

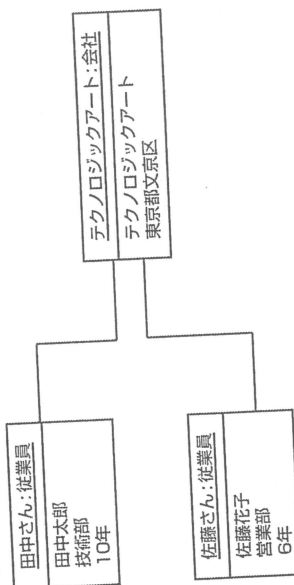
Part 2 オブジェクト指向の基礎

ここでは UML の表記法の詳細に関しては触れませんが、「田中さん」、「佐藤さん」は氏名や所属部署などの属性や、労働するという操作が共通であり、それらを抽象化して「従業員クラス」が定義されています。「株式会社テクノロジックアート」からは「会社」というクラスが定義されています。

「田中さん」と「佐藤さん」というオブジェクトを抽象化する場合、「人間クラス」を定義することもできます。また「会社クラス」をモデル化する場合、資本金という属性を持たせることもできます。このようにクラスは、対象領域の見方によってその定義が変わります。言い換えると、事象を抽象化する場合には、対象領域で必要とされている特性に注目するとともに、現在注目している問題領域や機能に対して適切な大きさでクラスを作成する必要があります。

このクラスの大きさをクラスの粒度と呼びます。  
システムが動作する時には、このクラスの枠組みから、田中さん・佐藤さん・テクノロジックアートのオブジェクトが生成され、それぞれのオブジェクトが協調しながら動くことになり  
ます。このように、クラスからオブジェクトを生成することを「インスタンス化する」とい  
います。

図 2-1-7 従業員-会社オブジェクト図



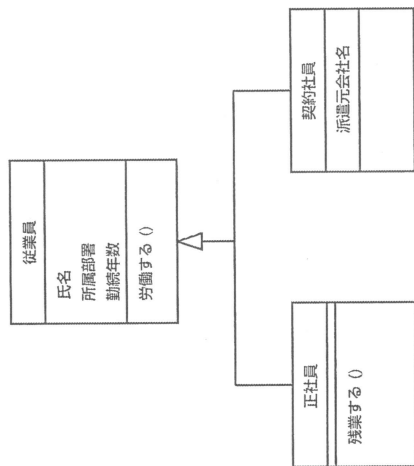
継承・汎化

前節では、オブジェクトを抽象化したクラスについて説明しました。前節の例では、田中さん、佐藤さんはどちらも従業員クラスとしてモデル化しましたが、実は田中さんは正社員、佐藤さんは契約社員だったとします。そして、正社員は残業することができませんが、契約社員は残業できないという決まりがあります。このような場合、従業員としての共通概念を残しつつ、差分を派生させたクラスを定義することができます。

継承 既存のクラス（属性や振る舞い）を利用し、さらに概念を追加して新しいクラスを定義することをいいます。

汎化 クラスから共通の概念をまとめあげて、より一般的なクラスを定義することをいいます。より大きなくりとなるので、クラス群の理解をしやすいくなります。

図 2-1-8 従業員クラス（継承クラス図）



ここでも UML の詳しい説明に関しては触れませんが、上の図は従業員クラス（従業員クラス）の概念をすべて受け継いで、正社員クラスと契約社員クラスを定義していることを示しています。このように、より一般的な概念のクラスから概念を引き継ぐことを「継承」と呼びます。正社員クラス、契約社員クラスは、ともに従業員クラスを継承しているため、氏名・所属部署・勤続年数という属性や、労働するという振る舞いも持っていることになり、さらに正社員クラスは、残業するという振る舞いも持っていることを表しています。  
逆に正社員クラスや契約社員クラスからみると、従業員クラスは共通の概念をまとめ上げています。このように共通の概念をまとめあげて、より一般的なクラスを定義することを「汎化」と呼びます。

継承や汎化の概念は、オブジェクト指向のもっとも基本的な考え方の 1 つです。これらの概念によって、物事をまとめて捉えたり、拡張する場合の手間を減らしたりすることができます。継承や汎化を使ってクラスを洗練していくアプローチは、人間の思考回路が事象や物事を分類、体系化して整理するプロセスに非常によく似ています。

可視性とインターフェース

オブジェクト指向における重要な考え方の 1 つとしてカプセル化を説明しました。カプセル化とは、詳しい内部情報を局所化するとともに、外部から隠蔽することでした。そして外部に対しては、必要な操作を公開することで、「私（オブジェクト）は\*\*\*\*しますよ」と明示しました。この外部に対して公開したクラスの仕様がインターフェースとされています。