



# cp-2. 属性, アクセサ

(C++ オブジェクト指向プログラミング演習,  
全3回)

<https://www.kkaneko.jp/cc/cpp/index.html>

金子邦彦





## メソッド

- **メソッド**は、オブジェクトに属する操作や処理のこと
- **引数（ひきすう）**とは、メソッドに渡す値のこと
- **メソッド**は、**クラス**に属する
- **属性**や**メソッド**にアクセスするときは「.」や「->」を用いる



## アクセサ

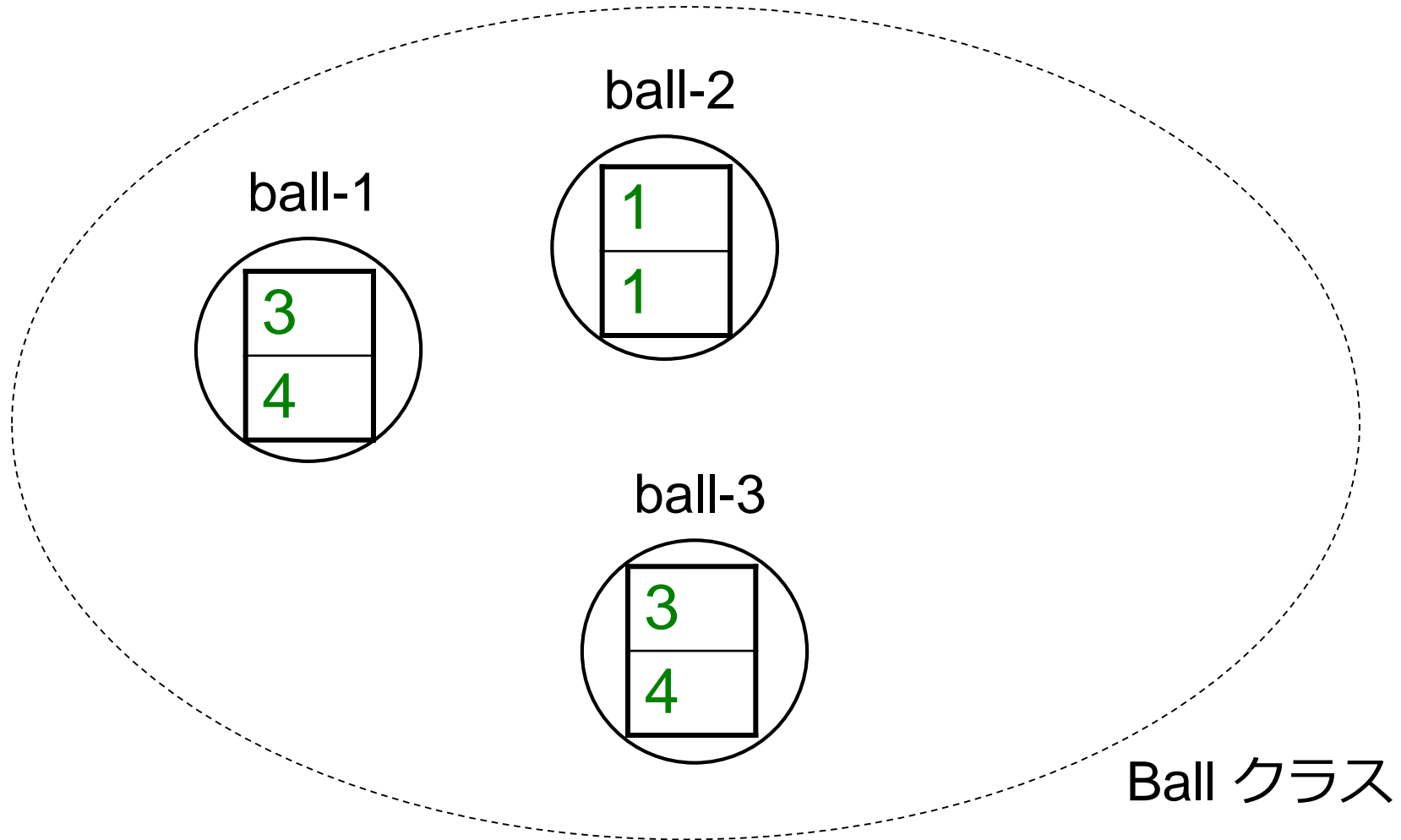
- **アクセサ**は、**属性**値の取得、属性値の更新を行うための**メソッド**



## 例題 1

- Ball クラスのオブジェクトを 3 個生成し, それぞれのメモリアドレスの表示を行う
- オブジェクトを生成したら, オブジェクトアドレスは, ポインタ変数に格納する

# クラスとオブジェクト



# ソースコード



ファイル名: Ball.h

```
#pragma once
class Ball {
public:
    double x, y; } 属性 (メンバ変数ともいう)
public:
    Ball( const double x, const double y );
    Ball( const Ball& ball );
    Ball& operator= ( const Ball& ball );
    ~Ball();
    double distance_to_0() const;
};
```

コンストラクタ,  
デストラクタ

# ソースコード



ファイル名: Ball.cpp

```
#include "Ball.h"
#include <math.h>
Ball::Ball( const double x, const double y ) : x( x ), y( y )
{
    /* do nothing */
}
Ball::Ball( const Ball& ball ) : x( ball.x ), y( ball.y )
{
    /* do nothing */
}
Ball& Ball::operator= (const Ball& ball )
{
    this->x = ball.x;
    this->y = ball.y;
    return *this;
}
Ball::~~Ball()
{
    /* do nothing */
}
double Ball::distance_to_0() const
{
    return sqrt( ( this->x * this->x ) + ( this->y * this->y ) );
}
```

# ソースコード



ファイル名: main.cpp

```
#include <stdio.h>
#include "ball.h"
int main( int argc, char** argv )
{
    Ball* b1 = new Ball( 3, 4 );
    Ball* b2 = new Ball( 1, 1 );
    Ball* b3 = new Ball( 3, 4 );
    fprintf( stderr, "b1: %f, %f\n", b1->x, b1->y );
    fprintf( stderr, "b2: %f, %f\n", b2->x, b2->y );
    fprintf( stderr, "b3: %f, %f\n", b3->x, b3->y );
    delete b1;
    delete b2;
    delete b3;
}
```

-> による属性アクセス



# Visual Studio 2019 C++ での実行結果例



C:\ Microsoft Visual Studio のデバッグ コンソール

```
b1: 3.000000, 4.000000  
b2: 1.000000, 1.000000  
b3: 3.000000, 4.000000
```

```
C:\Users\user\source\repos\ConsoleApp  
ました。  
このウィンドウを閉じるには、任意のキ
```

## 例題 2

- アクセサを定義し使用する

# ソースコード



ファイル名: Ball.h

```
#pragma once
class Ball {
private:
    double _x, _y; } 属性 (メンバ変数ともいう)
public:
    Ball( const double x, const double y );
    Ball( const Ball& ball );
    Ball& operator= ( const Ball& ball );
    ~Ball();
    double distance_to_0() const;
    double x() const { return this->_x; };
    double y() const { return this->_y; }; } アクセサ
};
```

コンストラクタ,  
デストラクタ

アクセサを定義. 属性の読み出しは可能

# ソースコード



ファイル名: Ball.cpp

```
#include "Ball.h"
#include <math.h>
Ball::Ball( const double x, const double y ) : _x( x ), _y( y )
{
    /* do nothing */
}
Ball::Ball( const Ball& ball ) : _x( ball.x() ), _y( ball.y() )
{
    /* do nothing */
}
Ball& Ball::operator= (const Ball& ball )
{
    this->_x = ball.x();
    this->_y = ball.y();
    return *this;
}
Ball::~Ball()
{
    /* do nothing */
}
double Ball::distance_to_0() const
{
    return sqrt( ( this->x() * this->x() ) + ( this->y() * this->y() ) );
}
```

メソッド本体内ではアクセサを使用

# ソースコード



ファイル名: main.cpp

```
#include <stdio.h>
#include "ball.h"
int main( int argc, char** argv )
{
    Ball* b1 = new Ball( 3, 4 );
    Ball* b2 = new Ball( 1, 1 );
    Ball* b3 = new Ball( 3, 4 );
    fprintf( stderr, "b1: %f, %f\n", b1->x(), b1->y() );
    fprintf( stderr, "b2: %f, %f\n", b2->x(), b2->y() );
    fprintf( stderr, "b3: %f, %f\n", b3->x(), b3->y() );
    delete b1;
    delete b2;
    delete b3;
}
```

アクセサによる  
属性アクセス

# Visual Studio 2019 C++ での実行結果例



C:\ Microsoft Visual Studio のデバッグ コンソール

```
b1: 3.000000, 4.000000  
b2: 1.000000, 1.000000  
b3: 3.000000, 4.000000
```

```
C:\Users\user\source\repos\ConsoleApp  
ました。  
このウィンドウを閉じるには、任意のキ
```



- 属性を public にする場合

属性値を意図せず書き換えるリスク

- アクセサを作る場合

アクセサでは、属性値の書き換え不可。

属性値を意図せず書き換えるリスクを抑制。