１２回目　クラス

今日の講義で学ぶ内容

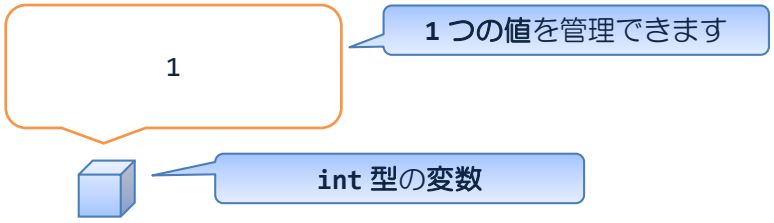
・クラスとは
・クラスの宣言と利用
・クラスの応用

クラス

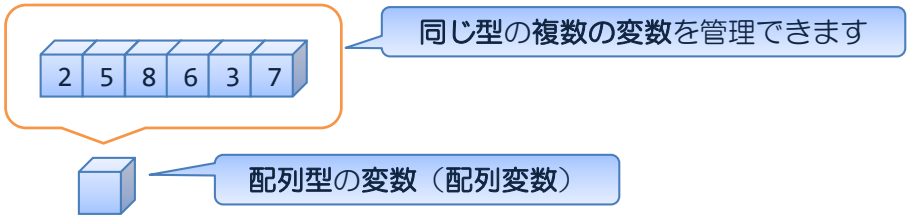
クラスとは

異なる複数の**型**の**変数**を内部にもつ**型**です

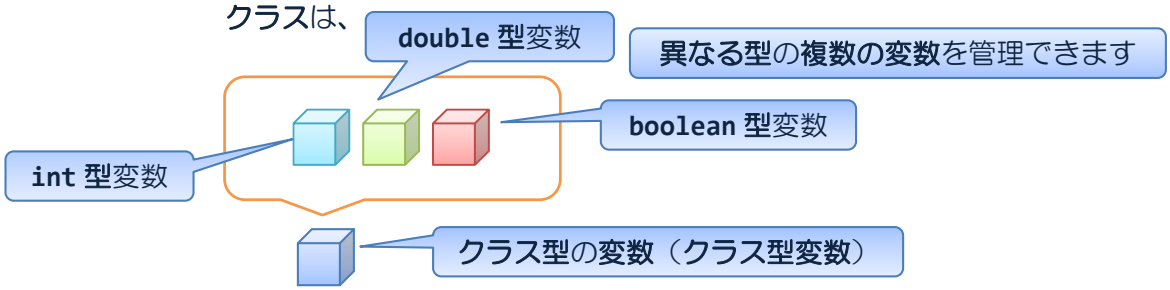
直観的に表現すると、
int 型や **double** 型は



配列型は、



クラスは、



クラスは**変数**の他に、**メソッド**をもちます
メソッドは c 言語での関数に相当します

クラスの宣言

クラスのフィールド（**変数**のことです）と**メソッド**（**関数**のことです）を宣言して、新しい**型**として利用できるようにします

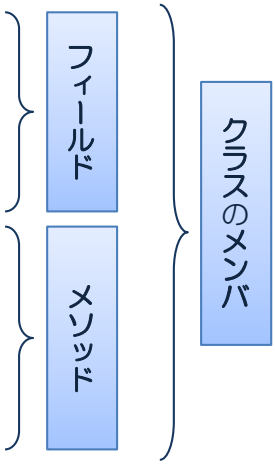
フィールドと**メソッド**は**クラスのメンバ**といいます

クラスの**宣言**はキーワード **class** を指定して次のように行います

```
class クラス名 {クラスのメンバ}
```

クラスの**メンバ**は以下のように宣言します

```
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)
    {
        メソッドの本体
    }
          ⋮
}
```



クラスの**メンバ**は**フィールド**と**メソッド**以外のものを持つこともできます
ここでは基本的な**フィールド**と**メソッド**をおさえておきましょう

メソッドについてはＪａｖａプログラミングⅡで詳しく説明します

たとえば、
車は**ナンバー**と**ガソリン量**をもっています

したがって、
ナンバーと**ガソリン量**を**フィールド**にもつ**クラス Car** は次のように宣言することができます

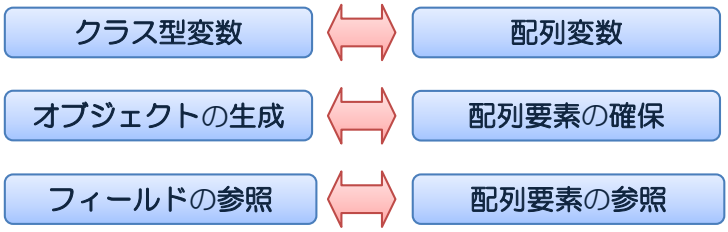
ソースコード例
ソースファイル名：Sample12_1.java

```
// クラス Car の宣言
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
}
```

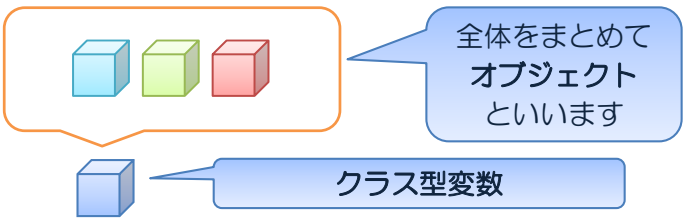
クラスの利用

クラスの利用手順　**クラス型変数の宣言** → **オブジェクトの生成** → **フィールドの参照**


クラスの利用手順を配列の利用手順と次のように
対応させると分かりやすいでしょう



クラス型変数の宣言　**クラス型変数**とは**オブジェクト**を扱う（代入する）**変数**です



クラス型変数は通常の変数と同様に**型**と**識別子**をもちます

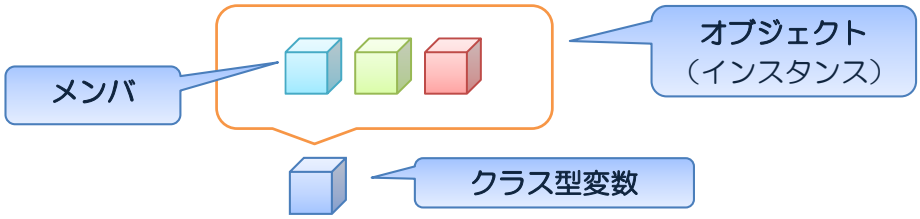
型（**クラス名**）と**識別子**を指定して次のように行います

```
クラス名　識別子;
```

たとえば、
Sample12_1.java で宣言したクラス Car 型の変数（クラス型変数）を宣言するには、

```
Car car1;   // クラス Car 型の変数 car1 を宣言
```

オブジェクトの生成　**オブジェクト**とは**クラス**の**メンバ**を**格納**するための**領域**です
クラスのオブジェクトは**インスタンス**とも呼ばれます



クラス名を指定して次のように行います

```
識別子 = new クラス名();
```

new 演算子は指定されたクラスのオブジェクトをコンピュータのメモリ上に作ります

オブジェクトの**生成**は**丸括弧"()"**で、**配列要素**の**確保**は**角括弧"[]"**です
違いに注意しましょう

たとえば、クラス Car 型のオブジェクトを生成するには、

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

フィールドの参照　**オブジェクト**の中の**フィールド**を**参照**（指定）して**値**を**代入**します

各**フィールド**の**参照**はクラス型変数の**識別子**と**フィールド名**を用いて次のようにします

識別子.フィールド名

識別子と**フィールド名**の間の**ピリオド**は、**○○の中の△△**と解釈するとよいでしょう

フィールドへの**値**の**代入**は、各**フィールド**を**参照**して次のように行います

```
識別子.フィールド名 = 値;
```

たとえば、
クラス Car 型のオブジェクト car1 のフィールド number と gas に値を代入するには、

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

フィールドは**オブジェクト**が**生成**されたときに予め以下の**デフォルト値**が代入されます

（型）	（デフォルト値）
boolean	false
char	0（'¥u0000'）
byte、short、int、long	0
float、double、	0.0
配列変数、クラス型変数	null

※null についてはＪａｖａプログラミングⅡで説明します

ソースコード例
ソースファイル名：Sample12_2.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
}

public class Sample12_2
{
    public static void main(String[] args)
    {
        Car car1;              // クラス Car 型の変数
        car1 = new Car();      // クラス Car 型のオブジェクトを生成

        // 上記を同時に行うこともできる
        // 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。");
    }
}
```

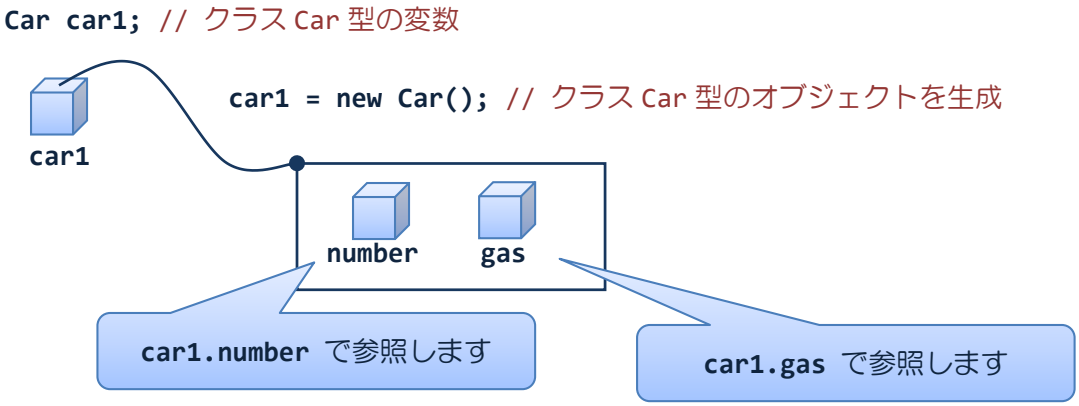
main() メソッドを含むクラスに public を付けて，このクラス名とファイル名を一致させましょう．これにより，インタプリタはどのクラスに main()メソッドがあるかを知ることが出来ます．

実行画面

```
車のナンバーは 9129 です。
ガソリン量は 30.0 です。
```

クラス型変数は参照型変数?基本型変数?

参照型変数には、**配列変数**と**クラス型変数**があります
例題 **Sample12_2** の**クラス型変数**の振る舞いは図的に次のように理解できます



クラスの配列を作ってみましょう

クラス型は **int 型**や **double 型**と同じで**型の 1 つ**です
int 型の配列や double 型の配列と同様に、クラス型の配列を作成することができます

ソースコード例
ソースファイル名：Sample12_3.java

```
// 車クラスの配列

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

public class Sample12_3
{
    public static void main(String[] args)
    {
        Car[] cars;           // クラス Car 型の配列型の変数（配列変数）
        cars = new Car[2];    // クラス Car 型の変数（配列要素）を 2 つ分

        cars[0] = new Car();  // クラス Car 型のオブジェクトを 1 つ生成
        cars[1] = new Car();  // 新たにクラス Car 型のオブジェクトを 1 つ生成

        // 各フィールドに値を代入
        cars[0].number = 9129;
        cars[0].gas = 30.0;

        cars[1].number = 1234;
        cars[1].gas = 15.5;

        // 各フィールドの値を出力
        for(int i=0;i<cars.length;i++)
        {
            System.out.println( i +"番目の車情報：");
            System.out.println("車のナンバーは" + cars[i].number + "です。");
            System.out.println("ガソリン量は" + cars[i].gas + "です。");
        }
    }
}
```

実行画面

```
0 番目の車情報：
車のナンバーは 9129 です。
ガソリン量は 30.0 です。
1 番目の車情報：
車のナンバーは 1234 です。
ガソリン量は 15.5 です。
```

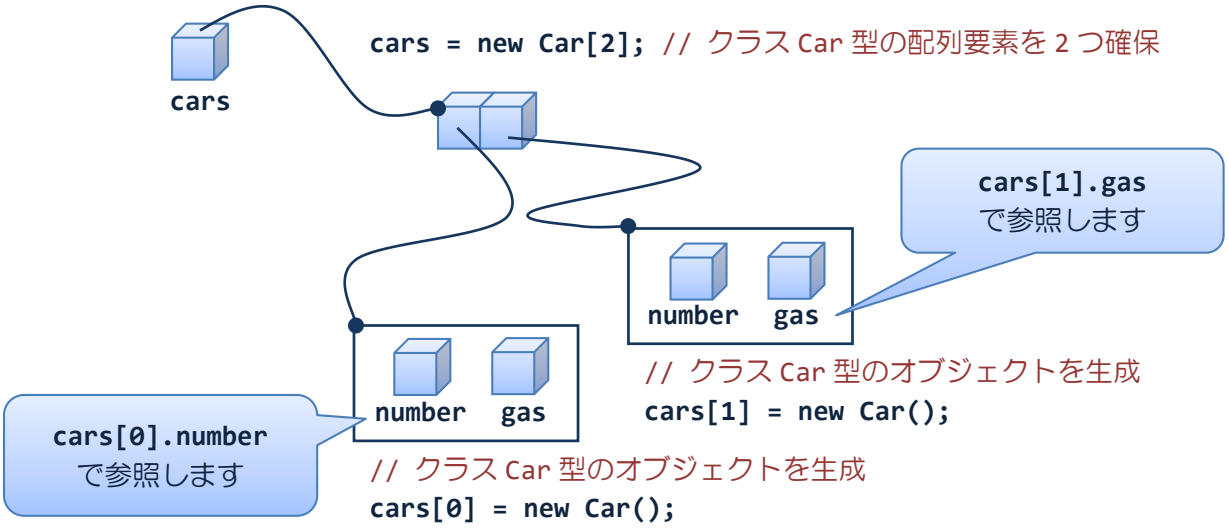
例題 **Sample12_3** の変数の振る舞いは図的に次のように理解できます

【int 型の配列の場合】

```
int[] ary; // int 型の配列型の変数
```

【クラス型の配列の場合】

```
Car[] cars; // クラス Car 型の配列型の変数
```



クラスのメンバにクラス型変数を宣言してみましょう

クラス型は **int 型**や **double 型**と同じで**型の 1 つ**です
int 型や double 型の変数をメンバにできるようにクラス型の変数をメンバにできます

ソースコード例
ソースファイル名：Sample12_4.java

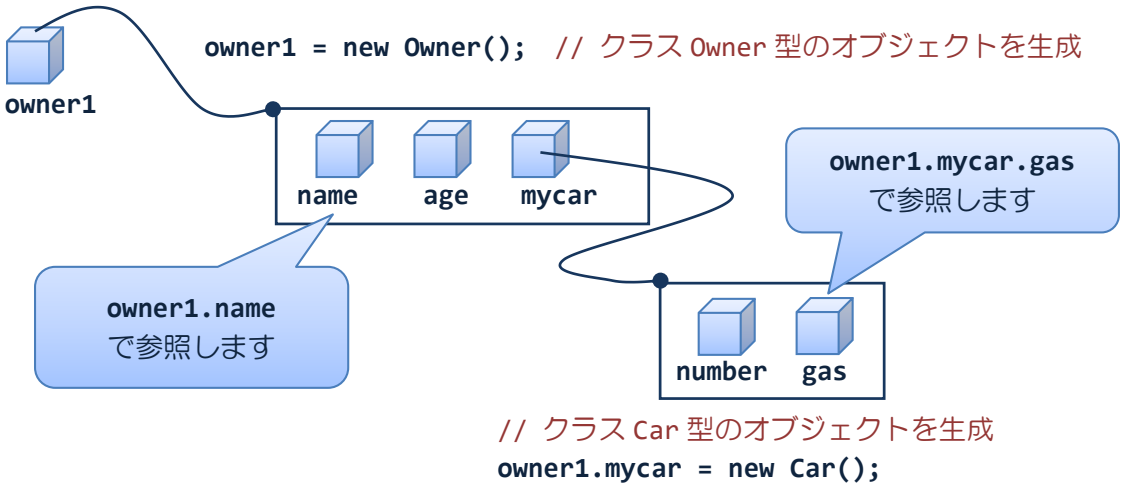
```
// 車クラスを別のクラスのメンバにする
// クラス Car の宣言
class Car
{
    int number;    // ナンバー
    double gas;    // ガソリン量
}
// クラス Car をメンバに持つクラス Owner の宣言
class Owner
{
    String name;
    int age;
    Car mycar; // クラス Car 型の変数をメンバにもつ
}

public class Sample12_4
{
    public static void main(String[] args)
    {
        Owner owner1;              // クラス Owner 型の変数
        owner1 = new Owner();      // クラス Owner 型のオブジェクトを生成

        owner1.name = "Java";
        owner1.age = 21;
        owner1.mycar = new Car(); // クラス Car 型のオブジェクトを生成
        owner1.mycar.number = 9129;
        owner1.mycar.gas = 30.0;

        // 各フィールドの値を出力
        System.out.println("所有者");
        System.out.println("名　前:"+owner1.name);
        System.out.println("年　齢:"+owner1.age);
        System.out.println("車");
        System.out.println("車ナンバー:"+owner1.mycar.number);
        System.out.println("ガソリン量:"+owner1.mycar.gas);
    }
}
```

実行画面

```
所有者
名　前:Java
年　齢:21
車
車ナンバー:9129
ガソリン量:30.0
```

例題 **Sample12_4** の変数の振る舞いは図的に次のように理解できます

```
Owner owner1;    // クラス Owner 型の変数
```



今日の講義のまとめ

・クラスを用いることにより、異なる複数の型の変数を一括して管理できます。

・クラスがもつフィールド（変数）やメソッドをクラスのメンバといいます。

・指定されたメンバをもつ新しいクラス（クラス型）の準備はクラスの宣言により行います。

・クラスを利用するときは、クラス型の変数を宣言し、オブジェクトを確保します。オブジェクトはクラスのメンバを格納するためのメモリ領域です。メンバへのアクセスは、ピリオドを用いて行います。

・クラス型の変数は参照型の変数です。

・クラス型の変数は int 型や double 型と同じように配列を構成できます。