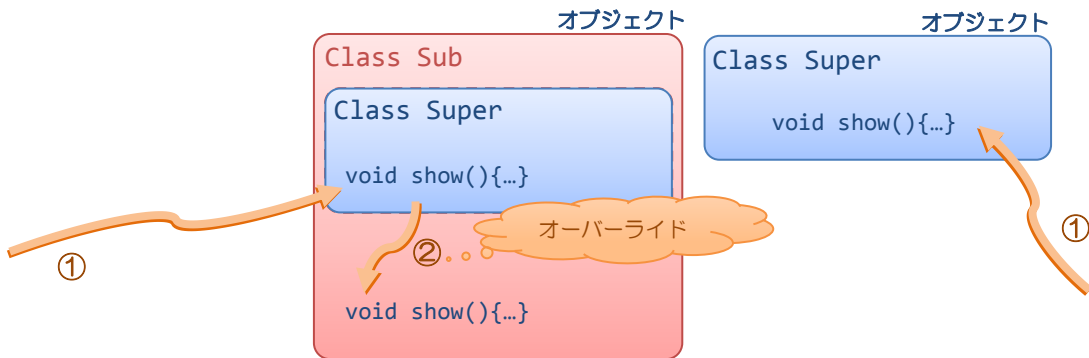


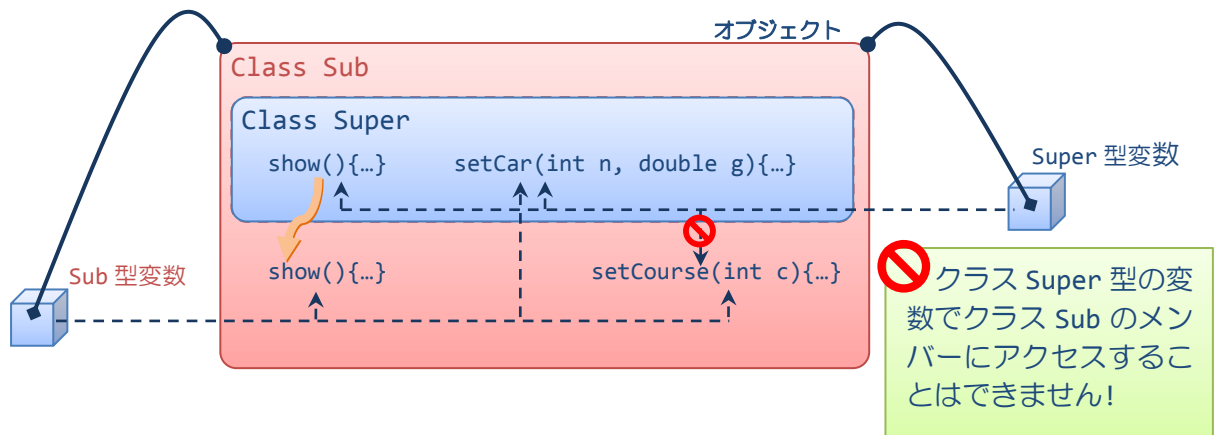
### § メソッドのオーバーライド

- **オーバーライド** スーパークラスのメソッドが呼ばれたとき、サブクラスのメソッドが代わりに実行される機能です
- **宣言** スーパークラスのメソッドと同じメソッド名、引数の数/型をもつメソッドをサブクラスで宣言します  
ただし、戻り値は次にしたいがいます※1
  - 基本型変数の場合 一致
  - 参照型変数の場合 一致、またはサブタイプ
- **動作** スーパークラスのメソッドが呼ばれたとき、実際のオブジェクトの型に依存してメソッドが選択され実行されます  
→ **ポリモーフィズム (多様性)** を実現



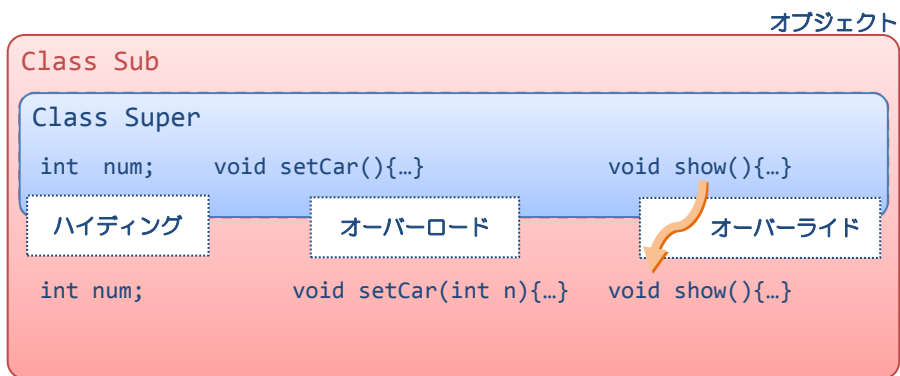
### § スーパークラス型変数とメンバーアクセスの関係

- **スーパークラスのクラス型変数** スーパークラスのオブジェクトだけでなく、そのサブクラスのオブジェクトも代入ができます
- **クラス型変数とメンバーアクセス** 使用するクラス型変数の型のオブジェクトへの参照としてメンバーにアクセスします



## § サブクラスでスーパークラスと同じ名前のメンバーを宣言

- 同じ名前の変数（型に関係なく）を宣言したら、  
→ スーパークラスの変数は、サブクラスの変数でハイディング（覆い隠す）されます
- 同じ名前、同じ引数の数/型（戻り値は\*1の通り）のメソッドを宣言したら、  
→ スーパークラスのメソッドとサブクラスのメソッドはオーバーライドの関係になります
- 同じ名前、異なる引数の数/型（戻り値に関係なく）のメソッドを宣言したら、  
→ スーパークラスのメソッドとサブクラスのメソッドはオーバーロードの関係になります

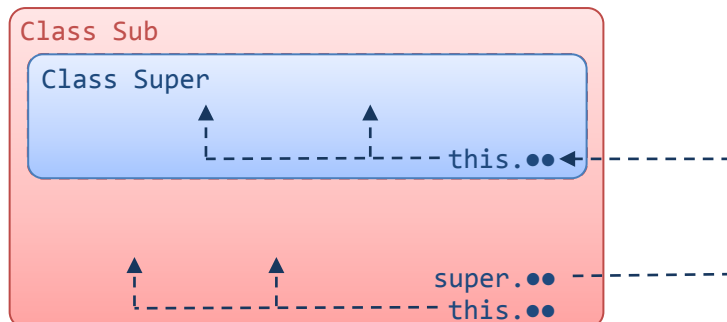


## § サブクラスからスーパークラスと同じ名前のメンバーへアクセス

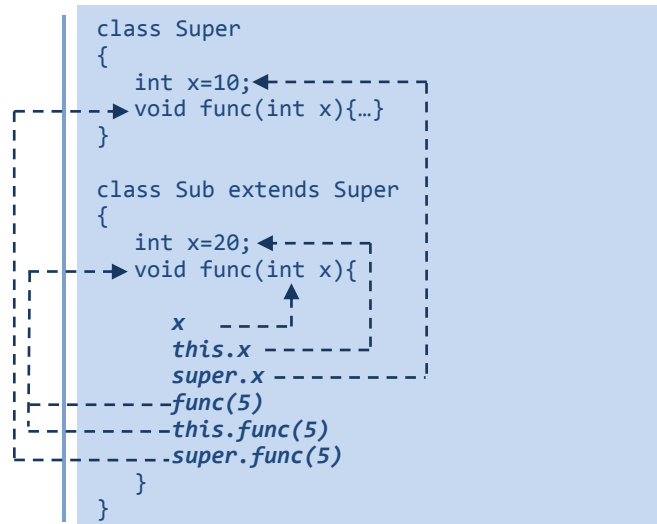
- `super.` 修飾子 スーパークラス型のオブジェクトへの参照としてメンバーにアクセスします

但し、`super.` 修飾子を用いる場合オーバーライドは行われません  
詳しく FAQ を参照して下さい

- `this.` 修飾子 このクラス型のオブジェクトへの参照としてメンバーにアクセスします



- 重複する識別子の区別 継承、またはフィールド、ローカル変数、パラメータなどによる変数やメソッドの識別子の重複は、**super.**と**this.**修飾子を用いて区別することができます



※識別子が重複しない場合、**super.**や**this.**修飾子は必ずしも必要ではありません

## § クラスの拡張やオーバーライドの制限

- **final** 修飾子 修飾される対象に応じて次のように機能を制限することができます
- クラス → そのクラスの拡張ができません
  - 変数 → その変数の値を変更できません (定数)
  - メソッド → そのメソッドをオーバーライドできません