R流

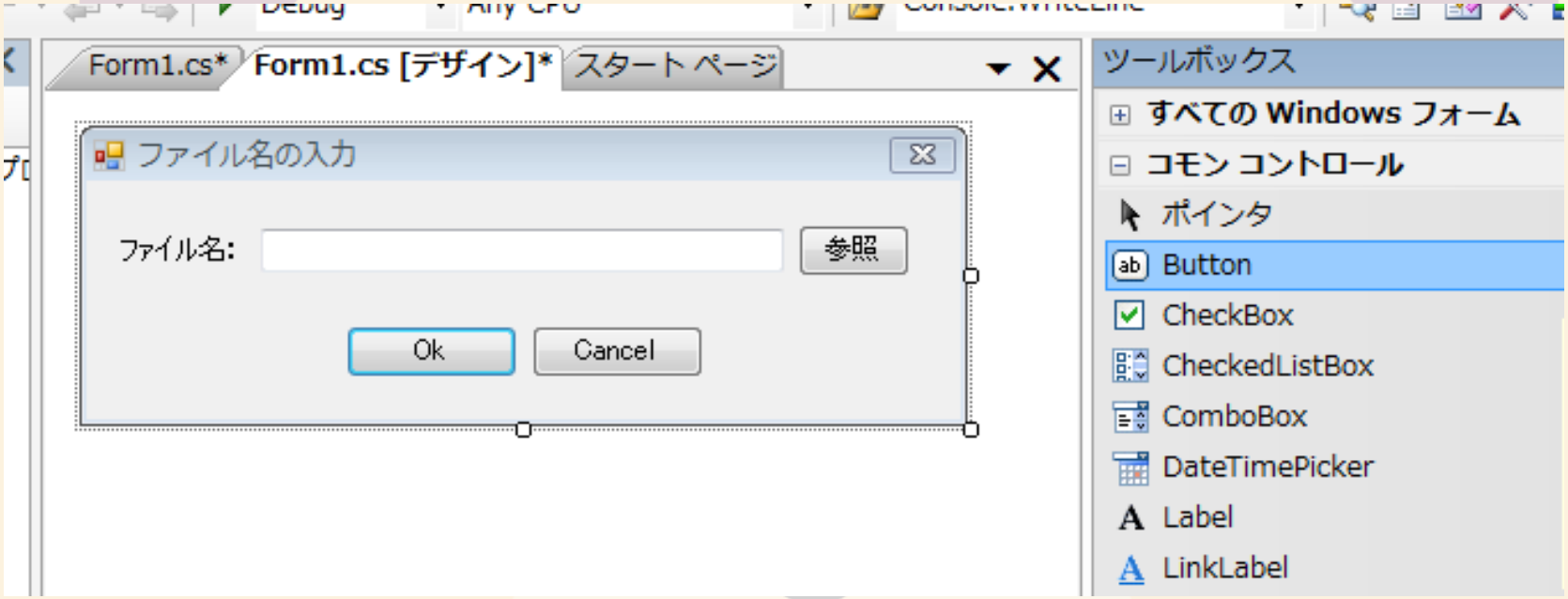
一瞬でわかる
.NETオブジェクト指向入門

2007年7月21日 R・田中一郎
http://blogs.wankuma.com/rti/

自己紹介

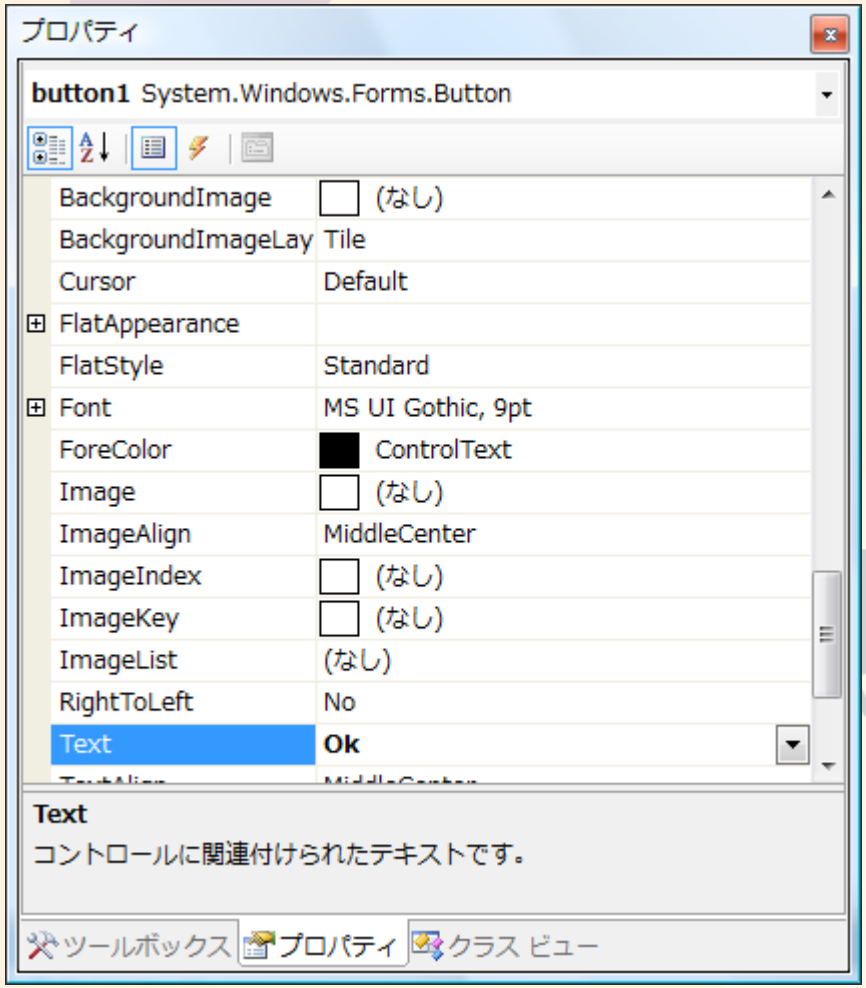
```
public class R・田中一郎 {
    public string 会員番号 { get { return "34";            }}
    public string 名前       { get { return "R・田中一郎";  }}
    public string 年齢       { get { return "18";            }}
    public string 職業       { get { return "IT関係";        }}
    public string 印象       { get { return "素敵だし爽やか"; }}
    public void   自己紹介(){
        Console.Writeline(
            "オブジェクト指向ファンの皆さん、こんにちは。" +
            "わんくま会員番号" + this.会員番号 + "番の" + this.名前 + "です。" +
            "年齢は、" + this.年齢 + "才。職業は、" + this.職業 "です。" +
            "今日は、オブジェクト指向ファンの皆さんを前にして、" +
            this.印象 + "な僕が、オブジェクト指向について語るということで、" +
            "些かガクブルな状態ではありますが、早速はじめたいと思います。"
        );
    }
}
```

てっとり早く理解しよう！



ダイアログボックスをデザイナで作っている画面です。
これを見ながら、てっとり早くオブジェクト指向の雰囲気だけをつかんでしまいましょう。

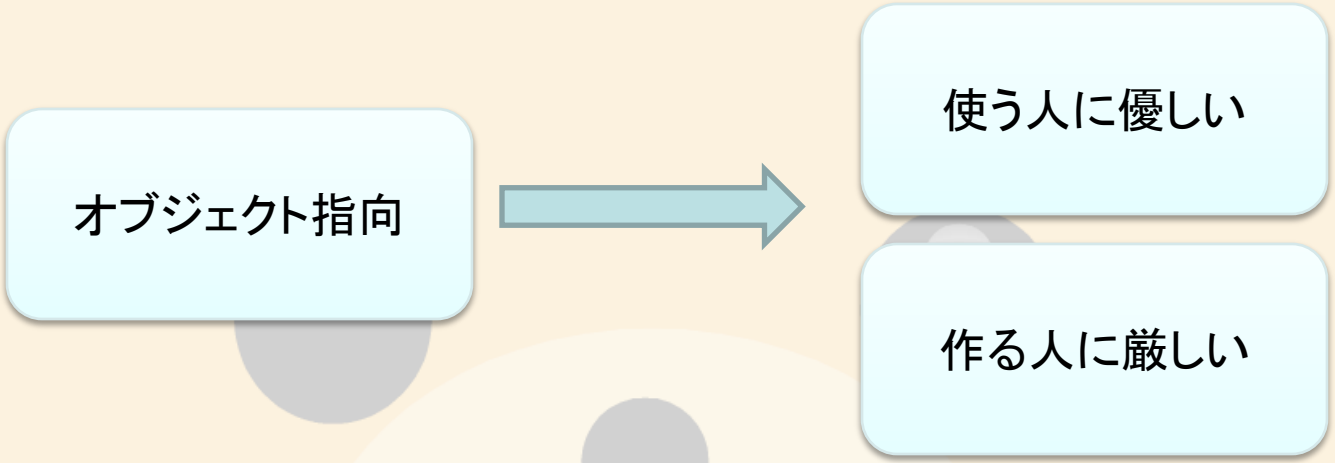
クラスはメンバで構成される



なんて表示する？
何色で表示する？
背景は何色にする？
イメージはどうする？

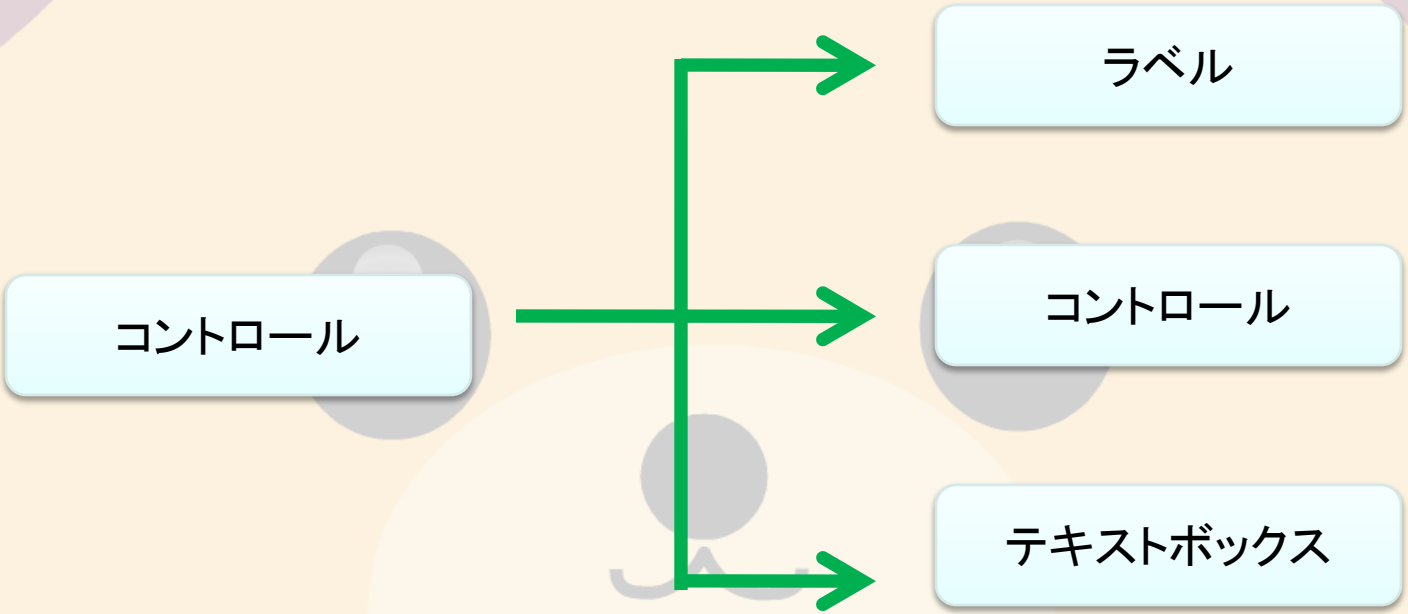
最前面に移動する。
フォーカスをあてる。
クリックイベントを発生させる。
などなどなどなど。

使うは易し・作るは難し



何故だー（Ｔ－Ｔ）

クラスは継承できる



注意！
本当は、もうちょっと複雑ですよ。

オーバーライドする

```
public class R・田中一郎　{
　～　略しますよー　～
  public void virtual　自己紹介() {　～　略しますよー　～ }
}

public class R・田中二郎　：　R・田中一郎 {
  public void override 自己紹介(){
        Console.Writeline(
            “構造化プログラミングファンの皆さん、こんにちは。" +
            "わんくま会員番号" + this.会員番号 + "番の" + this.名前 + "です。" +
            "年齢は、" + this.年齢 + "才。職業は、" + this.職業 "です。" +
            "今日は、オブジェクト指向ファンの皆さんを前にして、" +
            this.印象 + “な僕が、構造化プログラミングについて語ります。”
        );
    }
}
```

アクセス修飾子

- public 全体に公開
- internal 同じアセンブリのみ公開。
- protected 派生したクラスにのみ公開
- internal protected 同じアセンブリ内で派生したクラスにのみ公開
- private 自クラスのみ公開

結局、隠す、見せる、の判断だけ！

オブジェクトって何？

神様になって、天地創造をするプログラムをC＃で書いてみる。

```
class 海 {}
class 陸 {}
class 山 {}
class 川 {}
void 創造するよーん() {
    海 u = new 海();
    陸 r = new 陸();
    山 y = new 山();
    川 k = new 川();
}
```

大切なのは、中見なんですよ。

実際に設計してみよう

- ３並べって知ってますか？
- ○と×を、３×３のマス目に置くゲームです。
- ○と×が縦横斜めに3つ並んだら、その記号の主が勝ちで、ゲームオーバーです。
- 全てのマス目に○か×が置かれたら引き分けです。

System.Objectの継承は省略できます

```
R・田中一郎 r = new R・田中一郎();
Console.WriteLine(r.ToString());
```

Q.このコードは、エラーになりますか？

A.いいえ、エラーになりません。

ToString() メソッドなんて実装していないのに、
どうしてエラーにならないのでしょうか？

System.Objectとは？

public Object();

public virtual bool Equals(object obj)
public static bool Equals(object obj1, object obj2)

protected override void Finalize()
public virtual int GetHashCode()
public Type GetType()
protected object MemberwiseClone()
public static bool ReferenceEquals(objectobj1, object obj2)
public virtual string ToString()

お馴染みのメンバーですよ。

型に入れる

```
Button b  = new Button();
Control c = new Button();
object o  = new Button();

object r = new R・田中一郎();
r.自己紹介();
```

これは、実行できますか？

```
object r = new R・田中一郎();
r.自己紹介();
```

抽象クラス

- わんくま同盟クラスのメンバは、実際に何を返したり、何を実行すれば良いの？
- 必要なメンバだけはわかりますよ。
- じゃあ、メンバだけ実装して、後は勝手に継承してもらいましょう。

インターフェイス

- 抽象クラスを継承すると、ひとつしか継承できません。
- しかし、たくさんのインターフェイスを継承すれば、それだけたくさんの多様化ができます。
- インターフェイスを使ってみましょう。

ご清聴ありがとうございました。

皆さんも、月曜日から、

是非、

オブジェクト指向を実践してみて下さい。

http://blogs.wankuma.com/rti/