

クラス定義とオブジェクトの 生成、使用

目次

- 1 インスタンス化(クラス定義、オブジェクト)
- 2 コンストラクタ(オーバーロード)
- 3 **static**イニシャライザ
- 4 **値コピーと参照情報コピー**

1 インスタンス化 (クラス定義、オブジェクト)

```
Class Sample1{
```

```
1--  int  A;
```

```
2--  void setA(int i){  
    A=i;
```

```
}
```

```
Class Sample2{
```

```
    public static void main(~~){
```

```
3--  Sample1 x=new Sample();
```

```
4--      x.setA(100);
```

```
}
```

```
}
```

2 コンストラクタ

```
Class Sample3{
```

```
1-- void Print(){  
    System~();  
}
```

```
2-- void Print(int a,int b){  
    System~();  
}
```

```
3-- void Print(String c){  
    System~() ;  
}  
}
```

```
Class Sample4{
```

```
    public static void main(~~){  
        Sample3 x=new Sample();  
        x.Print();  
        x.Print(100,200);  
        x.Print("aiueo");  
    }  
}
```


3 staticイニシャライザ

```
class Sample5{  
    static{  
1-- System.out.print(“イニシャライザ”);  
    }  
    System.out.print(“コンストラクタ”);  
}
```


4 値コピーと参照情報コピー(1/2)

○基本データ型の値をメソッドに渡される場合

```
1 int num = 10;  
2 obj.method(num);  
3 System.out.print  
4 (num);  
:  
:  
:  
:
```



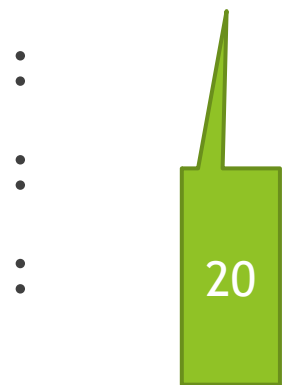
```
1 Void method1(int num){  
2 num += 10;  
3 System.out.print  
4 (num);  
:  
:  
:  
:
```



4 値コピーと参照情報コピー(2/2)

○参照データ型の値をメソッドに渡される場合

```
1 int[] array=(10,30);  
2 Obj.method2(array);  
3 System.pout.print  
4 (array[0]);
```



```
1 Void method2(int[] array){  
2 array[0] +=10;  
3 System.pout.print  
4 (array[0]);
```

