

チェックボックス

真偽の GUI 表現

アプリケーションの設定を行うためにユーザーに選択を促す時などは
選択済みか非選択状態かなどがよりわかりやすいコントロールを使う必要があります
Button は何らかの機能を発動させるのにはきわめて優秀なコントロールですが
真偽の設定をユーザーに問いかけるには、概観上あまり適切ではありません

真偽の状態、すなわちアプリケーションが設定している BOOL 型のデータを
より直感的にユーザーに見せるのがチェックボックスです
これも、ボタンのひとつで **System.Windows.Forms.CheckBox** で表されます

```
System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      System.Windows.Forms.Control
        System.Windows.Forms.ButtonBase
          System.Windows.Forms.CheckBox
```

```
public class CheckBox : ButtonBase
```

このクラスのコンストラクタは、デフォルトコンストラクタしかありません

```
using System.Windows.Forms;
using System.Drawing;

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }

    public WinMain() {
        CheckBox bt = new CheckBox();
        bt.Size = new Size(200, 30);
        bt.Text = "Kitty on your lap";
        Controls.Add(bt);
    }
}
```



このプログラムは、単純なチェックボックスコントロールを表示します
.NET のチェックボックスは、デフォルトで基本的な動作を有効にしているため
ボタンをクリックすれば、図のようにチェックマークがつきます
デフォルトの動作は **CheckBox.AutoCheck** プロパティで解除できます

```
public bool AutoCheck {get; set;}
```

このプロパティを false に設定すれば、クリックしても何も起こらなくなります
この場合は、イベントを使って自分で制御する必要があるでしょう

チェックボックスのチェック状態は **CheckBox.Checked** プロパティで表されます

```
public bool Checked {get; set;}
```

このプロパティを使って、チェックボックスのチェック状態を制御したり
あるいは、チェックボックスのチェック状態を知ることができます

```
using System.Windows.Forms;
using System.Drawing;

class WinMain : Form {
    public static void Main(string[] args) {
        Application.Run(new WinMain());
    }

    public WinMain() {
        CheckBox bt = new CheckBox();
        bt.Size = new Size(200, 30);
        bt.Text = "Kitty on your lap";
        Controls.Add(bt);
    }

    override protected void OnMouseUp(MouseEventArgs e) {
        CheckBox bt = (CheckBox)Controls[0];
        MessageBox.Show("チェック状態は" + bt.Checked + "です");
    }
}
```

このプログラムは、ウィンドウのクライアント領域をクリックすると
チェックボックスの状態を知らせるメッセージボックスを表示します

チェックボックスが、選択をしているかしていないかを示すだけならば2つの状態で事足りますが
「真である」「非選択」「偽である」の3状態を示さなければならないこともあるでしょう
つまり**曖昧な状態**というもうひとつの状態が必要なケースです

チェックボックスには **CheckBox.ThreeState** プロパティを設定することで従来の ON/OFF に加え、曖昧な状態を追加することができます

```
public bool ThreeState {get; set;}
```

このプロパティを true にすれば、3 状態チェックボックスになります
曖昧な状態を含むチェックの制御は bool 型で行うことはできません
そこで、3 状態チェックボックスは **CheckBox.CheckState** プロパティで制御します

```
public CheckState CheckState {get; set;}
```

このプロパティは、**System.Windows.Forms.CheckState** 列挙型です
CheckState 列挙型は、3 状態チェックボックスの状態を示すメンバを持ちます

```
[Serializable]  
public enum CheckState
```

この列挙型は、次のようなメンバを定義しています

メンバ	解説
Checked	コントロールがチェックされています
Indeterminate	コントロールが不確定状態になっています 不確定状態のコントロールは、通常、網かけ表示されます
Unchecked	コントロールがチェックされていません

```
using System.Windows.Forms;  
using System.Drawing;  
  
class WinMain : Form {  
    public static void Main(string[] args) {  
        Application.Run(new WinMain());  
    }  
  
    public WinMain() {  
        CheckBox bt = new CheckBox();  
        bt.Size = new Size(200 , 30);  
        bt.Text = "Kitty on your lap";  
        bt.ThreeState = true;  
        Controls.Add(bt);  
    }  
}
```



このプログラムは、曖昧な状態を含む3状態チェックボックスを表示します
図を見てわかるように、曖昧な状態はチェックの表示がほかの2状態とは異なります