

連載 オブジェクト指向と哲学

第 21 回 オブジェクト指向の設計主義と自由主義

河合 昭男

前回はクリストファ・アレグザンダー（以下 C.A.）とハイエク（Friedrich August von Hayek）に共通する設計主義批判の視点から、インターネットの「成長する全体」を考えました。

インターネットは誰でも自由に参加できますが、誰も全体をコントロールできません。その成長と自生的秩序を阻害するのがハッカーです。自分たちもインターネット社会というコスモスの住人であるにも関わらず、自らのコスモスを破壊するがん細胞です。一部のハッカー集団はコスモスの中に自分のタクシスを作りだし、テシスも自分たちで決めてしまい、全世界にそのテシスを押しつけるという一種の全体主義者です。コスモスとノモスを認めず、それをタクシスとテシスにおとしめようとするのがハッカー集団です。異なるハッカー集団が異なるタクシスとテシスを形成すれば戦国時代です。これは正義が勝つか悪が勝つかではなく、コスモスとノモスが存続できるか、その存在を肯定するか否定するかということです。

同じレベルのルールの違いで争わず、より上位のルールを発見し、それに従わなければ共存共栄はありません。

オブジェクト指向

今回はオブジェクト指向という考え方を、設計主義と自由主義という視点で考えてみたいと思います。

オブジェクト指向は本来プログラミング言語から誕生した概念です。カプセル化、継承、多態性がオブジェクト指向プログラミング言語の特徴です。

UML が誕生し、オブジェクト指向は上流系モデリング技術に発展してきました。プログラミング言語の特徴よりも、より包括的な概念に意味が変化してきました。

オブジェクト指向の考え方はひとこと言えば「自律分散協調モデル」です。その特徴は次の3つです。

世界をオブジェクトの集まりと捉える

オブジェクトは固有の責務を持つ

オブジェクト同志が協力し合ってひとつの仕事を行う

設計主義的オブジェクト指向

システム開発で活用されるオブジェクト指向は、当然ながら設計主義です。開発者がシステムに必要なクラスを抽出し、そこに責務を割り振り、コラボレーションモデルを作

成します。これが問題だというつもりはありません。

人間社会をオブジェクト指向で捉えることができます。会社で考えるなら ビジネスに必要な人員を集め、業務を社員に役割分担し、業務を組織構造のコラボレーションとして設計します。

会社というタクシスはそれぞれの都合でテシスを作り、秩序と成長をコントロールします。設計主義批判の対象ではありません。ビジネス目標を達成するためには、その設計が必要です。

自由主義的オブジェクト指向

責務を自分自身で決めるオブジェクト指向を、ここでは自由主義的オブジェクト指向と名付けます。

人間社会で考えるなら、会社に入ると社員の責務やコラボレーションはトップダウンで決まってゆきます。就活中なら、自分はどのような仕事をしたいのか選択の自由があります。どのような人と仕事をしたいのかも選択できます。

いっそベンチャーを志すなら、起業すれば自分の会社のオブジェクトモデルは自分で設計することができます。自身の責務やコラボレーションは自由に選択できます。

つまりこれからの自身の責務はまだ決まっていないし、自身で決められるという意味で、トップダウンに役割分担が決まる設計主義ではありません。

クラスとインスタンス

自由主義的オブジェクト指向には技術的な問題があります。

オブジェクト指向では責務はクラスに定義されます。同じクラスから生成されるオブジェクト(インスタンス)は同じ責務を持ちます。つまりオブジェクトは責務を自由に選べない。クラスに定義されている範囲内で状況により変化させることしかできません。

オブジェクトの存在以前にクラスが存在していなければなりません。そのクラスの設計は誰がするのかということです。

職業選択の自由

人間社会で考えるなら、例えば現在の日本の法律では人は職業を自由に選べますが、江戸時代には職業は選択できませんでした。親の職業を子が引き継ぐ、それが人が持って生まれた運命でした。

UML で表すなら、人のインスタンスは土農工商のどれかのインスタンスであり、「働く()」という責務はその4つのサブクラスで定義されたものになります。一旦オブジェクトが生成されたら他のサブクラスに移れません。(図1)

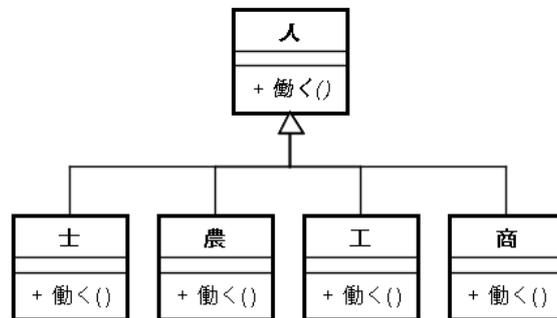


図 1 職業選択ができないモデル

職業選択ができるケースは、UML ではステートパターンで表現することができます。(図 2) 人のオブジェクト自体に「働く()」という責務は定義されていません。職業という抽象クラスの下に様々な職業を並べます。「働く()」という責務の具体的内容は、これら具体的職業別のクラスに定義されています。人のインスタンスは、これら具体的クラスのインスタンスから選んでリンクを張ることができます。

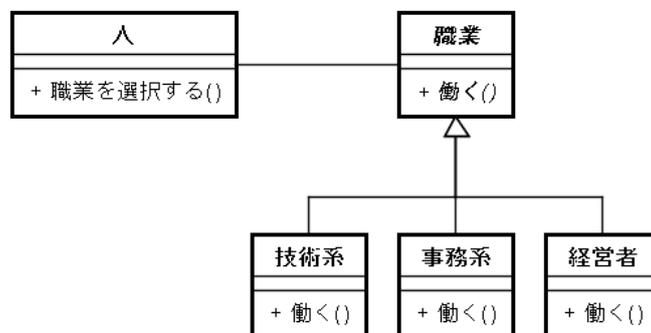


図 2 職業選択ができるモデル

図 1 のモデルでは、人のオブジェクトが生成されたときに職業が決まり、しかも変更もできません。図 2 のモデルでは、人のオブジェクトが生成された後で職業が選択でき、変更もできます。ならば図 2 のモデルを自由主義的オブジェクト指向といえるかということ、まだ問題があります。

「職業を選択する()」という責務の具体的内容、どのようなロジックで職業を選択するのかということを事前にクラスに定義しておくことが必要。

職業のサブクラスを事前に準備しておくことが必要。人オブジェクトが生成されてから新たな職業クラスを追加できない。

つまり、もも事前に設計が必要という意味で設計主義からは脱しきれていません。自由主義的オブジェクト指向の条件は、設計時でなく実行時に と が可能なモデルです。

世界をオブジェクトの集まりとして捉えるオブジェクト指向、現実世界をオブジェクト指向でモデリングするには何か重要なものが欠けています。自由がなければ創造や成長はありません。設計主義から逃れられません。

以 上