

Lesson 1 ▶ オブジェクト指向の考え方

Point

▶ オブジェクトの定義

- ① オブジェクトが存在する前に認識する「もの」がまず存在します。これが「主体」です。
- ② 「主体」がある目的を持って、ある視点から認識した対象が「オブジェクト」です。たとえば、パソコンという「主体」は、使う人にとっては「容量」や「速さ」などの属性が重要ですが、販売する人にとっては「仕入れ」や「売値」が重要となります。そこで「パソコン」という1つの主体から異なる目的や視点によって、「別々のオブジェクト」が作成されます。

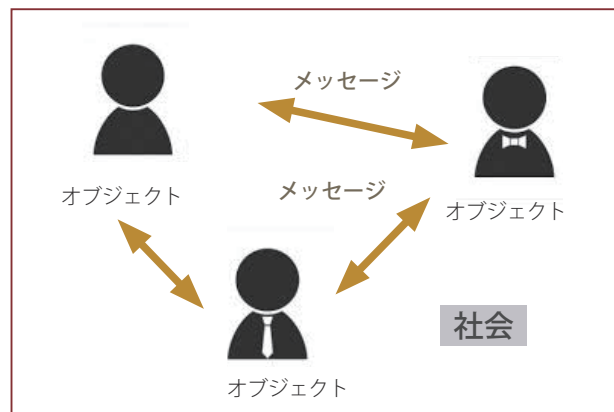
オブジェクト指向プログラミングにおけるオブジェクトは「データと処理の集まり」です。

▶ オブジェクトとは

オブジェクト指向とは、「物」=「オブジェクト」を中心に、「分析や設計」を行うことです。JavaやC++はオブジェクト指向の言語です。

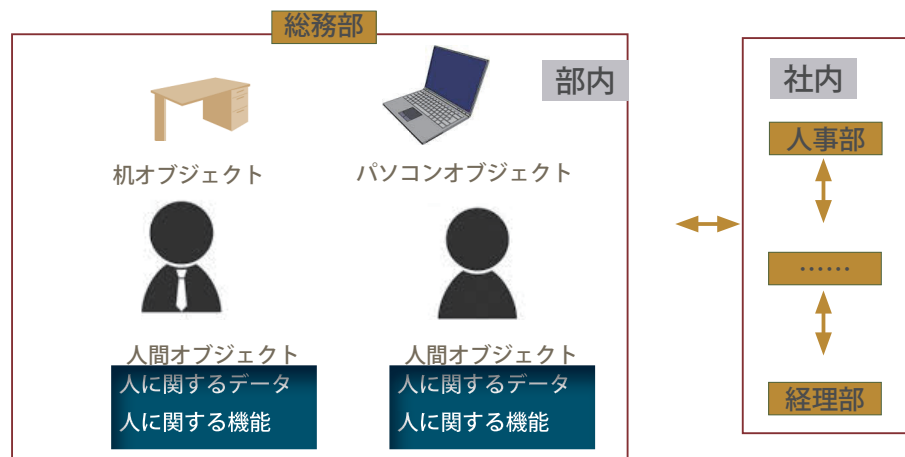
これらの言語を活用するためには、オブジェクト指向の基本概念をきちんと理解しておく必要があります。

人間社会をオブジェクト指向の考え方で眺めると、1人ひとりの人間がオブジェクトです。そして人間の集まりで社会というシステムを構築されています。人間はそれぞれ自分の役割を持っています。1人でできない仕事は会話を通してコミュニケーションを行い、ほかの人々と協力して成し遂げようとしています。



1つひとつのオブジェクトは独立して存在し、それぞれ固有の役割・責務を持ちます。それらは孤立した存在ではなく、互いにメッセージを通じて協調動作を行いながら、全体として1つの仕事を行います。これがオブジェクト指向の基本的な考えです。オブジェクト指向の考え方では、「人に関するデータ」と「人に関する機能」のみを人間オブジェクトの中に入れます。机やパソコンなど人間に関係のないものはそれぞれのオブジェクトを別に作ります。

それぞれのオブジェクトは自分の仕事のみを行い、自分のデータのみを参照します。各部では、それぞれ自分がやるべき作業を独立して行います。しかし、その部署ではできない作業を行う場合は、別の部署に依頼します。部署もオブジェクトとして捉えます。組織をオブジェクトと捉えると下記の図式となります。



Point

▷オブジェクトにできないもの

◎自然現象(例外を除く)

雲は、1つ1つを明確に識別することができません。雨、風も同様です。ただし自然現象でも台風はオブジェクトと考えることができます。台風は「台風第X号」というオブジェクトになります。

◎色

オブジェクトと考えるより、あるオブジェクトの属性と考えるのが一般的です。

色白の人が日焼けした場合、白が黒に変わりましたが、「白いAさん」と「黒いAさん」を別々のオブジェクトまたは、白と黒というオブジェクトがあって、その状態がAさんだったり、Bさんだったり、と考えるのも無理があります。

これは「Aさん」というオブジェクトの、「色」という属性値が変化したり、つまり状態が変化したりと考えるのが一般的です。

◎数量

色と同様、オブジェクトと考えるより、あるオブジェクトの属性と考えるのが一般的です。

◎人の感情

感情は、人とほかのオブジェクト、あるいは人と人との関係と考えるのが一般的です。

Point

▷プログラミングのアプローチ手法

◎手続き指向

どのように操作を実現するかを考える手法で一般的です。

C言語

◎データ指向

手続き指向とは反対に、必要なデータを中心に考えます。

リレーショナル・データベース

◎オブジェクト指向

手続き指向とデータ指向の両方から考えます。

Java、C++

▶ オブジェクトの特性

オブジェクトは、「物理的なオブジェクト」と「概念的なオブジェクト」に分類します。しかし、種類に関係なく下記の4つの特性を満たすものがオブジェクトの定義です。

*属性

オブジェクトは、固有の姿・形・性質などを持っています。このようなオブジェクトの状態は、属性の値で表現されます。属性は一般的に名詞で表現します。

物理的なオブジェクトの属性としては、「大きさ」「重さ」「色」などがあります。人をオブジェクトと考えると、「名前」「生年月日」「身長」「体重」「血液型」などはその属性です。

概念的なオブジェクトの属性には、「日時」「場所」「議題」などがあります。

*操作

オブジェクトは固有の振る舞いを持ちます。これを操作と呼び、動詞で表現します。物理的なオブジェクトの振る舞いには、「話す」「歩く」「食べる」「寝る」「働く」があります。

概念的なオブジェクトの場合は、「会議」をオブジェクトとすると、「開催場所を設定する」「開催日時を設定する」「議題を設定する」といった操作(振る舞い)を考えることができます。

*関係

オブジェクトは、単独で存在することはありません。必ずほかのオブジェクトと関係を持っています。

*アイデンティティ

車がナンバープレートで識別できるのと同じように、各オブジェクトは、ほかのオブジェクトから自らを識別できるアイデンティティを持ちます。

▶ プログラムにおけるオブジェクト指向

オブジェクトは、それらが互いに関係しあい、ある機能を実現します。プログラムにおけるオブジェクト指向も同様に、いくつかに分割した部品それぞれがデータと機能を持ちます。この部品がオブジェクトです。

Javaは、オブジェクト指向の言語です。クラスを複数定義し、クラスから生成したさまざまなオブジェクトを組み合わせるプログラミングをします。

プログラムを作成していく際には、どのようなオブジェクトにどのような機能を持たせるかを考えることからスタートします。

クラスの重要な役割はそのようなオブジェクトを設計し、定義することです。

この考え方で行う手法をオブジェクト指向プログラミングといいます。

