

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



クラスとは

変数と関数を併せもつデータ型です



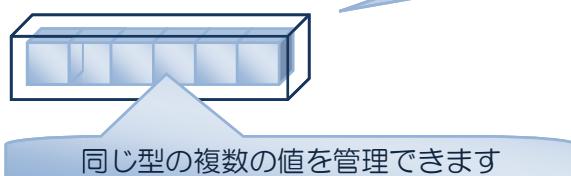
関数については後期で詳しく説明します
ここでは C 言語の関数と同じものと考えてください

直観的に表現すると、
int 型や double 型は



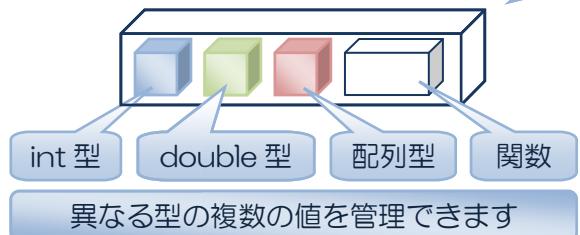
配列型は、

配列型の変数（配列変数）



クラスは、

クラス型の変数



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）を宣言して、新しいデータ型として利用できるようにします

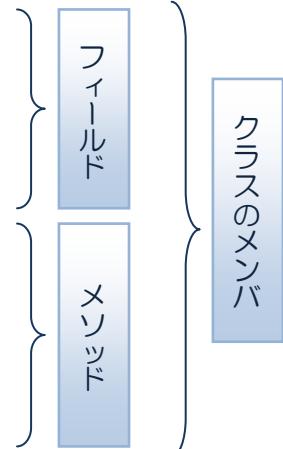
フィールドとメソッドはクラスのメンバといいます

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

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` は次のように宣言すればよいでしょう

ソースコード例

ソースファイル名 : `Sample12_1.java`

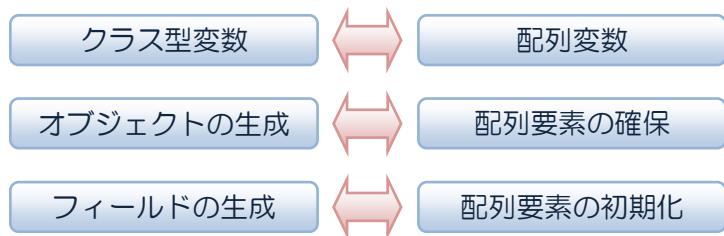
```
// クラス Car の宣言
class Car
{
    int number;      // ナンバーを格納する int 型の変数
    double gas;     // ガソリン量を格納する double 型の変数
}
```

クラスの利用

クラスの利用手順

1. クラス型変数の宣言
2. クラスのオブジェクトの生成
3. オブジェクトのフィールドの初期化

⚠️ 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



1. クラス型変数の宣言 クラス型変数とはクラスのオブジェクトを扱う変数です
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:

例えば、

Car car1; // クラス Car 型の変数を宣言

2. クラスのオブジェクトの生成 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(); // ← 丸括弧“()” ※配列の場合は“[]”です 違いに注意！！

⚠️ 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

例えば、

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

3. オブジェクトのフィールドの初期化 確保したフィールドに適当な値を入れておきます

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、〇〇の中の△△と解釈するとよいでしょう

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います

識別子|フィールド名 = 値;

例えば、

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

⚠️ フィールドは初期化を行う以前に予め以下の値が代入されています

(型) (デフォルト値)

boolean	false
char	‘\u0000’
byte、short、int、long	0
float、double、	0.0
配列変数、クラス型変数	null ※nullについては後期の講義で説明します

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
}

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

        // 上記を同時に実行することもできる
        // 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。");
    }
}
```



CPad に main() メソッドを含むクラス名を知らせるため、main() メソッドを含むクラス名をファイル名と一致させてください

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) --
```

❓ 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 の変数の振る舞いは図的に次のように理解できます

```
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
```

car1

number gas

car1.number で参照します

car1.gas で参照します

⚠️ クラスの配列を作ってみましょう

クラス型は int 型や double 型と同じデータ型の一つです
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
}

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

        car[0] = new Car(); // クラス Car 型のオブジェクトを一つ生成
        car[1] = new Car(); // 新たにクラス Car 型のオブジェクトを一つ生成

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

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

        // 各フィールドの値を出力
        for(int i=0; i<car.length; i++)
        {
            System.out.println((i+1)+"番目の車情報:");
            System.out.println("車のナンバーは" + car[i].number + "です。");
            System.out.println("ガソリン量は" + car[i].gas + "です。");
        }
    }
}
```

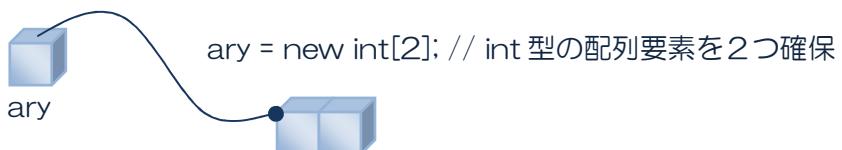
実行画面

```
>java Sample12_3
1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。
2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。
-- Press any key to exit (Input "c" to continue) --
```

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

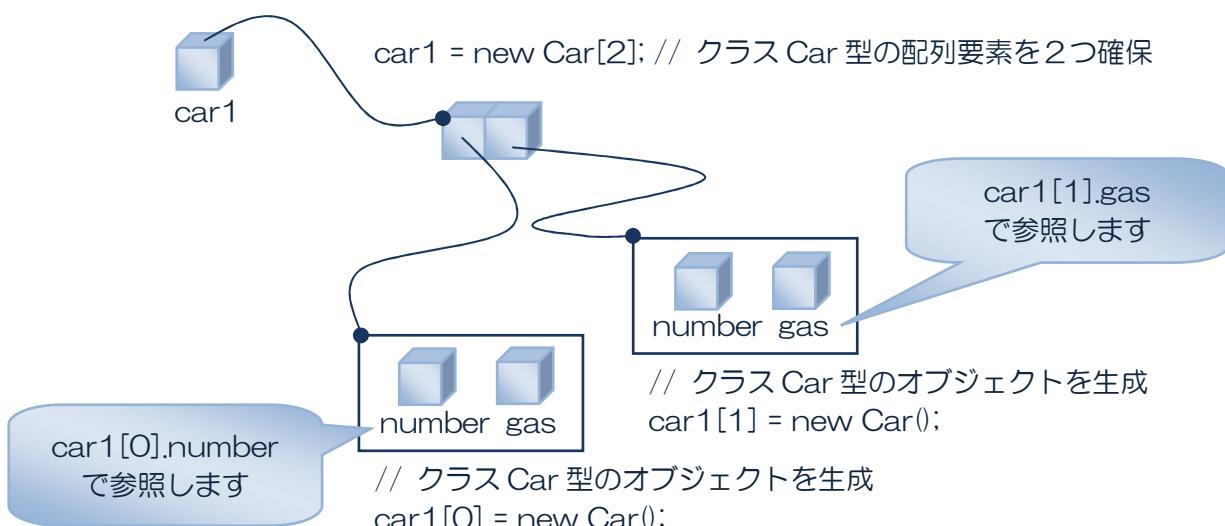
【int型の配列の場合】

```
int[] ary; // int型の配列型の変数
```



【クラス型の配列の場合】

```
Car[] car1; // 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 型と同じデータ型の一つです
int 型の変数や 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
}

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

```
>java Sample12_4
所有者
名 前:Java
年 齢:21
車
車ナンバー:9129
ガソリン量:30.0
-- Press any key to exit (Input "c" to continue) --
```

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
```

