

コーディング | Coding

世界で一番小さなデザインの方法

デザイン思考

テレビ、新聞、雑誌、ブログなど、「デザイン思考」という言葉がここ数年巷を賑わせているが、システム開発やプログラミングの経験者にとっては、それは、以前から身近なものであった。

ウォーターフォール、プロトタイプ、スパイラルなどいくつかのパターンがあるが、フィールドを調査し(現状分析)、問題を明確にし(要求定義)、様々なプロトタイプを試しつつ、コンセプトを明確にし(設計)、具体的なモデルをつくる(プログラミング)。その後システムレベル、ユーザーレベルのテストと調整をくりかえして運用に至る。また、その過程で得られた知見は、新たな開発手順やプログラミング手法の創出につながるのである。デザイン思考のサイクリックなプロセスがここにある。もちろん、「デザイン思考」はデザイナーにとっても以前から日常的なものであったわけだが、時代はコンピュータの急速な発展・普及の最中にあり、現代に生きるデザイナーの思考は、少なからずその影響を受けていると考えるのが自然であろう。

スパゲッティーのようにつながった抽象度の低いプログラムから、構造化プログラミング、そしてオブジェクト指向とデザインパターンの共有へ。この数十年の歴史の中で、プログラミングの手法は進化してきた。

プログラマーが日常的に行っているコーディングの中には、そうしたデザイン思考にもとづく経験があり、その知見が私たちの暮らしに大きなイノベーションをもたらしている。以下、筆者がプログラミングとデザイン活動の中で発見したキーワードを通してデザインの方法を考察したい。

小さいことをうまくやる

プログラムは一般に関数と呼ばれる機能体の集まりとして記述される。数学で学ぶ関数と同じで、例えば f という名前の関数を使って $y=f(x)$ と書けば「入力 x に処理を施した結果を y に戻す(格納する)」という意味になる。例えば、C 言語の `isdigit` 関数は、キーボードから入力された文字が 10 進数の数字 ('0' - '9') であるか否かについて真偽を返す。`isdigit(9)` の戻りは `true`。非常に小さなプログラムであるが、それがどんな状況においても問題なく動くという信頼を得られれば、同様の小さな関数を集めることで、例えば「入力された情報が生年月日のデータとして正しいか否かを判定する」という、少し大きなプログラムに発展する。

こうして出来上がった関数をいつでもつかえるように整理・保管(ライブラリ化)しておけば、次以降の仕事は加速度的に楽になる。実際に私自身、10 年以上前に書いたプログラムを現在でも Web システム上で活用している。自分の作ったものを一生使い続けられるというのは大きな喜びだ。「UNIX の哲学」の筆頭は *Small is Beautiful*。「小さいことをうまくやる」である。プログラマーは小さなプログラムを丁寧につくることが、自分たちの未来を快適にすることを知っている。

生物の細胞は近隣の細胞との関係を調整しつつ、「自分ゴト」をうまくやることだけに専念している。それでも全体は動くのだ。中央集権で考えない。小規模な機能体を自律分散的に協調させるという発想がシステムデザインの理想である。

外部の影響を受けない、外部を汚染しない

小さなプログラムは、それ自体で完結し、外部環境とは関わらないといことが重要である。プログラムは様々な場面で部品として活用されるので、それが他の影響を受けて振る舞いを変えてしまうようでは困るし、逆に部品内部の出来事が直接外部の存在に影響するようではいけない。

プログラムの外部にあって、複数のプログラムから参照・改変される存在のことを「外部変数」というが、プログラミングの世界では、この外部変数を極力使わないようにシステムを作ることが理想とされていて、入力と出力以外には一切関わらないプログラムだけがライブラリ化されて「一生モノ」となる。

一方、私たちをとりまく多くのモノはこの問題から逃れることができていない。例えば電子機器は「入力動作」以外に、気温の変化や電磁波の影響を受けて誤動作することがあるし、また、自動車は「移動する」という出力以外に、排気ガスや排気音が周囲の環境に影響を与えてしまう。定められた入力以外の影響は受けず、また定められた出力以外には外部に影響を与えない。これがものづくりの理想だ。

実体を動かさない、情報を動かす

ある下位プログラムが、上位のプログラムに結果を渡す場合、そこには大きく 2 つの方法がある。ひとつは結果としてのデータそのものを渡す「実体渡し」といわれるもので、もうひとつは、結果のデータが存在しているメモリー上の番地を渡す「アドレス渡し」と

いわれるものである。例えて言えば、子会社で完成した部品をトラックで直接親会社に持ち込むのが「実体渡し」で、「〇番の倉庫に置いてある」などとその所在を親会社に報告し、物を動かす手間を省くのが「アドレス渡し」だ。

OS の記述言語として知られる C 言語によるプログラムでは、この「アドレス渡し」がシステムの効率的な動作に大きく貢献している。

「実体」に重量がある現実の世界では特に、その輸送に大きなエネルギーを必要とする。モノの移動を最小限にとどめ、情報を動かす。宅配便は個人宅へではなく、最寄りのコンビニや共有ロッカーまでとして、あとは届けたことを知らせる。送り手と受け手が場所を共有して情報だけ送るというのが物流の理想である。

実体はそのまま、見せ方を変える

例えば C 言語には `#include`、また例えば HTML には `<link>` というものがあり、既存のライブラリーやスタイルシート(以下 CSS)をヘッダーで読み込んで利用することができる。本体(BODY)と前置き(HEAD)が区分されたことで、例えば Web サイトの構築では「内容と形式の分離」という、物質媒体では不可能なデザインの方法が採用される。文字の種類、大きさ、色、背景色、そして各要素の配置方法など、情報要素のビジュアルを定義した CSS は、HEAD で読むことを前提に独立したファイルとして記述されるのが標準的で、これによって、記事(HTML)に手を加えることなく、CSS の入れ替えだけでサイトをリニューアルすることが可能になる。実体はそのまま、見せ方を変える。プログラム部品だけでなく、「記事」というレベルの情報資源も、永遠のものとなるのである。

名前を付ける

CSS は、記事を囲むタグの名称やタグに付与された ID や CLASS を使って、対象のビジュアルを操作する。例えば、`h1{font-size: 24pt;}` と書けば、`h1` タグで囲まれたすべての要素の文字サイズを 24pt に、また例えば、`#logo{width:40px;}` と書けば、`logo` と名付けられた要素の幅を 40px に制御できる。対象に何らかの名前が付されているということは、それだけで、外部からのコントロールを可能にするのである。

夢枕獏の小説「陰陽師」に登場する安倍晴明は、「この世で一番短い呪とは、名だ」といった。「呪とはな、ようするに、ものを縛ることよ」なのである。

言語学者 F. ソシュールは「言葉は『ものの名前』ではない」と言った。対象は所与のものとしてあるのではなく、「言葉」の存在が環境や事物を区分して、世界を立ち上がらせるのだ。

プログラムのデザインパターンにも名前がついている。名前があることによって、問題のタイプや解決策のイメージが明瞭になり、メンバー間での情報共有がより効率よくなるからである。

Design の語源はラテン語の *designare* で、印を付ける、区分して描くこと。また、*Designate* は、示す、指示する、任命する、名付ける、呼ぶといった意味を持つ。プログラミングの世界における「名付ける」という行為には、情報デザインの原型がある。

価値は所与のものではない

我々は、言語を介して世界を認識している。つまり、我々の住む世界は、物理的な世界ではなく、名付けられることによって再構成された擬似現実である。

レヴィ＝ストロースによれば、人間の作り出すすべての社会システムは、同一の状態にとどまらないように構造化されている。擬似現実(共同幻想)は根拠を持たぬがゆえに脆い。秩序は絶えず組み換えられることによってしか平衡状態を保つことができないのだ。それは生物というシステムが動的平衡を保っているのと同じである。

使う言葉を変えれば世界も異なって見える。物事の意味や価値は所与のものではなく、絶えず更新されつづけているのである。

何かが差異化され、区分けられ、名付けられると同時にそれが生まれる。だとすれば、価値はいくらでも生成できる。創造するということは、とても簡単に、そして無限に可能なのだ。

資源を消費してモノをつくることだけがデザイナーの仕事ではない。名付けることによって対象を顕在化させること、既存の要素を異なる文脈の中に転用すること、要素間の関係を組み換えること。そうして新たな価値を生み出すことも、デザイナーの役割のひとつではないだろうか。

コーディングという言葉には「符号化」という意味もある。この世で最も小さくて面白いデザインの方法とは「名付ける」ということである。

「風」はその言葉の存在によって、我々の住む世界に吹いている。

参考文献

Mike Gancarz (芳尾桂 監訳), 2001, UNIX という考え方, オーム社
E.F.Schumacher, 1973, Small is Beautiful, Blond & Briggs
福岡伸一, 2007, 生物と無生物のあいだ, 講談社現代新書
夢枕獏, 1999, 陰陽師, 文藝春秋
丸山圭三, 2008, 言葉とは何か, ちくま学芸文庫
Claude Lévi-Strauss (大橋保夫訳), 1976, 野生の思考, みすず書房