オーバーロード

同名のメソッド

メソッドやコンストラクタといったものはシグネチャで個別に識別されます
シグネチャとはメソッドの識別子、仮パラメータ、修飾子によって構成され
シグネチャが異なるメソッドは同名でも異なるメソッドであると判断されます

これを利用することで、異なる引数を受け取る同名のメソッドを作れます
これはオブジェクト指向の多様性の一面でオーバーロードと呼びます
例えば、次のメソッドはそれぞれまったく別のメソッドとコンパイラは判断するでしょう

void Max(int x , int y) {}
void Max(ref int x , ref int y)
void Max(float x , float y) {}

これらのメソッドは名前は同じですが、シグネチャが異なっています
この恩恵は、高い保守性を必要とする膨大なシステムになるほど受けるでしょう
C言語では関数を拡張して引数をふやした場合は名前を変更する必要がありましたが
オブジェクト指向ではシグネチャさえ異なっていれば名前は衝突してもかまわないのです

```
class Test {
 static void Main() {
  int x = 0;
  Kitty("Magical nyan nyan");
  Kitty(ref x);
  Kitty(x);
  Kitty();
 }
 static void Kitty(string str) {
  System.Console.WriteLine(str);
 }
 static void Kitty(ref int x) {
  System.Console.WriteLine("Kitty on your lap");
 }
 static void Kitty(int x) {
  System.Console.WriteLine("Tokyo mew mew");
 }
 static void Kitty() {
  System.Console.WriteLine("Di Gi Charat");
 }
}
```

このプログラムで宣言されている Kitty() メソッドは全て異なるものです
どの Kitty() メソッドが呼び出されるかは、呼び出し側のパラメータ型で決定します
次のような結果がコンソールに出力されるでしょう

Magical nyan nyan
Kitty on your lap
Tokyo mew mew
Di Gi Charat

例えば、Main() メソッドで呼び出した Kitty(ref x) は
Kitty(ref int x) メソッドの型に一致するのでこのメソッドが呼び出されます
メソッドをオーバーロードすることによって、同名のメソッドに異なる意味を与えられます

もっとも、オーバーロードすることによってまったく異なるメソッドとして動作するといえ
一般的には同名のメソッドは論理的に単一の処理のために動作するべきです

オーバーロードの代表的な例は WriteLine() メソッドでしょう
このメソッドは、数値、浮動小数、文字列などのあらゆる型を受け取ることができました
これは、WriteLine() メソッドがオーバーロードされているためです
同様にコンストラクタをオーバーロードさせることも可能です

```
class Test {
 public Test() {
  System.Console.WriteLine("Kitty on your lap");
 }
 public Test(string str) {
  System.Console.WriteLine(str);
 }

 static void Main() {
  Test obj = new Test();
  obj = new Test("Tokyo mew mew");
 }
}
```

Test クラスにはオーバーロードされた二つのコンストラクタがあります
パラメータを何もしていしない Test() と文字列を指定する Test(string) です
メソッド同様に、呼び出された型で実行されるコンストラクタが決定します