

グループボックス

コントロールのグループ

ラジオボタンのように、グループを認識するようなコントロールや特定の論理的な意味を持つコントロールグループを管理する必要がある場合それらを全てメインウィンドウに追加して管理するという方法はオブジェクト指向という崇高な理念の理想からは大きく外れたものといえます

このような処理を求められる場合、グループを制御する専用のコントロールを作ることが望まれます専用のコントロールは、もちろん自分で Control クラスを拡張してもよいのですが .NET では **System.Windows.Forms.GroupBox** を使ってグループ化できます

```
System.Object
  System.MarshalByRefObject
    System.ComponentModel.Component
      System.Windows.Forms.Control
        System.Windows.Forms.GroupBox
```

```
public class GroupBox : Control
```

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定義されていません基本的に、このクラスは Control をグループ管理専用拡張したというだけです何らかの処理を行うための機能などは追加されていません

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

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

    public WinMain() {
        string[] kitty = {"Rena", "Yuki", "Mimi"};
        string[] taruto = {"Taruto", "Charlotte", "Chitose"};

        GroupBox[] gb = {new GroupBox(), new GroupBox()};
        gb[0].Bounds = new Rectangle(10, 10, 200, 130);
        gb[1].Bounds = new Rectangle(220, 10, 200, 130);
        gb[0].Text = "Kitty on your lap";
        gb[1].Text = "Magical nyan nyan TARUTO";

        PutButton(gb[0], kitty);
        PutButton(gb[1], taruto);

        foreach (GroupBox ctrl in gb) Controls.Add(ctrl);
    }
    public void PutButton(Control ctrl, string[] text) {
        RadioButton[] bt = new RadioButton[text.Length];
        for (int i = 0; i < bt.Length; i++) {
            bt[i] = new RadioButton();
            bt[i].Bounds = new Rectangle(20, 20 + (30 * i), 150, 30);
            bt[i].Text = text[i];
            ctrl.Controls.Add(bt[i]);
        }
    }
}
```



図の四角い枠組みがグループボックスコントロールですこのプログラムでは、論理的に意味の異なるラジオボタンの配列をそれぞれ、異なるグループとして管理するためにグループボックスを作成して追加していますラジオボタンは、メインウィンドウではなくグループボックスに追加されているのです