プログラミング基礎

基本／参照型あれやこれや

基本型と引数：補足

```
static void main(String[] args)
{
   int x;
   x = 10;
   System.out.println("x=" + x);

   changeArgument(x);

   System.out.println("x=" + x);
}

void changeArgument (int a)
{
   a = 0;
}
```

メソッドを使うときに指定した引数
（ここではxのこと）⇒「**実引数**」と呼ぶ

メソッドを定義した（作った）ときの引数
（ここではint aのこと）⇒「**仮引数**」と呼ぶ

基本型と引数：補足

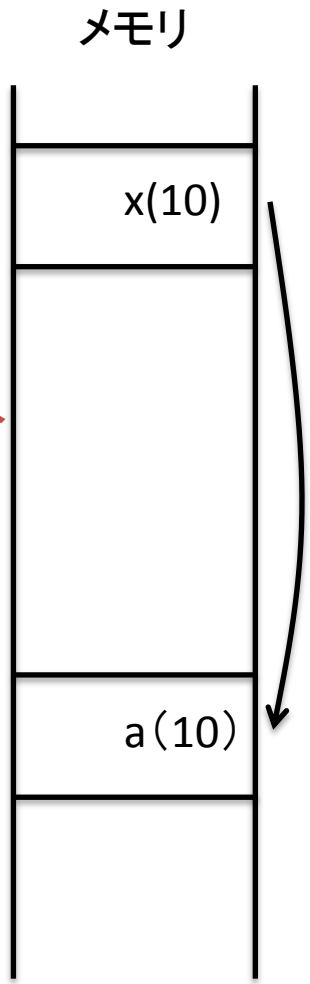
- 基本型の仮引数に対して値を代入しても
  実引数に影響を与えない

  ⇒このような状況を「引数を**値渡し**で渡す」と呼ぶ

基本型と引数：補足

```
static void main(String[] args)
{
   int x;
   x = 10;
   System.out.println("x=" + x);

   changeArgument(x);

   System.out.println("x=" + x);
}

void changeArgument (int a)
{
   a = 0;
}
```

プログラムがここへ到達したとすると



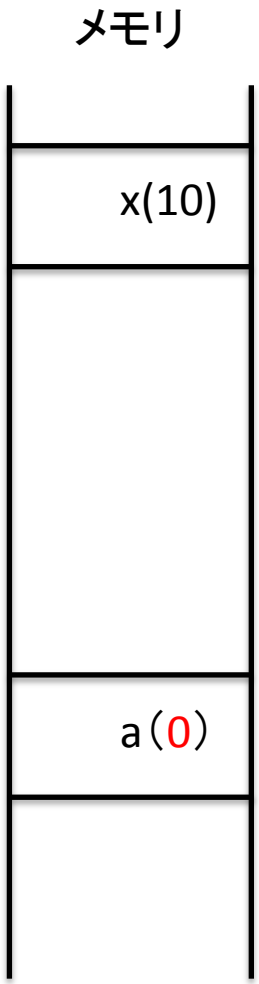
基本型の引数は，
実引数の値を，

一時的に作られた
仮引数名の変数へ
コピーする

基本型と引数：補足

```
static void main(String[] args)
{
   int x;
   x = 10;
   System.out.println("x=" + x);

   changeArgument(x);

   System.out.println("x=" + x);
}

void changeArgument (int a)
{
   a = 0;
}
```

この行が実行されたとする

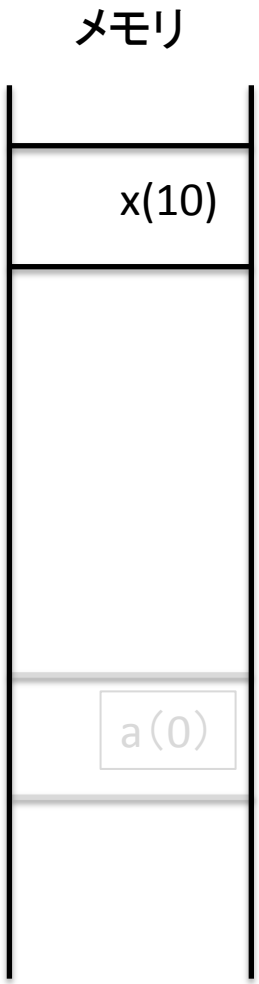


基本型と引数：補足

```
static void main(String[] args)
{
    int x;
    x = 10;
    System.out.println("x=" + x);

    changeArgument(x);

    System.out.println("x=" + x);
}

void changeArgument (int a)
{
    a = 0;
}
```

この行が終わったとする

⇒仮引数に対して値を代入しても
　実引数（ここではxのこと）
　に影響を与えない
⇒「**値渡し**」と呼ぶ



⇒xの値は変化しない

メソッドのために
作られた変数は
自動的に消滅する

参照型と引数:補足

- 参照型の仮引数に対する操作は,
  実引数に対する操作と同等であることに
  注意する

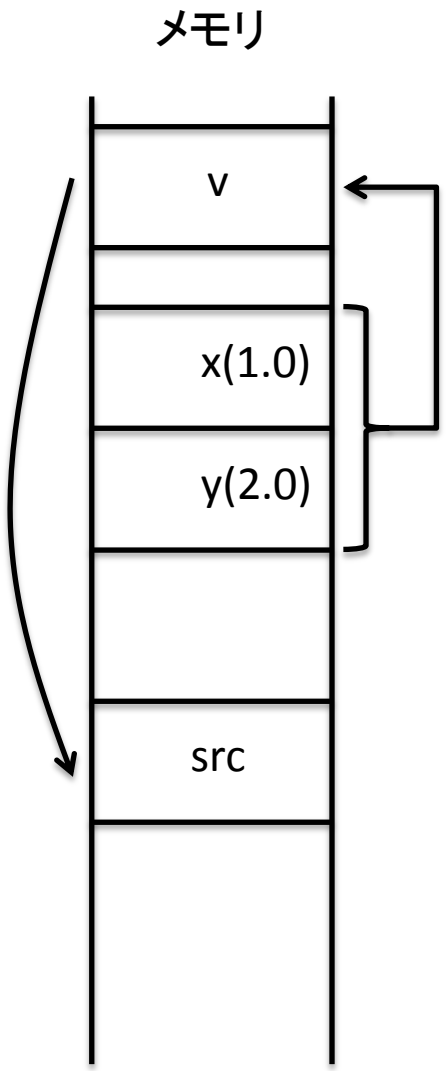
参照型と引数：補足

```
static void main(String[] args)
{
   Vector2D v;
   v = new Vector2D(1.0, 2.0);
   System.out.println("v=(" + v.x + ", " + v.y + ")");

   changeArgument(v);

   System.out.println("v=(" + v.x + ", " + v.y + ")");
}

void changeArgument(Vector2D src)
{
   src.x = 0.0;
}
```

プログラムがここへ到達したとすると

参照型の引数は
実引数**のアドレス**を

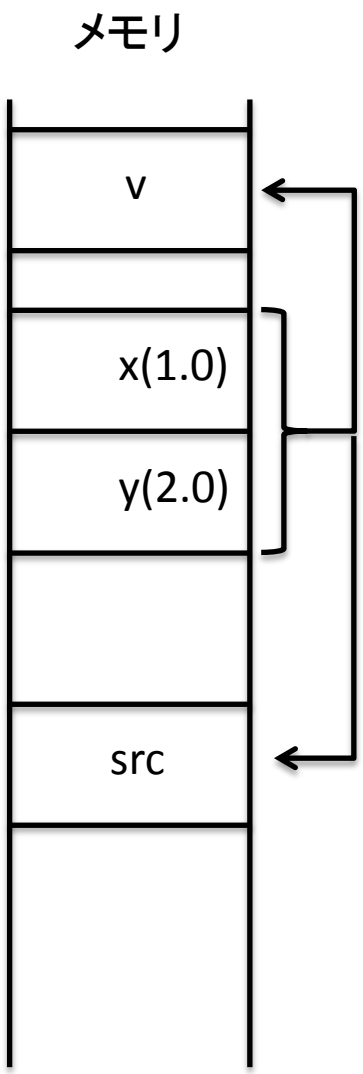
一時的に作られた
仮引数名の変数へ
コピーする



参照型と引数：補足

```
static void main(String[] args)
{
    Vector2D v;
    v = new Vector2D(1.0, 2.0);
    System.out.println("v=(" + v.x + ", " + v.y + ")");

    changeArgument(v);

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

void changeArgument(Vector2D src)
{
    src.x = 0.0;
}
```

プログラムがここへ到達したとすると

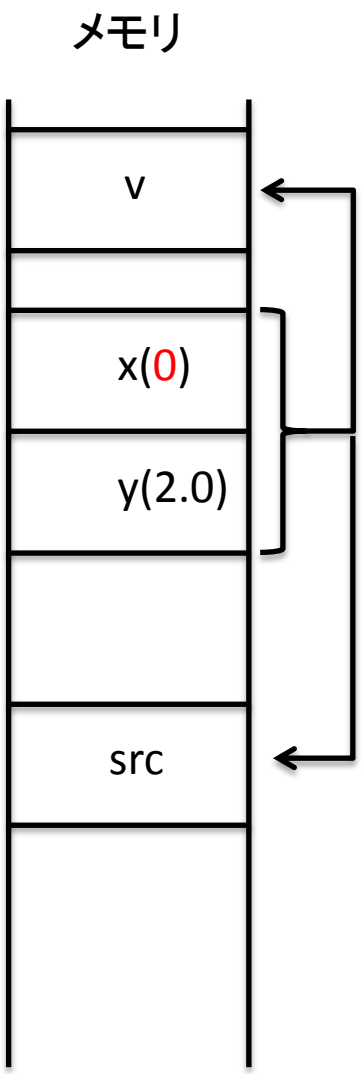


結果vとsrcは同じインスタンスを指している

参照型と引数：補足

```
static void main(String[] args)
{
    Vector2D v;
    v = new Vector2D(1.0, 2.0);
    System.out.println("v=(" + v.x + ", " + v.y + ")");

    changeArgument(v);

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

void changeArgument(Vector2D src)
{
    src.x = 0.0;
}
```

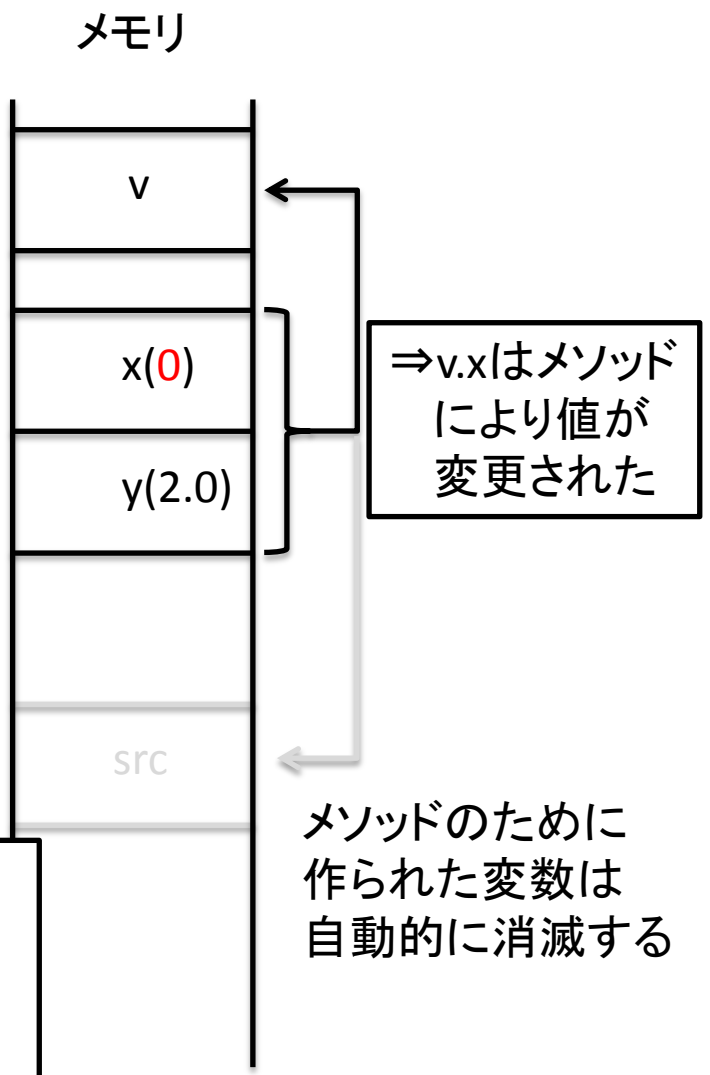
プログラムがこの行を
実行したとする



参照型と引数：補足

```
static void main(String[] args)
{
    Vector2D v;
    v = new Vector2D(1.0, 2.0);
    System.out.println("v=(" + v.x + ", " + v.y + ")");

    changeArgument(v);

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

void changeArgument(Vector2D src)
{
    src.x = 0.0;
}
```

プログラムがこの行を
を終えたとする



⇒参照型の仮引数に対する操作は
実引数に対する操作と
同等であることに注意する
（この行はv.x = 0.0;と同じということ）

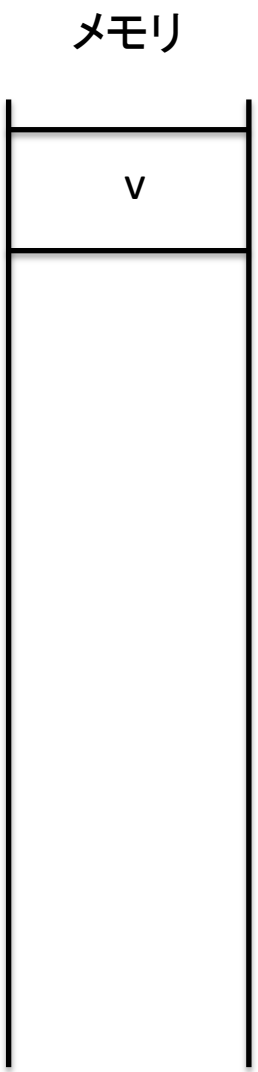
参照型と戻り値

- 参照型を戻り値とすることも当然可能である

参照型と戻り値

```
static void main(String[] args)
{
    Vector2D v;
    v = createZeroVector();

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

Vector2D createZeroVector()
{
    Vector2D tv;
    tv = new Vector2D(0.0, 0.0);
    return tv;
}
```

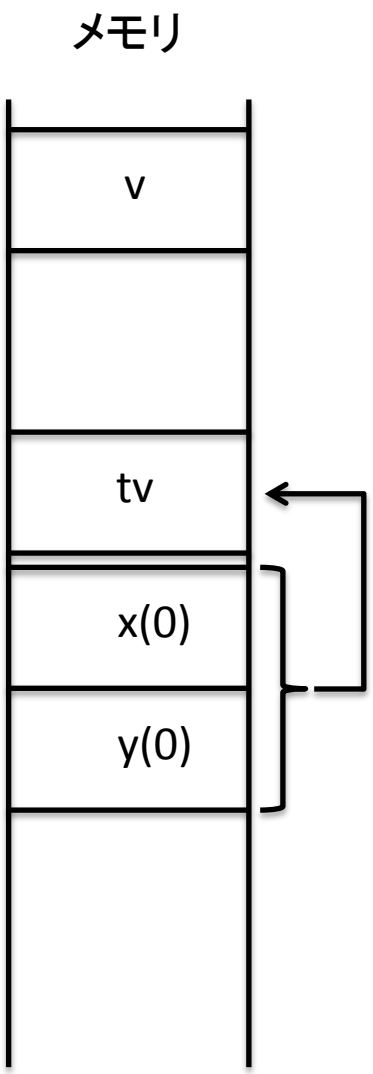
プログラムがこの行を
を終えたとする



参照型と戻り値

```
static void main(String[] args)
{
    Vector2D v;
    v = createZeroVector();

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

Vector2D createZeroVector()
{
    Vector2D tv;
    tv = new Vector2D(0.0, 0.0);
    return tv;
}
```

プログラムがこの行を
を終えたとする

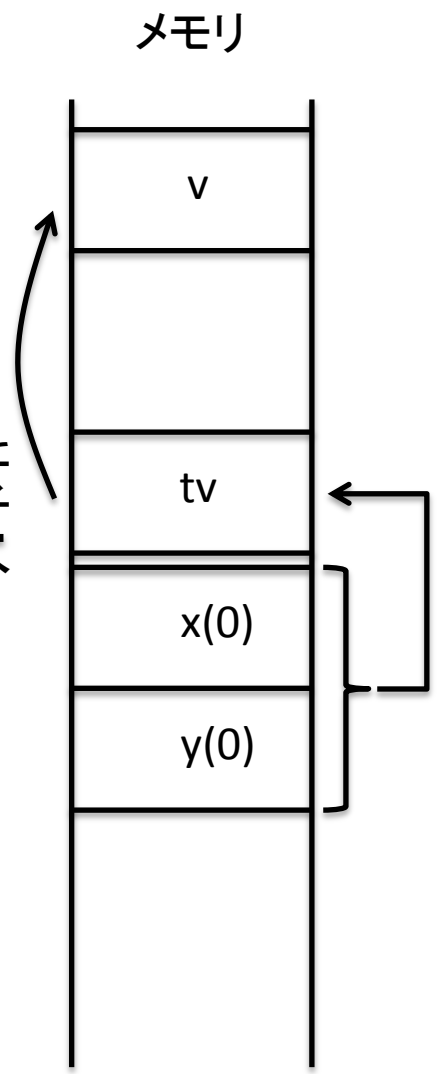


参照型と戻り値

```
static void main(String[] args)
{
    Vector2D v;
    v = createZeroVector();

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

Vector2D createZeroVector()
{
    Vector2D tv;
    tv = new Vector2D(0.0, 0.0);
    return tv;
}
```

プログラムがcreateZeroVector
メソッドを終えたとすると

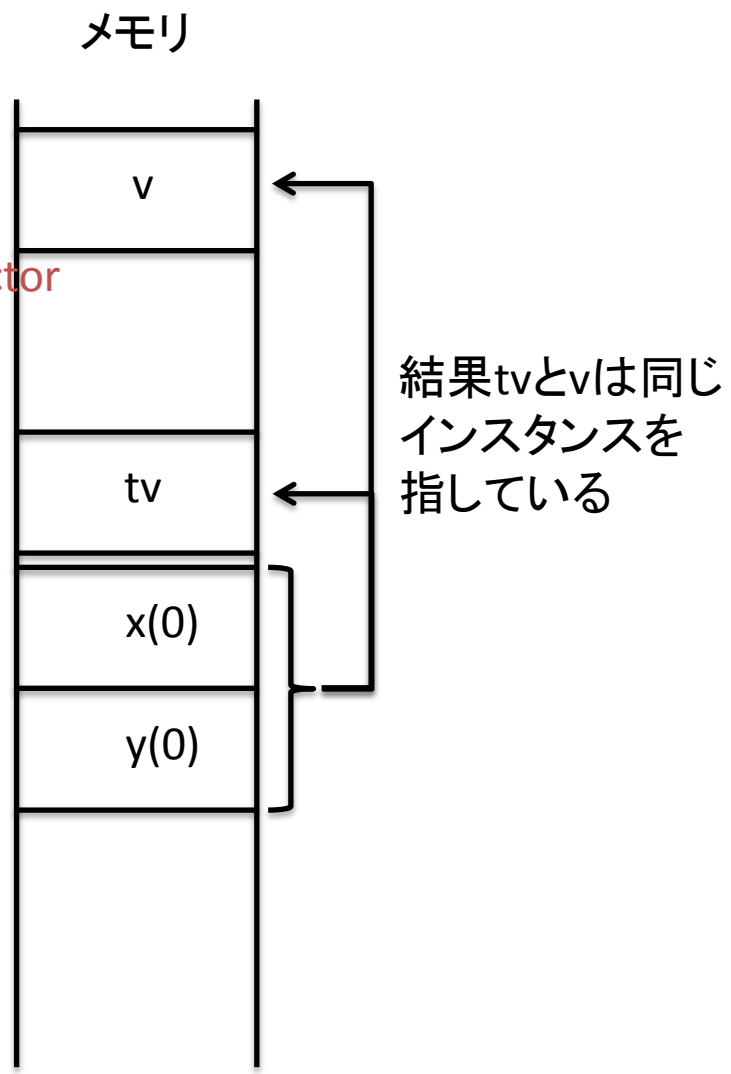
returnに指定された
値（アドレス）が，代入演算子
によりvへ



参照型と戻り値

```
static void main(String[] args)
{
    Vector2D v;
    v = createZeroVector();

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

Vector2D createZeroVector()
{
    Vector2D tv;
    tv = new Vector2D(0.0, 0.0);
    return tv;
}
```

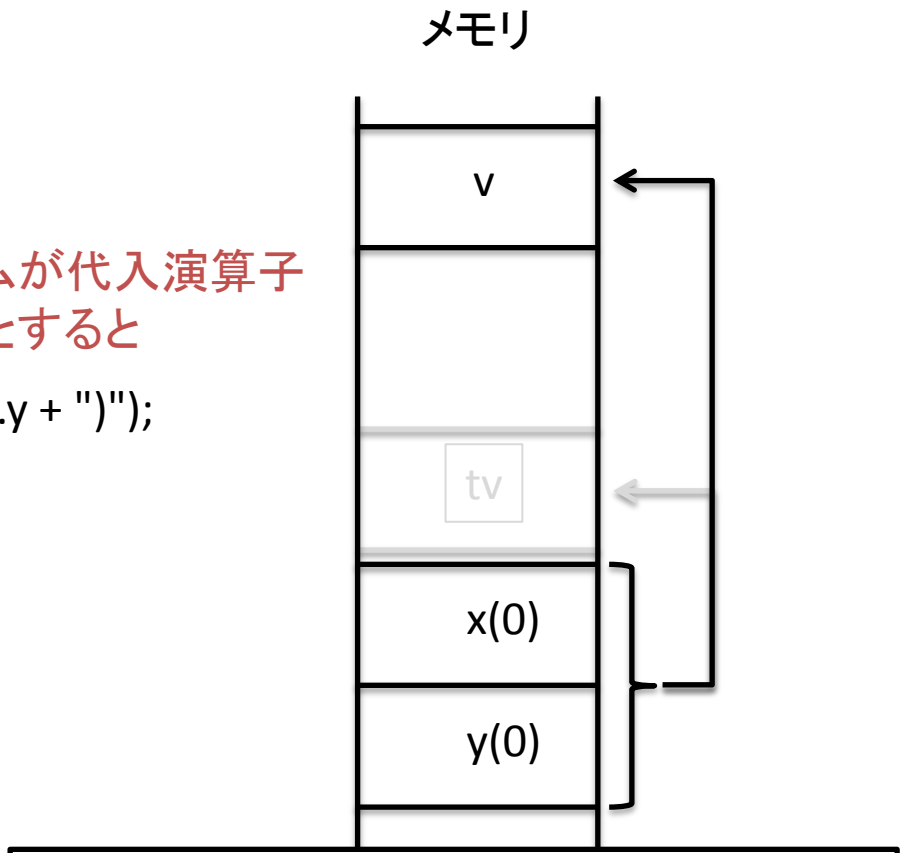
プログラムがcreateZeroVector
メソッドを終えたとすると



参照型と戻り値

```
static void main(String[] args)
{
    Vector2D v;
    v = createZeroVector();

    System.out.println("v=(" + v.x + ", " + v.y + ")");
}

Vector2D createZeroVector()
{
    Vector2D tv;
    tv = new Vector2D(0.0, 0.0);
    return tv;
}
```

プログラムが代入演算子を終えたとすると



メソッドが終了したので，tvは消滅するが，
メソッド内で作られたインスタンス
のアドレスをvが持っているので，
インスタンスが消えることは無い