



静的メンバ

応用編2日目

静的メンバとstatic

メンバ：
フィールドとメソッドの総
称

- **静的メンバ**… インスタンスを生成せず利用できるメンバ
 - 静的なメソッド … **クラスメソッド**
 - 静的なフィールド … **クラスフィールド**
- **インスタンスメンバ**… インスタンスを生成して利用するメンバ
 - インスタンスメソッド
 - インスタンスフィールド
- **static(スタティック)**修飾子をつけるだけで定義できる
- 呼び出し方・利用方法には様々な制約がある
- **SampleEx201**参照

呼び出し方のルール

- **インスタンスメンバ**

- インスタンスを生成しないと利用できない
- **(インスタンス名).フィールド名**
- **(インスタンス名).メソッド名(...)**
- 例: `s.a = 1; s.add(1,2);` (`s`:インスタンス名とする)

- **クラスメンバ**

- インスタンスを生成しないでも利用できる
- **(クラス名).フィールド名**
- **(クラス名).メソッド名(...)**
- 例: `Hoge.n = 1; Hoge.mul(2,3);` (`Hoge`:クラス名とする)

Main()メソッド

- **Main()**メソッド … クラスメソッドの一つ
- `main()`メソッドがると、プログラムはそこから実行される
- 引数のString型
 - コンソールから呼び出されるときに追加される文字列

```
Hoge.exe abc def ghi
```

とすると、

```
args[0] = "abc"  
args[1] = "def"  
args[2] = "ghi"
```

クラスメンバの注意点

- **クラスメソッド内でインスタンスメンバは使えない。**
 - インスタンスメンバはインスタンスを生成しなくては使えない。
 - クラスメンバはインスタンスを生成しなくても使うことが出来る。
- **インスタンスメソッドの中でクラスメンバを使うことは可能。**

クラスメンバとインスタンスメンバ①

```
class Data
{
    private static int num = 0;
    private int id;
    public Data(int id)
    {
        this.id = id;
        num++;
        Console.WriteLine("値 : {0} 数:{1}", this.id, num);
    }
    public static void ShowNumber()
    {
        Console.WriteLine("Dataオブジェクトの数:{0}", num);
    }
}
```

クラスメンバ

インスタンス生成を
しないで呼び出す。

クラスメンバとインスタンスメンバ②

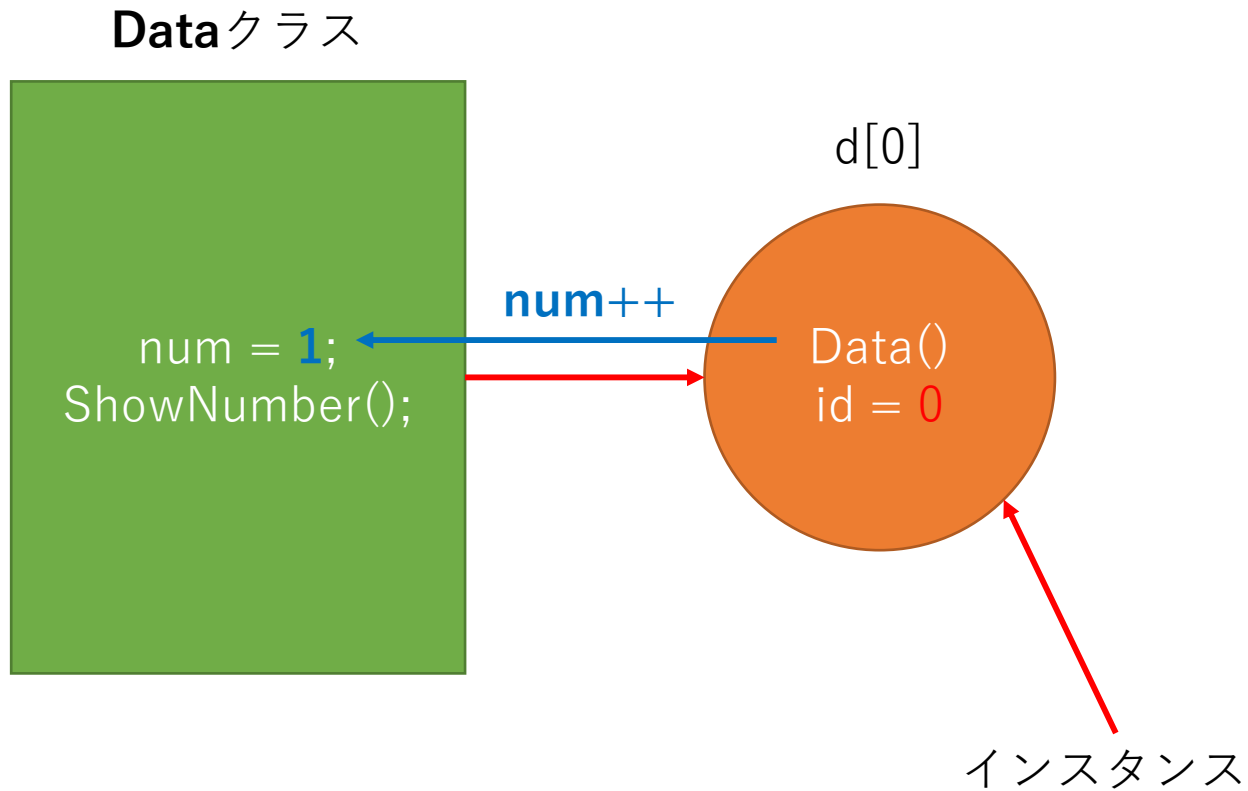
Dataクラス

```
num = 0;  
ShowNumber();
```

インスタンス生成を
しなくても利用可能

クラスメンバとインスタンスメンバ③

```
d[0] = new Data(0);
```



クラスメンバとインスタンスメンバ④

```
d[1] = new Data(100);
```

