

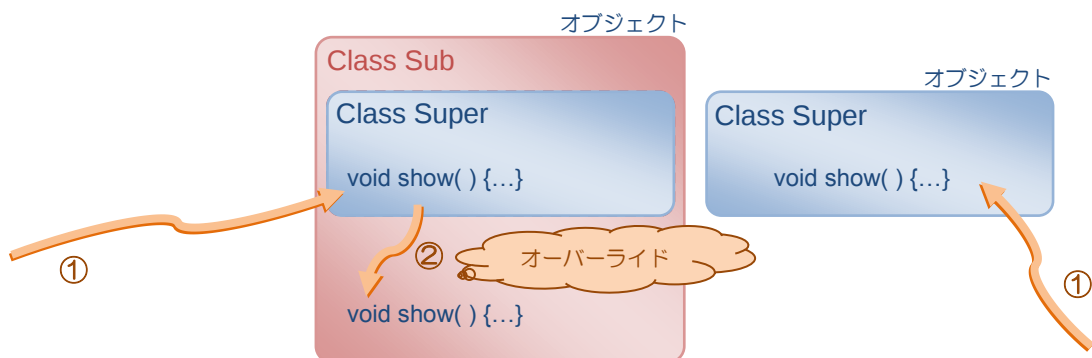
メソッドのオーバーライド

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って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

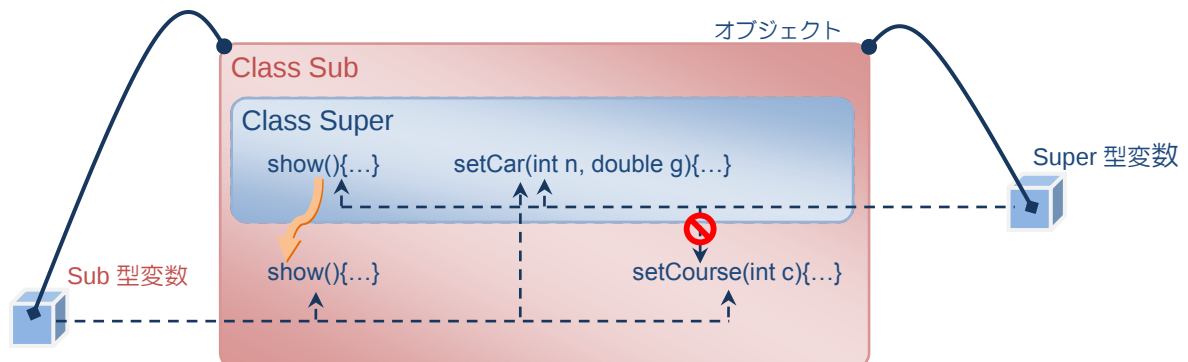
動作 オブジェクトの型に依存してメソッドが選択され実行されます
➔ ポリモーフィズム（多様性）を実現



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

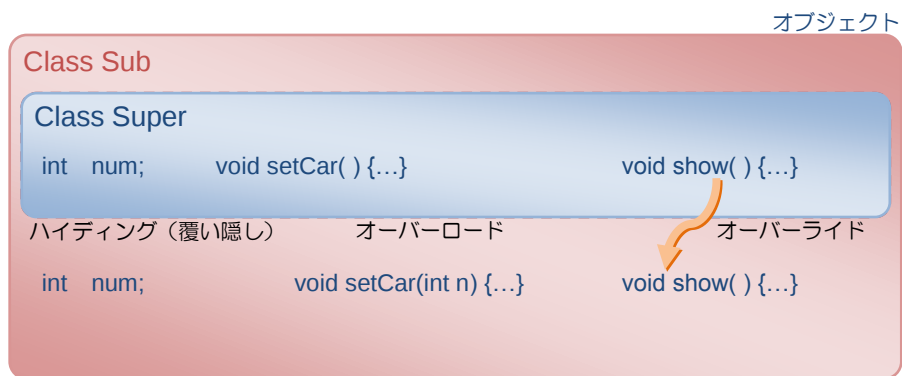
クラス型変数とメンバアクセス 使用するクラス型変数の型のオブジェクトへの参照としてメンバーにアクセスします



※クラス Super 型の変数でクラス Sub のメンバーをアクセスすることはできません

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

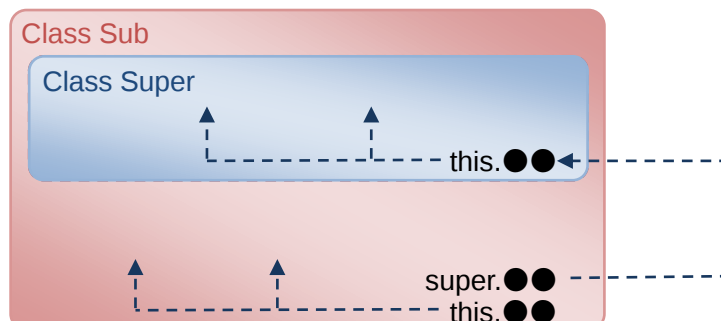


サブクラスでスーパークラスと同じ名前のメンバーへアクセス

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

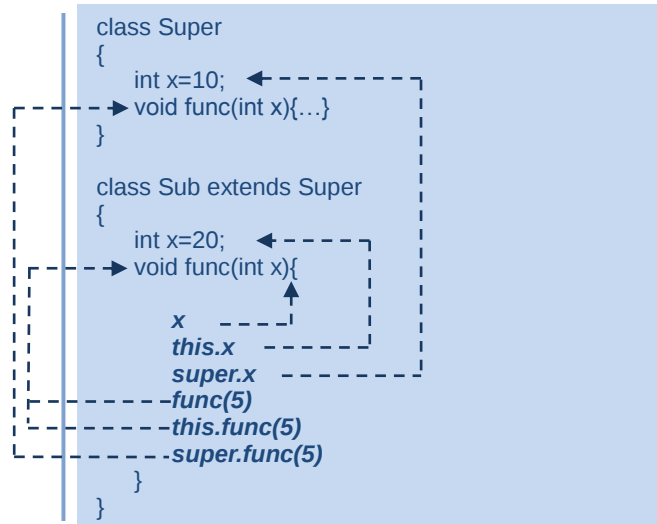
但し、super.修飾子を用いる場合オーバーライドは行われません
詳しくFAQを参照して下さい

this. 修飾子 自身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.`修飾子と `this.`修飾子を用いて区別することができます



継承・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 → 拡張させない
- 変数 → 値を変更させない（定数）
- メソッド → オーバーライドさせない