

R流・1時間でわかる 実践オブジェクト指向プログラミング

2007年11月17日

R・田中一郎

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

Microsoft MVP for Visual Developer C#



わんくま同盟 東京勉強会 #14

自己紹介

```
public class R・田中一郎 : Iわんくま同盟, Iお笑い同盟, Iわんくま同盟音楽部 {  
    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。"  
        );  
    }  
}
```



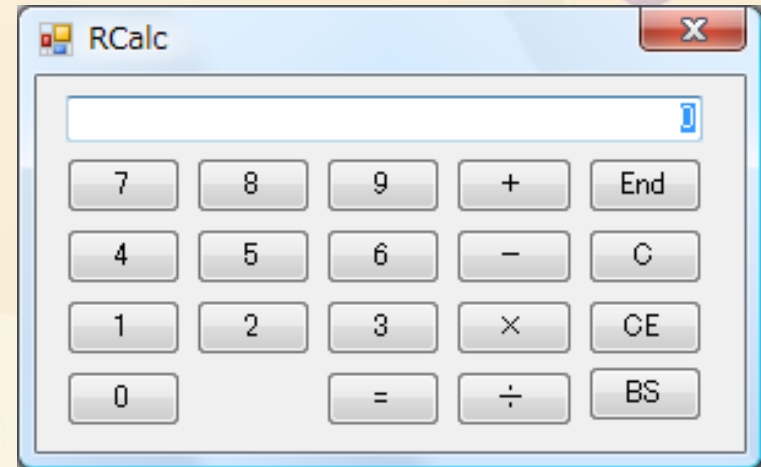
オブジェクト指向の三大要素

- カプセル化
メンバ(メソッド・プロパティ)の公開を抑制します。
暗に内部でのみ処理する部分を隠ぺいします。
- 継承
元となるクラスやインターフェイスを継承できます。
継承元のメンバの利用や置き換えが可能です。
- ポリモーフィズム(多態性)
同じメンバを呼び出しても、オブジェクトによって異なる
振る舞いをします。

電卓を作ろう！

電卓を構成するオブジェクトとは？

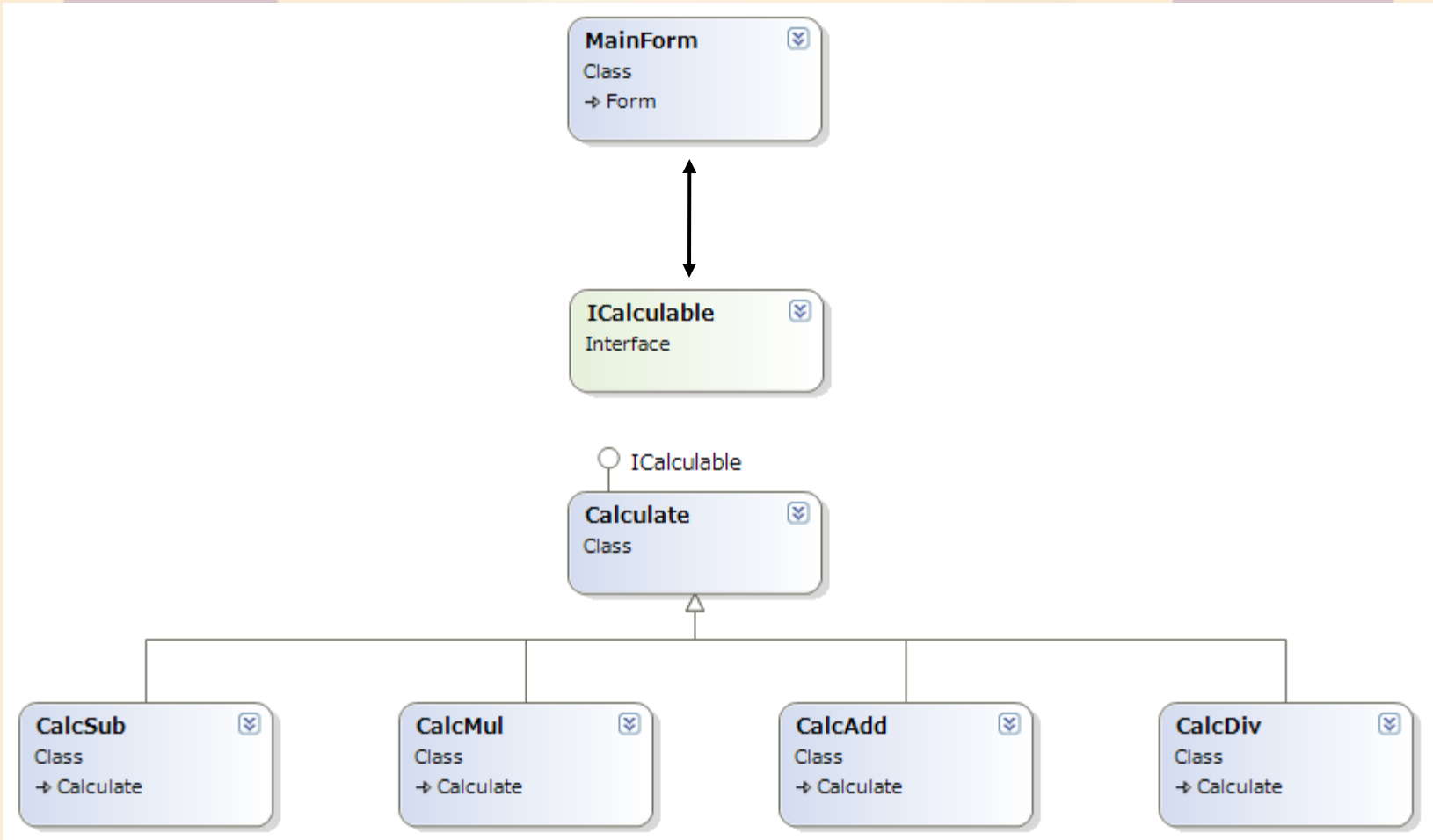
- 一行ディスプレイ
 - 0～9の数字ボタン
 - + - × ÷ の演算ボタン
 - その他の機能ボタン
-
- 数字列を記録するレジスタ
 - 前回入力された数値列を覚えておくレジスタ
 - 演算処理(加算・減算・乗算・除算)



電卓のオブジェクト

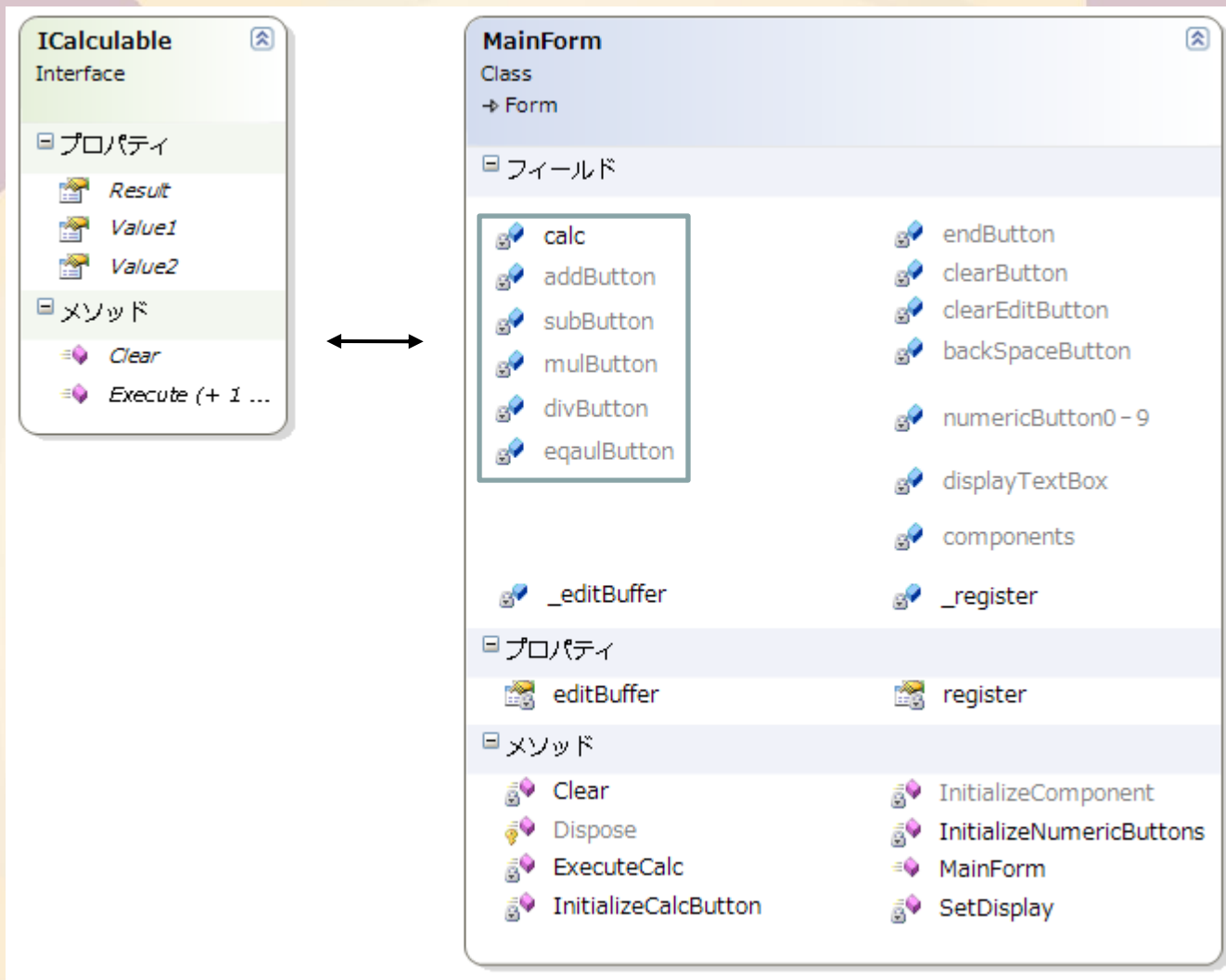
- 一行ディスプレイ→ TextBox
- 0～9の数字ボタン→ Button
- + - × ÷ の演算ボタン→ Button
- その他の機能ボタン→ Button
- 数字列を記録するレジスタ→ string
- 前回入力された数値列を覚えておくレジスタ→ string
- 演算処理(加算・減算・乗算・除算)→ ICalculable

オブジェクトの関係



演算オブジェクトには、必ず ICalculable インターフェイスを経由してアクセス

フォームと演算オブジェクトの関係



R流インターフェイスとは？

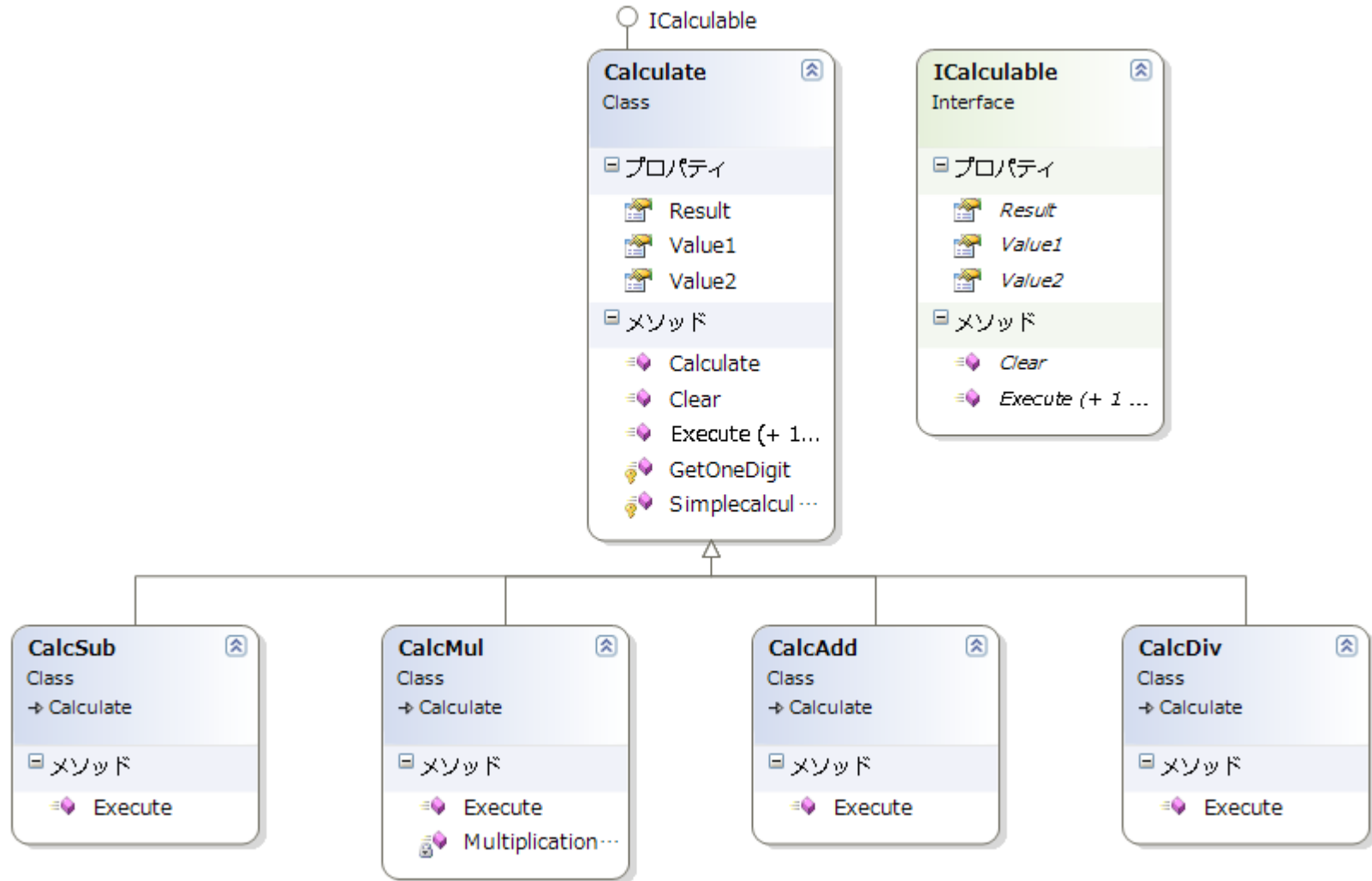
以下は、ある家電品のインターフェイスの一部です。



再生ボタン・早送り・巻き戻しボタンなどがあります。
これらのスイッチを押した際の振る舞は、容易に想像できます。

これらのインターフェイスは、様々な家電品で組み合わせて再利用できます。
また、必要なものだけを利用することもできます。

演算オブジェクトの関係



R流クラス・インターフェイス解

クラスは次の3つの役割がある。

型……データの型

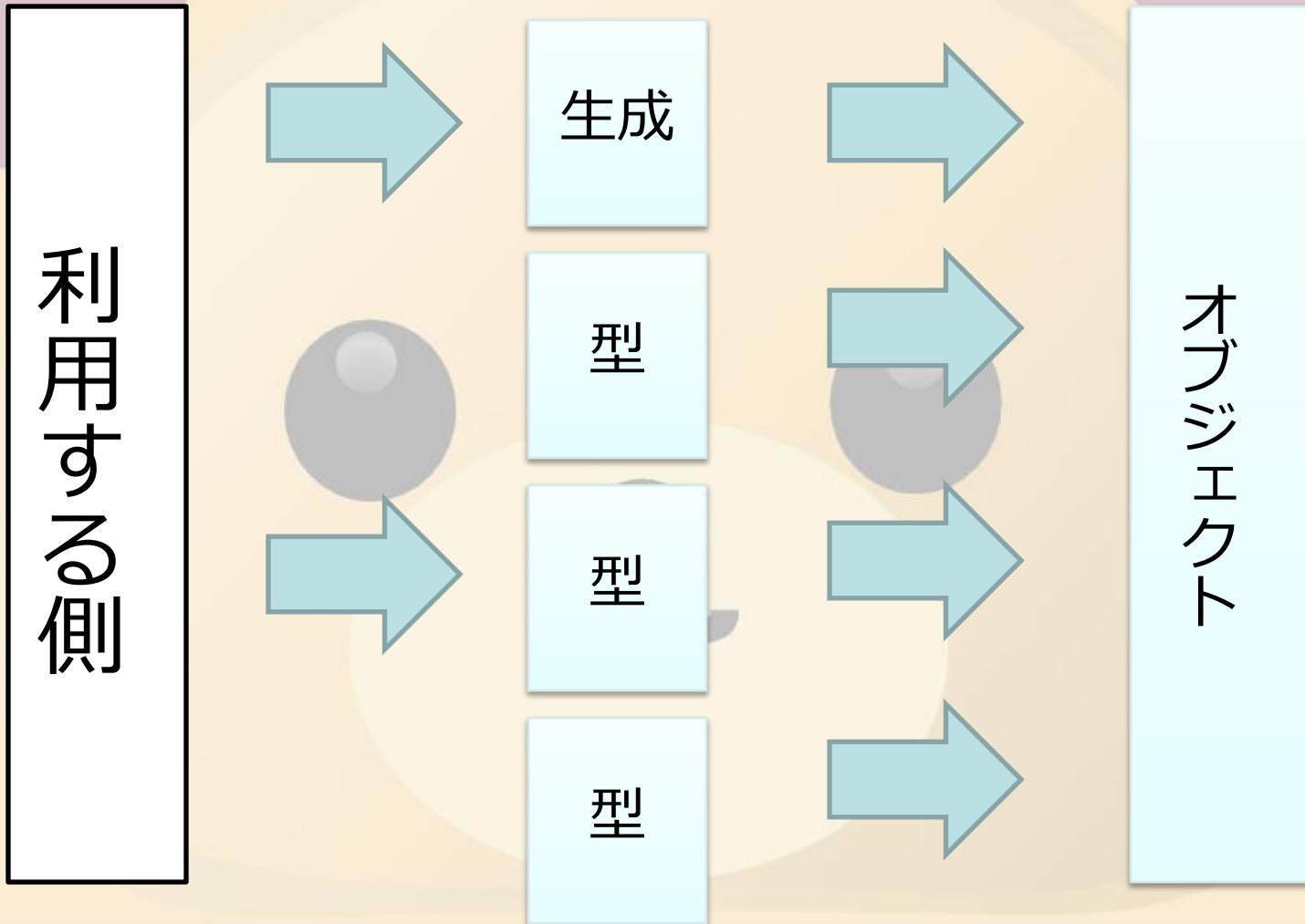
生成……オブジェクトを生成

実装……実際の振る舞い

インターフェイスは型としての役割しかない。

(故にクラスが複数継承できる)

R流クラス・インターフェイスとオブジェクト



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、

オブジェクト指向プログラミングを
実践してみてください。

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

