**第9章　クラスとインスタンス**

**この章ではJAVAのクラスとインスタンスについて説明します。**

これまで、説明の中で何度か出てきましたが、Javaは「オブジェクト指向」のプログラミング言語です。

この章で学習するクラスは、継承(第11章)、多態性(第12章)と合わせて、オブジェクト指向の3大特徴と言われており重要です。

クラスについて、概念や作成方法をしっかり学習しましょう。

**クラス**

クラスとは、プログラムを実行するための処理をまとめたオブジェクトで、クラスの処理の中にはメンバ変数や処理を実行するためのメソッドがあります。

メンバ変数とはクラス内で使用する変数のことで、メソッドとはメンバ変数などを使って行う処理をまとめたものです。

プログラミングでは、1つのシステムを作るにあたり、何千から何万文字のコードを書きシステムを動かしています。これが一塊で管理されていた場合、修正が必要になったとき修正箇所を見つけるだけでも膨大な手間がかかります。

そこでプログラミングでは、1つのシステムを役割ごとに分けることで、管理をしやすくしています。Javaで使われているオブジェクト指向もシステムを役割ごとに分ける、分け方の1つです。

**1.オブジェクト指向とは**

オブジェクト指向とは、1つ1つのクラスを役割を持った**モノ（object)**として扱い、「モノ、またはモノとモノの関係性」と、「モノの操作」を分けて作り、組み立てて完成させるという考え方です。

１つの**モノ**を **複数の部品** に分けて作る際、 **モノ** が持つ**プロパティ(属性)** や **メソッド(操作)** をクラス内で定義します。

クラスは **「車を作る設計図」** のようなものになります。

**2.インスタンスとオブジェクト**

Javaのようなオブジェクト指向のプログラミング言語では、処理をまとめたクラスを実行するために、そのクラスのオブジェクトを生成する必要があります。

そして、生成したクラスのオブジェクトから、そのクラスのメンバ変数を参照したり、メソッドを呼び出して使用します。

クラスの中のメソッドを呼び出すためには、以下のように記述します。

* //クラス定義
* class Classroom{
* // メソッドの宣言・定義
* void methodSample(){
* 実行処理...
* }
* public static void main(String[] args) {
* //オブジェクトを生成
* Classroomｃ = new Classroom();
* //メソッドの呼出し
* c.methodSample();
* }
* ｝

クラスのオブジェクトを生成する方法は、クラス名のあとにオブジェクト名を書きます。

オブジェクトには new演算子を使って生成する インスタンスを格納します。

なお、クラスのオブジェクトは異なるオブジェクト名でいくつでも生成することができます。

**2-1.クラスを作り方(クラスの定義)**

クラス名は、好きなキーワードで構いません。

注：クラスは常に**大文字**で始まり、**Javaファイルの名前はクラス名と同じにす**る必要があります。

以下の例では、Carというキーワード(名前)のクラスにしました。

次に、**オブジェクト(モノ)** が持つ、**プロパティ(属性)** や **メソッド(操作)** をクラス内で定義します。

例えば、車をオブジェクトとして、考えてみます。

車には、**色などの属性** と、 **アクセルなどの操作** があります。

これをCarクラス内の定義に当てはめていくと、以下のように作成することができます。

* public class Car {
* String color;
* int speed;
* void go(){
* ...
* }
* }

3行目で属性（メンバ変数）として色という文字列の宣言、5行目でメンバ変数speedを宣言しています。

7行目で操作（メソッド）として、「go」メソッドが宣言されています。{ ... }内に「go」メソッドがどのような動作をするのかを書きます。

**2-2.クラスの使用**

クラスを使用する為には、まず、そのクラスにアクセス可能である必要があります。

クラスを使うには、以下のように変数を宣言する必要があります。

* void メソッド() {
* Car car1;
* ～
* }

ローカル変数を宣言すると、初期値は設定されず、そのまま値を代入せずに使おうとするとコンパイルエラーになります。

フィールド（メンバー変数）を定義すると、デフォルトの値はnullになる実体（インスタンス）は、具象クラスに対してnewで生成します。

* Car car1;
* car1 = new Car();
* あるいは
* Car car1 = new Car();

**2-3.インスタンスを作成する**

インスタンスとは、クラスから作成される実体です。

クラスは上記の通り、インスタンスを作成するための設計図のようなものです。

すでにCarという名前のクラスを作成したので、これを使ってインスタンスを作成してみます。

※バッテリーは100あると宣言した属性も追加しました。

※また、操作として宣言した **go** という動作の説明に、バッテリーの消費も追加しました。

**例**

[Car.java]

* package JSample;
* public class Car{
* String color;
* int battery = 100;
* void go() {
* this.battery -= 10;
* System.out.println("バッテリーを10%使用しました");
* System.out.println("現在のバッテリーは、" + battery + "%です");
* }
* void charge() {
* this.battery += 5;
* System.out.println("バッテリーは5％を充電しました");
* System.out.println("バッテリ残量は、" + battery + "%です");
* }
* }

7行目で属性として、元々のバッテリーは100あると宣言されています。

9行目で「go」メソッドが宣言され、{ }内に操作の内容が説明されています。

10～12行目で、「go」メソッドを使用すると、バッテリーの値が10減り（10行目）、11・12行目でバッテリーを10使用したことと、現在のバッテリー残量をそれぞれ出力する、という処理が行われます。

14行目では「charge」メソッドが宣言されており、{ }内に操作の内容が説明されています。

**例**

[JSample9\_2.java]

* package JSample;
* public class JSample9\_2 {
* public static void main(String[] args) {
* Car car1 = new Car();
* car1.color = "red";
* System.out.println("バッテリ残量は、" + car1.battery + "％です");
* car1.go();
* car1.charge();
* }
* }

**実行結果**

元々あるバッテリー量は、100です  
バッテリーを10％使用しました  
バッテリー残量は、90％です  
バッテリーは5％を充電しました

バッテリー残量は、95％です

**練習**

**問題**[JEx9\_1.java]

下記クラスのインスタンスを作成し、メソッドを呼出いし、実行してください。

* package JSample;
* class Employee {
* int employeeNo;
* String name;
* void setEmployeeNo (int newEmployeeNo) {
* employeeNo = newEmployeeNo;
* }
* void setName(String newName) {
* name = newName;
* }
* void dispID() {
* System.out.println(name + "さんの社員番号は " + employeeNo + "です");
* }
* }