

# C#4.0 ファーストコンタクト

中博俊

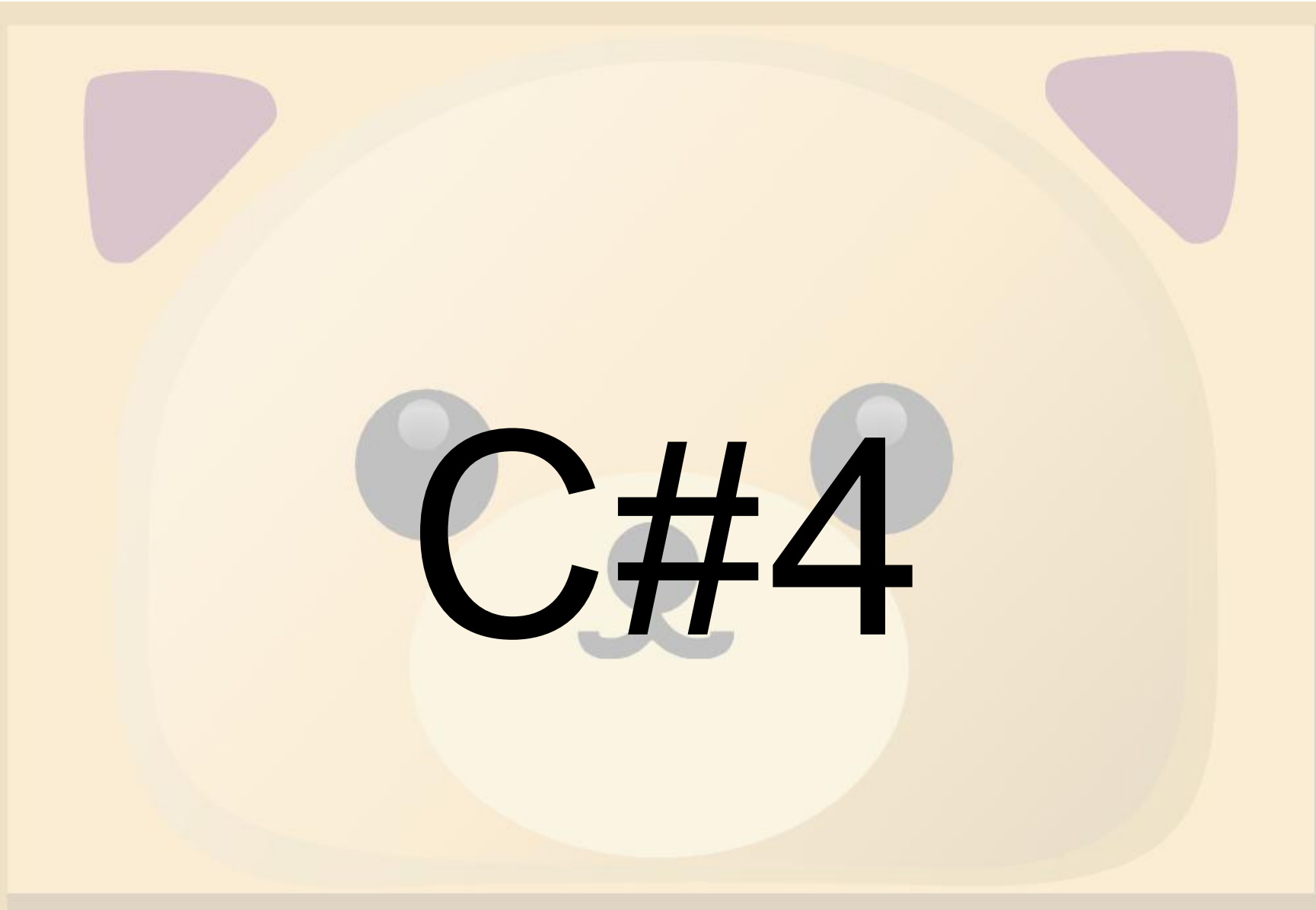
## Visual Studio の変更

- 次期Visual Studio
- Visual Studio 2010
  - 変更、修正箇所が半端無く多い
- VSがWPFベースに
- テスト支援がすごい
  - 再現情報の収集
  - ビデオの収集



## 言語系

- C# 4.0
- VB 10.0
- Python, Ruby DLR搭載?
- F#搭載?
- M搭載?
- C++0x一部搭載(ちよつとしよぼめ)



C#4

## C#4の新機能

- Dynamic Types
  - IronPython Script Support
- Office Programmability
  - Named Parameters
  - No-PIA
- Parallel Programming



dynamic

```
public class target {  
    public void Talk() {  
        Console.WriteLine("wan"); } }  
static void Main(string[] args) {  
    var x = CreateInstance();  
    x.Talk(); }  
static object CreateInstance() {  
    return new target(); } }
```



dynamic

```
var x = CreateInstance();  
//x.Talk();  
var type = x.GetType();  
var mi = type.InvokeMember("Talk",  
    BindingFlags.Instance  
    | BindingFlags.InvokeMethod  
    | BindingFlags.Public,  
    null, x, null);
```



dynamic

**dynamic x**

**=CreateInstance();**

**x.Talk();**





dynamic

存在しないメソッドを呼んだ場合には？

**RuntimeBinderException**

```
at _stub_$1##1 (Closure , CallSite , Object )  
  at  
  System.Scripting.Actions.MatchCaller.CallVoid1[T0] (Action  
  `2 target, CallSite site, Object[] args)  
  at  
  System.Scripting.Actions.CallSite`1.UpdateAndExecute (Object[] args)  
  at  
  System.Scripting.Actions.UpdateDelegates.UpdateVoid1[T,T0]  
  ] (CallSite site, T0 arg0)
```



dynamic

Intにdynamicは？

```
static void M(object o) {  
    dynamic i = (int)o; }  
for ( var loop = 0 ; loop < i ;  
loop++ )
```

Error 1 Operator '<' cannot be applied to operands of type 'int' and '::dynamic'



## Dynamic は Duck Typingに<sup>あ</sup>らず

```
public class Dog { public void Talk()
    { Console.WriteLine("BowWow"); } }
public class Cat { public void Talk()
    { Console.WriteLine("Mew"); } }
public interface IAnimal { void Talk();}
static void Main(string[] args)
    {
        var d = new Dog();
        var c = new Cat();
        d.Talk();
        c.Talk();
        IAnimal a = d as IAnimal;
        a.Talk();
    }
```



## Iron Python Support

```
using IronPython.Hosting;  
using  
    Microsoft.Scripting.Hosting;  
ScriptRuntime py =  
    Python.CreateRuntime();  
dynamic random =  
    py.UseFile("random.py");  
random.shuffle(items);
```



Office Develop & Named Parameters

```
ord.Selection.PasteSpecial(  
    Link: true,  
    DisplayAsIcon: true);
```



## Task

```
using System.Threading;
using System.Threading.Tasks;
static void M(object o)
{
    int i = (int)o;
    for ( var loop = 0 ; loop < i ;
loop++ )
        Console.WriteLine(
            string.Format("{0}:{1}", o,
loop ) );
}
```



## Task

```
class Program {  
    static void Main(string[] args) {  
        var tm = TaskManager.Default;  
        if (Environment.ProcessorCount < 4) {  
            tm = new TaskManager(new  
TaskManagerPolicy(1, 4, 4)); }  
        var lt = new List<Task>();  
        for (var i = 0; i < 10; i++) {  
            Task t = Task.StartNew(M, i, tm);  
            lt.Add(t); }  
        Task.WaitAll(lt.ToArray());  
    }  
}
```



Parallel.For

Parallel.For (1, 30,

$i \Rightarrow \{ M(i); \}$ );

でもVirtualMachine上では  
はパラレルで動かない





Parallel.For

```
Parallel.For<int>(1, 30, 1,  
    () => 1,  
    (i, s) => M(i),  
    (i) => { },  
    new TaskManager(  
        new TaskManagerPolicy(1, 4,  
            4)),  
    TaskCreationOptions.SelfReplica  
ting);
```



## Parallel LINQ (PLINQ)

```
(from x in Enumerable
    .Range(1, 30) .AsParallel(4)
 where x % 2 == 0
 select x)
    .ForAll(i =>
        Console.WriteLine(i));
```

## キーとなるクラスとデリゲート

- `public delegate void Action2(object obj)`
- `public delegate void Action2<T1,T2>(T1 obj1, T2 obj2)`
- `public delegate void Action2<T1,T2,T3>(T1 obj1, T2 obj2, T3 obj3)`
- `public delegate T Func2<T>()`
- `public delegate TResult Func2<T,TResult>(T val)`
- `Task`
- `TaskManager`
- `ParallelEnumerable`

C#4 まとめ

# Parallel

プログラミングが

アツイ！



わんくま同盟 大阪勉強会 #26