

表 3-2-1 多重度の例

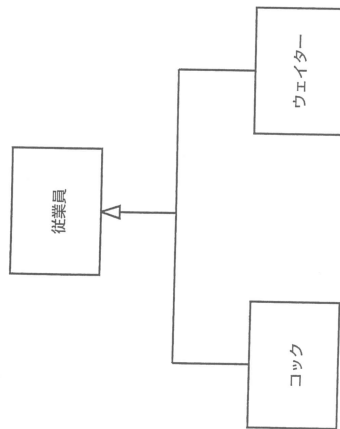
例	意味	該当する数
5	固定値	5
1..3	範囲を示す値	1, 2, 3
*	0以上の繰り返し	0, 1, 2, 3, 4, 5, ...
7..*	下限以上の繰り返し	7, 8, 9, 10, 11, 12, ...

汎化

汎化とは、クラス間の抽象-具象の関係を表します。あるクラスを抽象化して新たにクラスを定義した場合、元のクラスと新たなクラスとの関係は汎化となります。レストランの例で考えると、「コック」も「ウェイトー」も「ウェイトー」もレストランの「従業員」です。

このため、「コック」、「ウェイトー」を抽象化して「従業員」クラスを定義することができます。この場合、「コック」、「ウェイトー」と「従業員」の関係は「汎化」となります。図示すると以下のようになります。

図 3-2-9 従業員クラス (汎化)



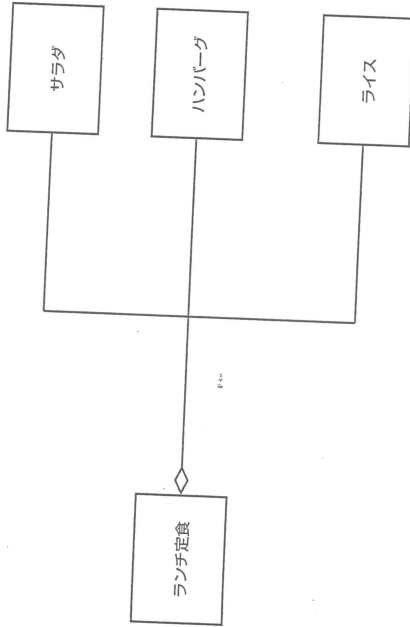
汎化関係にあるクラスのうち、新たに定義されたより抽象的なクラス（「従業員」クラス）を、「スーパークラス」または「親クラス」と呼び、もともとなる具象のクラス（「コック」クラス、「ウェイトー」クラス）を「サブクラス」または「子クラス」と呼びます。

汎化関係にあるクラスの間では、スーパークラスで定義された関連や操作、属性は、すべてサブクラスに引き継がれます。

集約

クラス間の関係が全体一部分になる場合、集約を用いて表します。レストランの例で考えると、「ランチ定食」に「サラダ」、「ハンバーグ」、「ライス」が含まれているとすると、これらの関係は「集約」となります。表記方法は以下のようになります。

図 3-2-10 ランチ定食クラス (集約)



コンポジション

集約の中でも特に強い関係をコンポジションと呼びます。集約とコンポジションはどちらもクラスの間で全体一部分の関係があることを表しますが、コンポジションの場合は、全体クラスと部分クラスの生存期間が同じになります。

つまり、全体クラスのインスタンスが消滅すると部分クラスのインスタンスも消滅してしまいます。例えば「ノートパソコン」は「画面」と「キーボード」と「バッテリー」から構成されます。ここで「ノートパソコン」のインスタンスが消滅すると、「画面」、「キーボード」、「バッテリー」のインスタンスも消滅してしまうため、これらの関係は「コンポジション」となります。表記方法は以下のようになります。

図 3-2-11 コンポジション

