メソッド

クラスの動作

前回はクラスの宣言とメンバ変数を解説しましたが
これだけでは、普通に Main() メソッドで変数を定義したほうが楽だと思うかもしれません

クラスにメンバ変数だけを定義して、それに単純にアクセスするだけでは
たしかに、通常の変数と変わらないので意味がありません

しかし、クラスはメンバ変数などを参考にして動作するメソッドの存在があります
メンバ変数は「名前」とか「性別」といったオブジェクトのデータを格納するものでしたが
メソッドはそれを参照してオブジェクトの動作を決定させるものです
現実をシュミレートするなら、「鳴く」とか「食べる」という動作はメソッドで行います

メソッドは C 言語などの手続き型言語で言う「関数」に位置します
事実、C 言語に近いオブジェクト指向言語である C++ は「メンバ関数」という呼び方をします
ただし、メソッドは関数と異なりクラスの一員なのです
メソッドはクラスに動的な働きを持たせる手段であり、クラスの動作を表します

メソッドもメンバなので、アクセス制御が可能です
この場では、変数同様に自由アクセスの public を指定します
さらにメソッドは値を受け取ることができ、値を返すこともできます
例えば、二つの値を受けとってそれらを比較し大きいほうを返すメソッドなどを作れます
こうすれば、メソッドに値を渡すだけで計算処理はメソッドに任せることができます

type method-name(param-list) { statements... }

これが、基本的なメソッドの宣言方法です
type には、このメソッドが呼び出し元に返す型を指定します
method-name はメソッドの名前を、param-list はメソッドが受け取るパラメータを
そして、statements にこのメソッドの本文を記述します

メソッドが値を返す場合、その値の型を type に指定しますが
メソッドが何も値を返す必要がない場合は void を指定します
そう、私たちがこれまで何度も書いてきた Main() メソッドに指定していた void です
また、何も値を受け取らない場合も括弧を記述する必要があることにも注意してください

メソッドを呼び出す場合も、メンバ変数へのアクセスと同じです

```
class Kitty {
 public string name;
 public void Write() {
  System.Console.WriteLine("名前 = " + name);
 }
}

class Test {
 static void Main() {
  Kitty rena = new Kitty();
  rena.name = "RENA";
  rena.Write();
 }
}
```

rena.Write() で rena オブジェクトの Write() メソッドを呼び出しています
Kitty クラスの Write() メソッドは WriteLine() メソッドを呼び出し name メンバを出力します
Write メソッドはオブジェクトに関連付けられているので
メンバ変数のアクセスにオブジェクトを指定する必要はありません

因みに、異なるクラス間でメンバの名前が衝突することはありません
Kitty クラスで WriteLine() メソッドを宣言しても
Console クラスの WriteLine() メソッドと名前が衝突することはありません

メソッドは値を受け取り、また値を返すことが出きると書きました
受け取る値を引数またはパラメータと呼び、返す値を「戻り値」と呼びます
引数を受け取るには、メソッドの宣言時に受け取る仮引数を指定します
例えば、次のメソッドは文字列を受け取り数値を返すというものです

int method(string str) { ... }

method メソッドの内部では、受け取った引数にアクセスする時は変数 str を使います
複数の引数を受け取る場合は カンマ , で引数を区切ります

int method(int x , int y , int z ...)

メソッドから返す値は return ステートメントで返します
以前は何も値を指定しませんでしたが、return は式を指定できましたね

```
class Calc {
 public int value;
 public int getPercent(int pc) {
  return (int)(value * ((double)pc / 100));
 }
```

```
}

class Test {
 static void Main() {
  Calc obj = new Calc();
  obj.value = 1000;
  System.Console.WriteLine(obj.getPercent(5));
 }
}
```

Calc クラスは value メンバと getPercent() メソッドを持ちます
getPercent() メソッドは value の値のうち指定パーセントを返します
プログラムでは、5 を指定しているので、この関数は value の 5% の値を返します

前のページへ　戻る　次のページへ