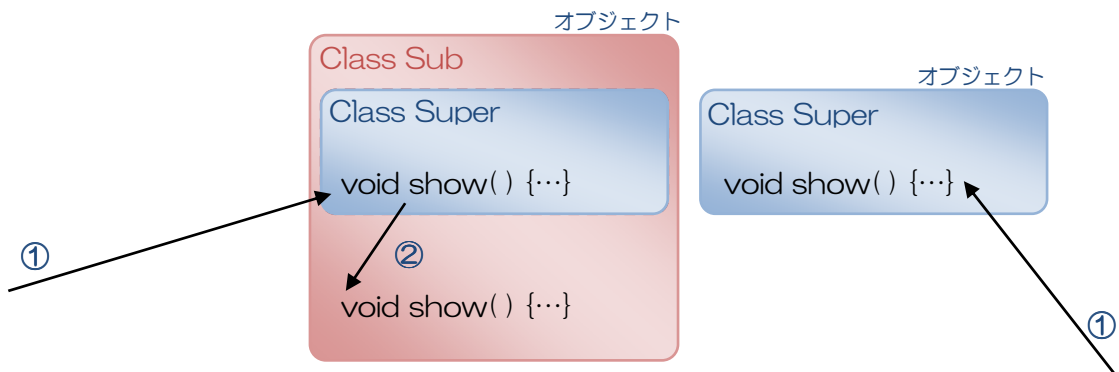


メソッドのオーバーライド

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、戻り値は次にしたがいいます (※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

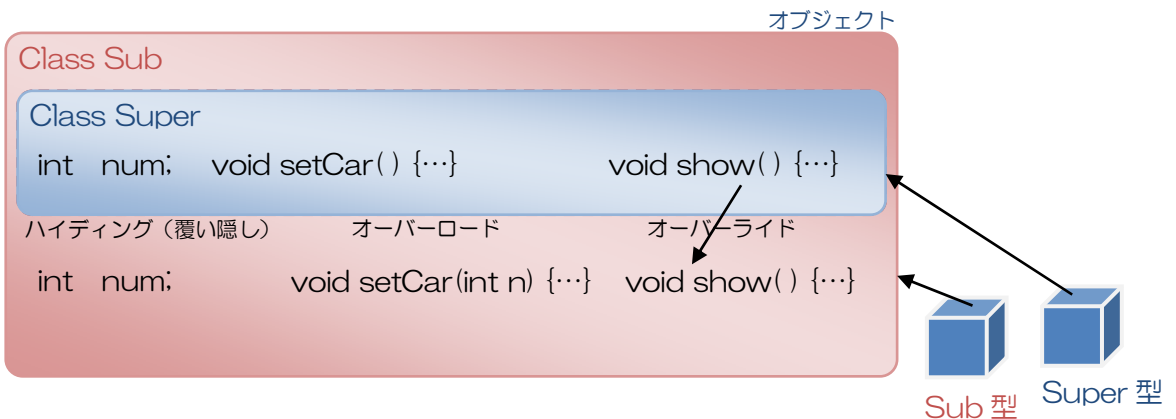
動作 オブジェクトの型に依存してメソッドが選択され実行されます
➔ ポリモーフィズム (多様性) を実現



同じ名前のメンバーの宣言

スーパークラスと

- ・ 同じ名前の変数 (型に関係なく) をサブクラスで宣言 ➔ ハイディング (覆い隠す)
※オーバーライドされない
- ・ 同じ名前、同じ引数の数/型 (戻り値は※1) のメソッドをサブクラスで宣言 ➔ オーバーライド
- ・ 同じ名前、異なる引数の数/型 (戻り値に関係なく) のメソッドをサブクラスで宣言 ➔ オーバーロード



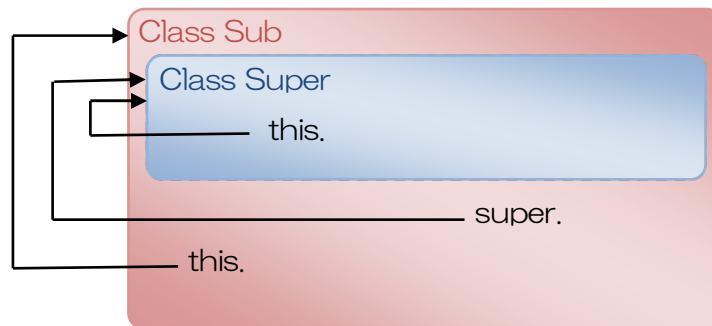
スーパークラスのメンバーへのアクセス

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

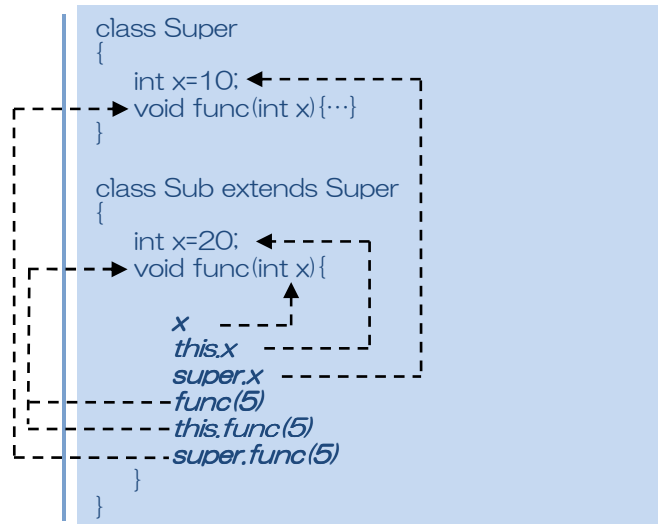
クラス型変数とメンバーアクセス 使用するクラス型変数の型のオブジェクトへの参照としてメンバーにアクセスします

super. 修飾子 スーパークラスのオブジェクトへの参照としての意味をもち、現在の自身のオブジェクトへの参照となります
ただし、super.修飾子を用いる場合オーバーライドは無視されます

this. 修飾子 自クラスのオブジェクトへの参照としての意味をもち、現在の自身のオブジェクトへの参照となります



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 → 継承させない
- ・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