

1 ● polymorphism (多態性) について

2
3 interface (インターフェイス) を使う例 :

4 鳴く :

```
5     interface MakeEiSound{  
6         // make a sound ... 音を立てる  
7         void makeEiSound(String sound);  
8     }
```

9
10 好き (食べ物) :

```
11     interface LikeThis{  
12         void likeThis(String like);  
13     }
```

14
15 この、interface を使えば、いろいろな、例えば、ペットに、使えます。

16 そして、いつも、同じ名前のメソッドを使うようになれば、

17 すぐに、そのメソッドの、アクションを連想できるので、

18 コードの可読性が、向上します。

19 例えば、このように使えます。(イメージコード)

20
21 class Dog "implements" MakeEiSound, LikeThis

```
22     public void makeEiSound(String sound);
```

```
23     / 実装しなければならない! /コードを書く
```

```
24     / "実装する" ("implements"キーワード)
```

```
25     / publicキーワードが、必要!
```

```
26     / 理由は、interface 学習で、解決する予定。
```

```
27     public void likeThis(String like);
```

```
28     / "implements"しなければならない!
```

```
29     void hobby(String "hobby");
```

```
30     私は、"hobby"が、趣味です。
```

```
31     // このメソッドは、ここで初めて、作られる。  
32  
33
```

```
34
35 class Cat "implements" MakeEiSound, LikeThis
36     public void makeEiSound(String sound);
37         / 実装しなければならない！
38     public void likeThis(String like);
39         / 実装しなければならない！
40     void beGoodAt(String "good");
41         私は、"good"が、得意です。
42             // このメソッドは、ここで初めて、作られる。
43
44 class Bird "implements" MakeEiSound, LikeThis
45     public void makeEiSound(String sound);
46         / 実装しなければならない！
47     public void likeThis(String like);
48         / 実装しなければならない！
49     void eyeCan(String "can");
50         I can "can".
51             // このメソッドは、ここで初めて、作られる。
52
53 * "~"は、色付けのために、ダブルクォーテーションを、使っています。
54     / Stringオブジェクトとは、限りません。
55 EmEditorの色分け機能。
56 CSSを使ってやると、死ぬほど時間がかかります！
57     / 正規表現 と、最新のCSSのやり方 を使えば、
58     / 解決できるでしょうが。
59
```