



独習 Java (第3版)

5.8 ローカル変数と変数のスコープ

5.9 メソッドのオーバーロード

5.10 引数の引き渡し

発表者

佐々木研 田島勇樹



5.8 ローカル変数と変数のスコープ(1)

■ ローカル変数

< 特徴 >

- ・メソッド内部で宣言され、そのメソッドの実行中にしか存在しない。
- ・複数のメソッドで同じ名前のローカル変数を使用可能。
ただし、同じ名前でもまったく別物。
- ・ローカル変数には既定値が代入されない。
代入せず値を参照するとコンパイルエラーになる。
- ・ローカル変数はスコープ内でしか使えない。



5.8 ローカル変数と変数のスコープ(2)

■ スコープ

その変数に直接アクセスできるプログラム内の範囲
Javaの場合ブロックがスコープに相当

- ・ローカル変数はブロックの入口({)で作成、出口(})で破棄される。 ローカル変数の値は使い捨て
- ・メソッドに引数として渡されるパラメータはローカル変数と同様メソッド内でのみ有効



5.8 ローカル変数と変数のスコープ(3)

- ・ローカル変数は静的変数またはインスタンス変数と同じ名前で宣言可能。

ローカル変数は静的変数、インスタンス変数を隠す。
その名前を使うとローカル変数にアクセス

ステートメントブロック内で宣言可能

例do,for,whileループ構造内 ブロック内でのみ使用

5.8 ローカル変数と変数のスコープ(4)

- ローカル変数が静的変数、インスタンス変数を隠すプログラム

```
class Myobject{
static short s=400;//静的変数
int i=200;//インスタンス変数
void f(){
system.out.println("s="+s);
system.out.println("i="+i);
short s=300;//ローカル変数
int i=100;//ローカル変数
system.out.println("s="+s);
system.out.println("i="+i);
}
}
```

```
class LocalVariables{
public static void main(String args[]){
MyObject myObject=new MyObject();
myObject.f();
System.out.println("s="+myObject.s);
System.out.println("i="+myObject.i);
}
}
```

出力結果

```
s=400
i=200
s=300
i=100
s=400
i=200
```

5.8 ローカル変数と変数のスコープ(5)

・ステートメントブロック内での宣言の例 出力結果

例 for(int j=0;j<5;j++){	j=0 k=100
//jはローカル変数	j=1 k=100
int k=100;//kはローカル変数	j=2 k=100
System.out.println("j="+j+" k="+k);	j=3 k=100
}	j=4 k=100

forループの外側で変数j,kを
参照するとコンパイルエラー



5.9 メソッドのオーバーロード(1)

- ・メソッドのオーバーロードとは・・・

名前が同じでシグネチャが異なるメソッドが複数あること

名前とシグネチャが同一であるメソッドが2つある コンパイルエラー

- ・基本的な用途が同じであるメソッド群に同一の名前をつけられる

例 println()メソッド データ型 boolean,char,int,long・・・

- ・既定のパラメータ値を簡単に用意できる。

例 必須のパラメータが1つ、省略可能のパラメータが2つのメソッド

パラメータの違う3つのメソッドのオーバーロードを定義可能

ただし、省略可能引数を指定しない場合それがないものとして呼び出す

5.9 メソッドのオーバーロード(2)

・メソッドのオーバーロードのプログラム

```
class Suuji{
void move(int a){
System.out.println("a="+a);
}
void move(int a,int b){
System.out.println("a="+a);
System.out.println("b="+b);
}
void move(int a,int b,int c){
System.out.println("a="+a);
System.out.println("b="+b);
System.out.println("c="+c);
}
}
```

```
class Jisuu{
public static void main(String args[]){
Suuji s=new Suuji();
s.move(1);
s.move(1,2);
s.move(1,2,3);
}
}
```

出力結果

```
a=1
a=1
b=2
a=1
b=2
c=3
```

5.10 引数の引き渡し(1)

■ 引数の引渡し

- ・すべてのメソッド引数は値で渡される。
メソッドには引数のコピーが渡される。
つまりコピーを変更してもメソッド外部にはわからない。
- ・引数が配列またはオブジェクトの場合
参照だけがコピーされる。
配列またはオブジェクトの変更はメソッド外部でもわかる。
- ・引数はスタックに積まれて引き渡される。
引数を呼び出す　スタックに格納(プッシュ)
メソッドの処理から戻る　取得(ポップ)

5.10 引数の引き渡し(2)

```
class ArrayArgument{
public static void main(String args[]){
int x[]={11,12,13,14,15};
display(x);
change(x);
display(x);
}
public static void change(int x[]){
int y[]={21,22,23,24,25};
x=y;
}
```

```
public static void display(int x[]){
for(int i=0;i<x.length;i++)
System.out.print(x[i]+" ");
System.out.println("");
}
}
```

出力結果

```
11 12 13 14 15
11 12 13 14 15
```



練習問題

- ・ コマンドラインで数字を1つ入力したらその数字を表示するメソッド、数字を2つ入力したらその数字間の距離を測るメソッド、数字を3つ入力したら立法体の体積を測るメソッドの3つを作成せよ。ただしメソッドのオーバーロードをすること。
- ・ 5.10 引数の引き渡し(2)のプログラムで、関数の実行後に出力結果の値が変わるようにすること。