチェックボックスとパネルを選択し設置します。(ドラッグ&ドロップで設置)





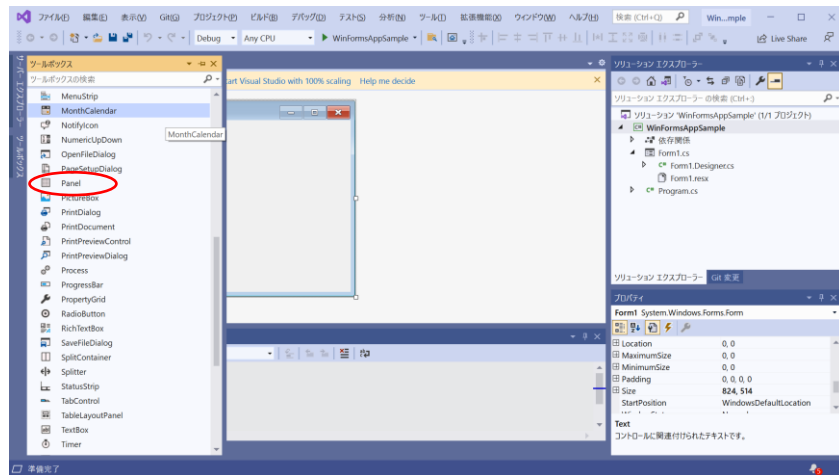


図 10 すべてのコントロールとチェックボックス、パネル

下図のようにチェックボックスとパネルを設置します。

位置の変更や部品のサイズはマウスのドラッグで変更します。

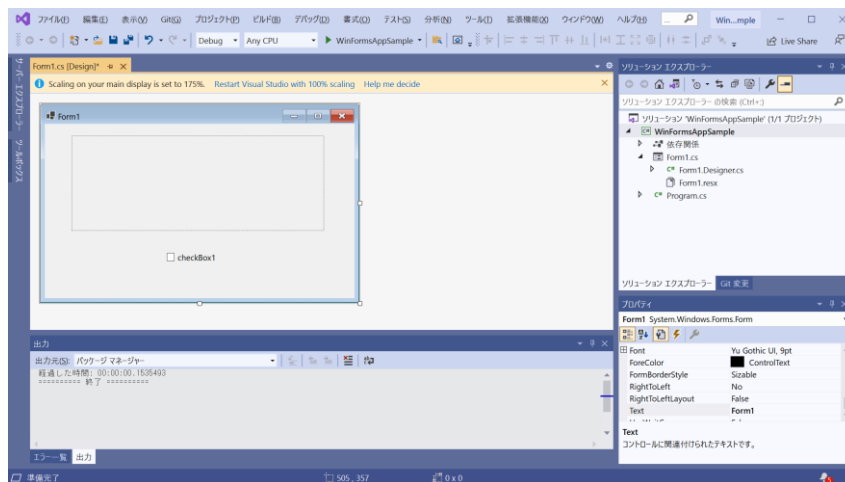


図 11 チェックボックス、パネルの配置例

パネルの部品をクリックし右下のプロパティウィンドウから

Border Style 属性を変更し見た目の変更を行います。(今回は Fixed3D に設定)

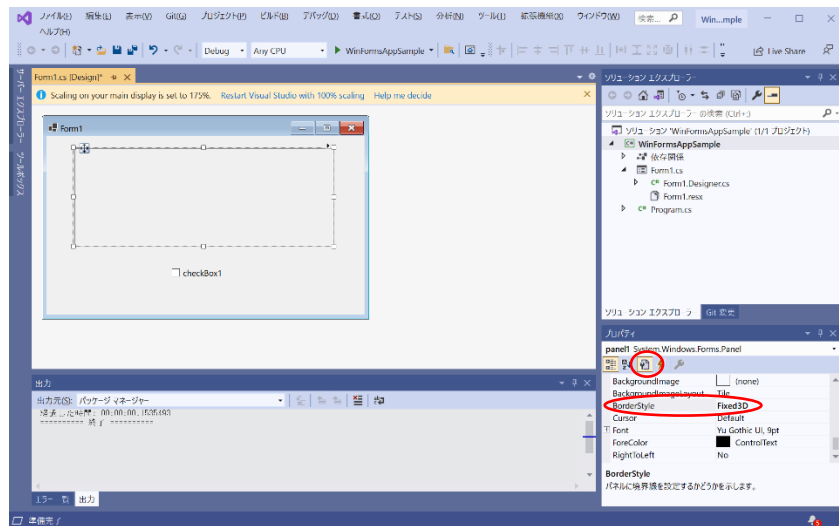


図 12 パネルの部品属性変更

チェックボックスも同様にプロパティウィンドウから Appearance を変更し見た目をボタンに変更します。

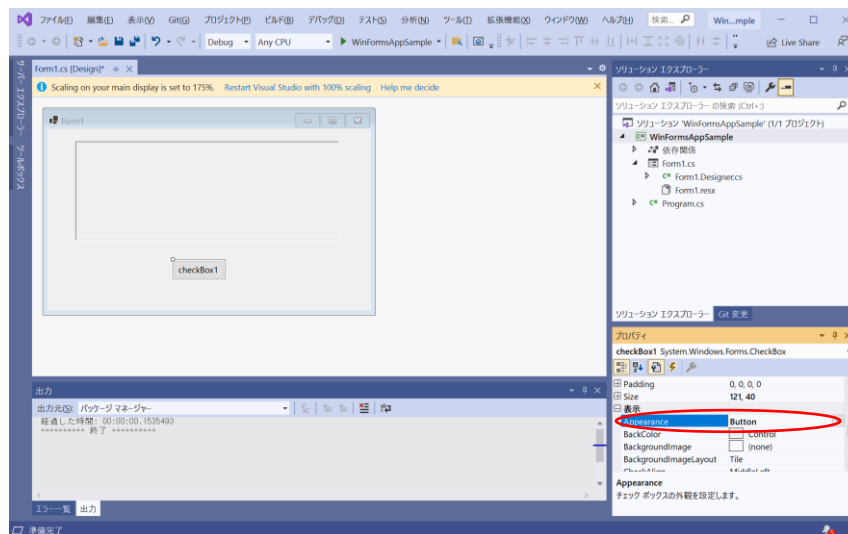


図 13 チェックボックスの属性変更

### 3.3 イベントハンドラの実装

デザイナー上のボタンに見た目を変更したチェックボックスをダブルクリックしソースウィンドウとイベントハンドラを開きます。

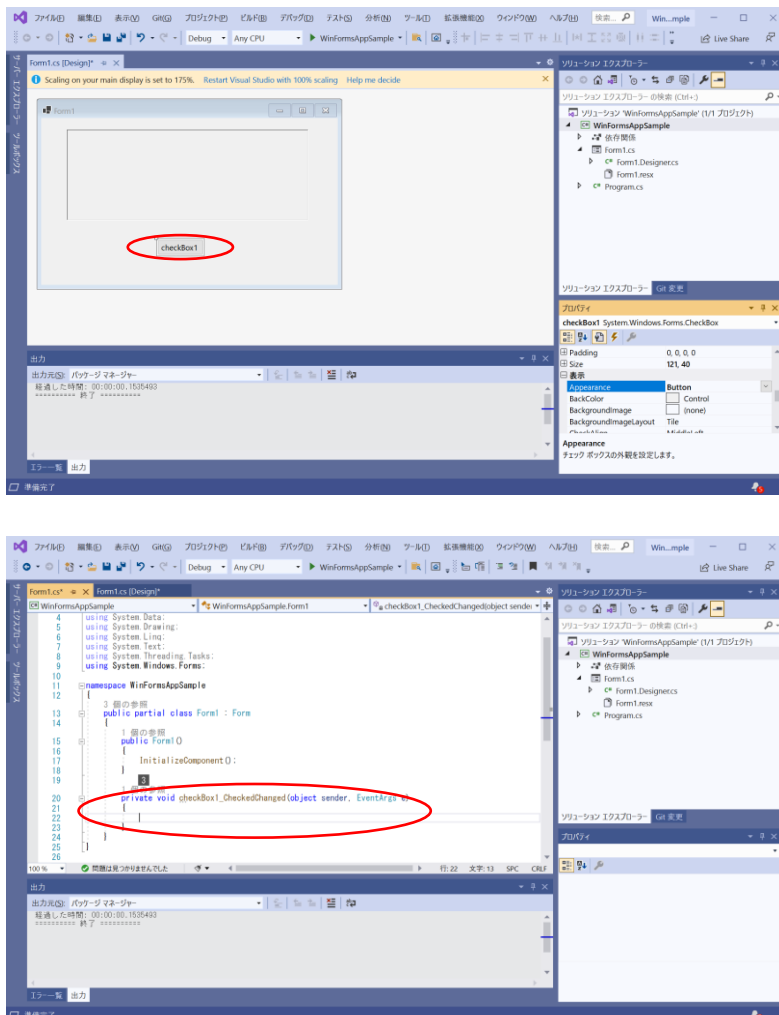


図 14 イベントハンドラの作成

イベントハンドラ内に下記のソースを記入します。

```
if (checkBox1.Checked==true)
{
    panel1.BackColor = Color.LimeGreen;
}
else
{
    panel1.BackColor = Color.Red;
}
```

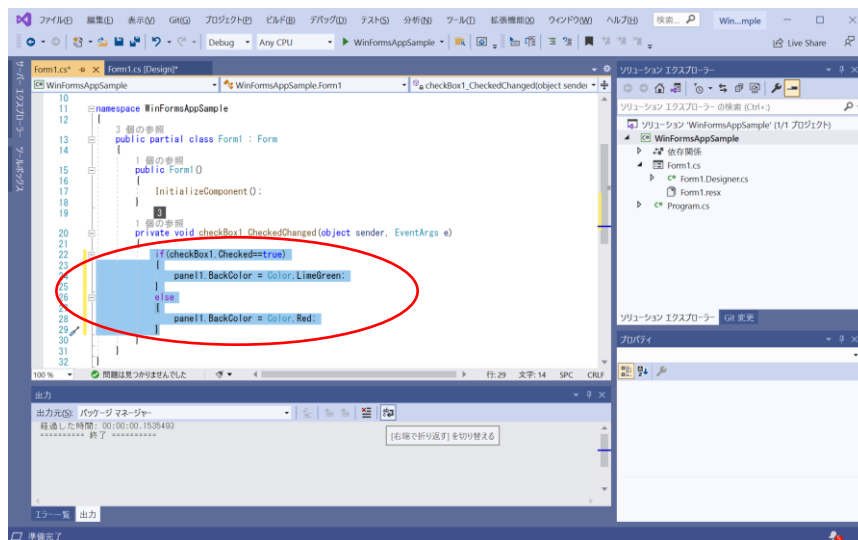
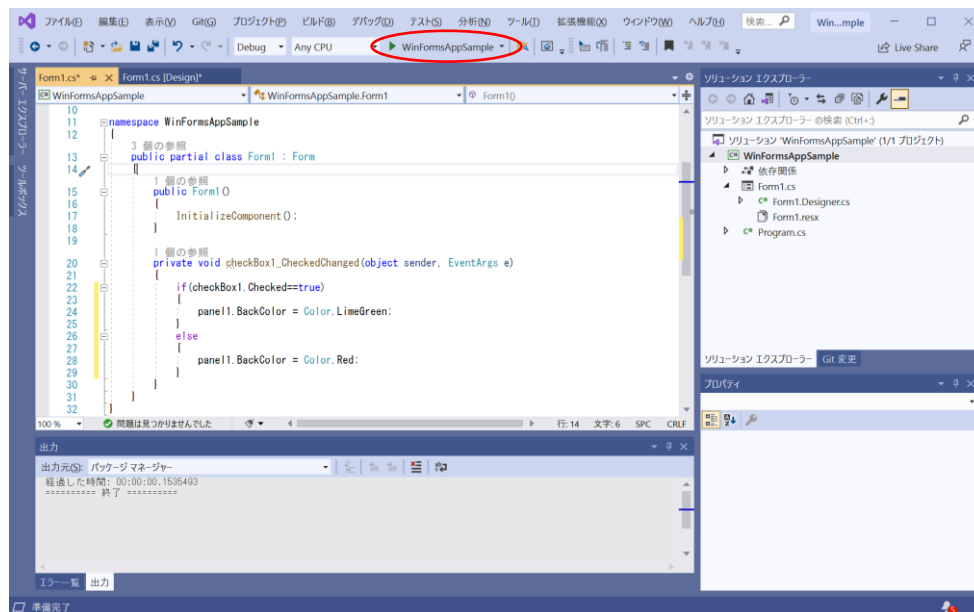


図 15 イベントハンドラの実装コード

### 3.4 サンプルプログラムの実行

デバッグ開始ボタンをクリックしサンプルプログラムを実行します。



プログラムが起動し初期状態になります。

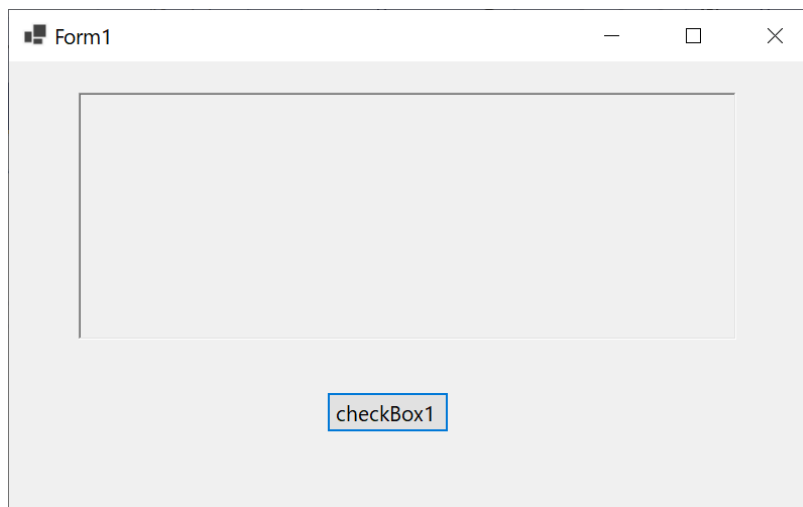


図 16 起動したサンプルプログラムの初期画面

checkBox1 をクリックし最初は緑色、緑色の状態でもう一度クリックし赤色になれば正常に動作しています。

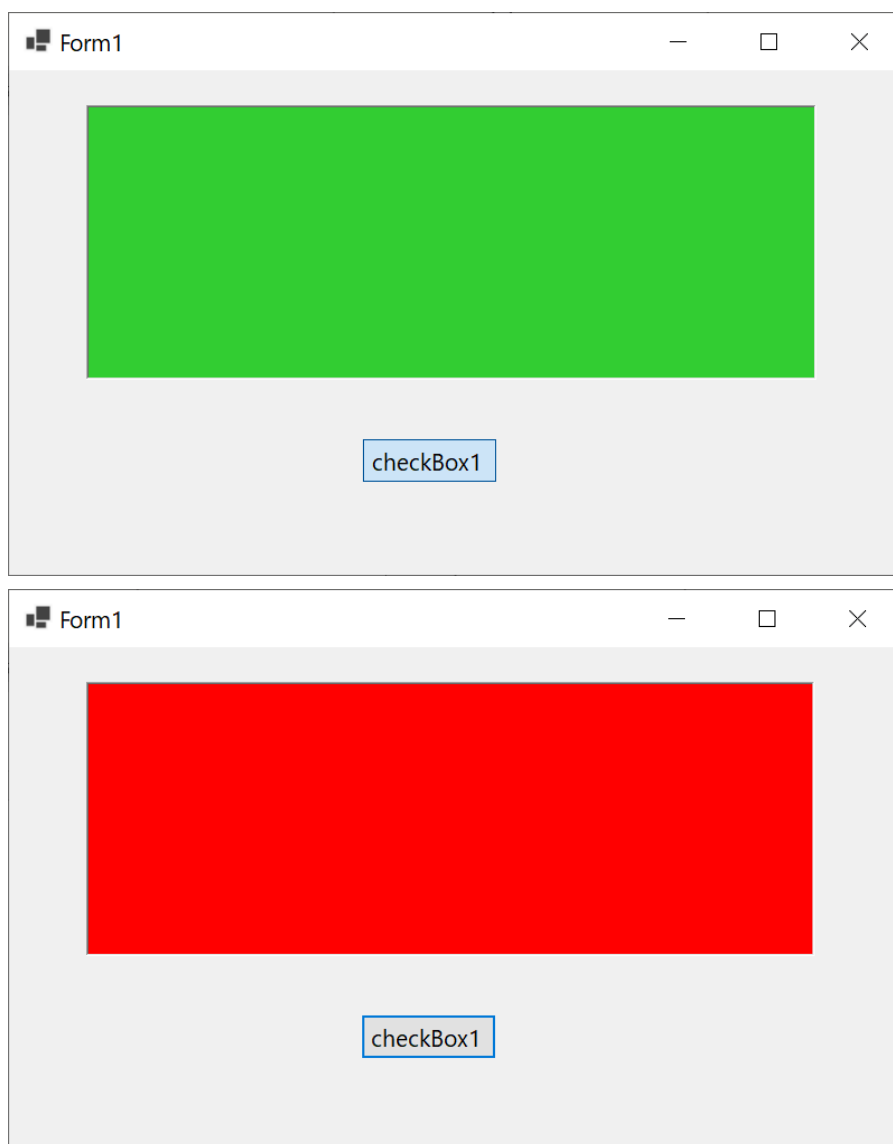


図 17 サンプルプログラムの動作イメージ

簡単になりますがこのように WindowsForm アプリケーションはデザイナー上で部品を配置し部品ごとのイベントハンドラを実装し作成します。

下記、同様の内容になりますが参考サイトになります。

[C# で Windows フォーム アプリを作成する - Visual Studio | Microsoft Docs](#)