

Java プログラミング I

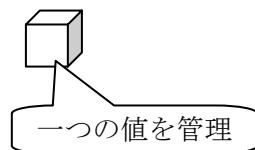
12回目 クラス
2006年7月10日(月)

● クラスとは

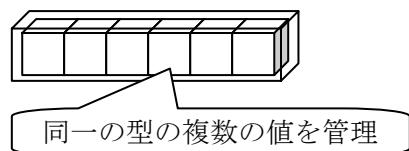
クラス

変数と関数を併せもつデータ型
直観的には、

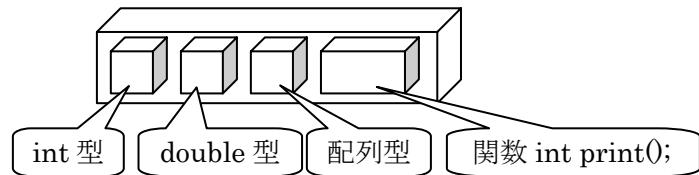
int 型や double 型



配列型



クラス



● クラスの宣言

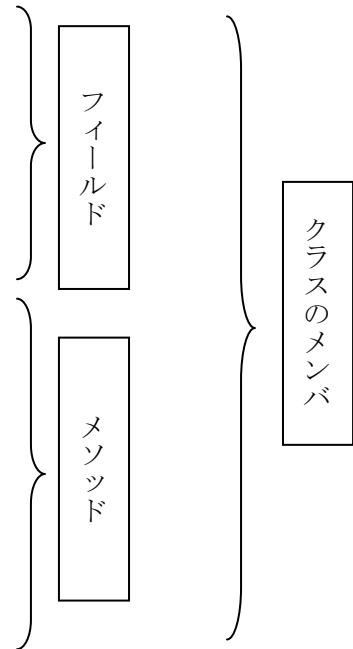
クラスの宣言

クラスのフィールド（変数）とメソッド（関数）を定義して、新しいデータ型として利用できるようにする。
フィールドとメソッドはクラスのメンバと呼ばれる。

クラスの宣言はキーワード `class` を指定して次のように行う

```
class [クラス名]
{
    型 [フィールド名];
    .
    .
    .

    戻り値の型 [メソッド名](引数リスト)
    {
        文;
        ..
        return [式];
    }
    .
    .
}
```



※メソッドは後期の授業で解説する

例えば、ナンバーとガソリン量をフィールドにもつクラス `Car` は次のように宣言できる

```
class Car
{
    int number;      // ナンバーを格納する int 型の変数
    double gas;     // ガソリン量を格納する double 型の変数
}
```

● クラスの利用

クラスの利用手順

クラスのオブジェクト（インスタンス）を扱う変数の宣言



クラスのオブジェクトの生成



オブジェクトの各フィールドの初期化

クラスのオブジェクト

クラスのメンバを格納するための領域

クラスのオブジェクトを扱う変数の宣言は、クラス名と識別子を指定して次のように行う

クラス名 **識別子**;

例えば、

```
Car      car1; // クラス Car 型のオブジェクトを扱う変数を宣言
```

クラスのオブジェクトの生成は、new 演算子とクラス名を指定して次のように行う

識別子 = new **クラス名**();

例えば、

```
car1 = new Car(); // クラス Car 型のオブジェクトを生成
```

オブジェクトがもつ各フィールドは、オブジェクトを扱う変数の識別子とピリオド “.”、フィールド名を用いて参照する。フィールドの初期化は各フィールドを参照して次のように行う。

識別子.**フィールド名** = **値**;

例えば、

```
car1.number = 9129;  
car1.gas=30.0;
```

ソースコード例

ソースファイル名 : Sample12_1.java

```
// 車クラスの宣言とその利用

// クラス Car の宣言
class Car
{
    int number; // ナンバー
    double gas; // ガソリン量
}

class Sample12_1
{
    public static void main(String[] args)
    {
        Car car1;           // クラス Car 型のオブジェクタを扱う変数
        car1 = new Car();   // クラス Car 型のオブジェクトを生成

        // 上記を同時にできる
        // Car car1 = new Car();

        // 各フィールドに値を代入
        car1.number = 9129;
        car1.gas = 30.0;

        // 各フィールドの値を出力
        System.out.println("車のナンバーは" + car1.number + "です。");
        System.out.println("ガソリン量は" + car1.gas + "です。");
    }
}
```

実行画面

```
>java Sample12_1
車のナンバーは 9129 です。
ガソリン量は 30.0 です。
-- Press any key to exit (Input "c" to continue) --
```