

強いプログラムを作るテクニックを学ぶ

第4回 モジュール品質を知る

酒井 郁子
館 伸幸

前回(2011年3月号, pp.146-151)では、要求分析の重要性を解説した。設計の最初のインプットである「要求仕様書」からソフトウェア要求を分析するには、それなりの知識と経験が求められる。

今回は、できあがった要求仕様書からどのようなプログラムを作成するべきかを解説する。設計基本要素であるモジュールの品質について物造先輩が熱く語る。
(編集部)

(先輩：物造)

え〜中途半端な仕様書を野比さんに渡したまま放任した結果、馬具取部長から嚴重注意されました…。手抜きしようとか意地悪しようかと思ったわけでは…。野比さんが「任せて!」みたいな得意気な顔をしているから当然知っているかと…ブツブツ

(新人：野比)

はいはい、言い訳は分かりました。もう、済んだことは忘れますから。

今回は、設計品質について先輩が教えてくれるのですよね。部長からは、設計する前にどういうプログラムにすればよいのかをしっかりと習いなさいと言われました。よい設計のポイントを知らなければ、自分が設計したものがOKなのかも分からないじゃないですか!

(先輩：物造)

そうだったね。その通りだよ、野比さん。いいところに

気が付いたね。

設計品質についての知見を持たずして、よい設計、ひいてはよいプログラムが作れることはないのだよ。では、僕が過去のプロジェクトの例を使って、モジュール分割の良し悪しを教えてあげよう。

(…突然、立ち直り、過去のドキュメント類やコードを見せて得意気に語り出す)

1. 設計にはいろいろな視点がある

● 論理面だけで構造を組み立てる —— 静的設計

本連載の第1回目で、設計によって、①分かりやすい境界で分離し、②階層構造で組み立て、③モジュールの独立性を高めるようにする、ということを話しました。これらは、プログラムの「動き」を考慮に入れず論理面のみで構造を組み立てるためのもので、「静的設計」と呼びます。

● プログラムの振る舞いを考慮する —— 動的設計

ソフトウェアを設計する際には、静的設計だけでなく時間やリソースなどを考慮に入れたプログラムの振る舞いも設計する必要があります。これを静的設計に対して「動的設計」と呼びます。

● 静的設計 → 動的設計の順序で設計

静的設計では処理の順序や時間という概念を持ち込むことなく、純粹に論理的に要件を分解し、階層構造を組み立てます。プログラム開発経験のある人ほど、設計するときに処理順序やリアルタイム性などの時間条件に絡んだ実現方法をどうするかが気になって頭から離れられないものです。しかし、何の処理をどこに置くかを見定めずに動きだけを考えると、似かよった処理を何箇所も作ってしまったり、冗長な処理を作ってしまったという事になります。



図1 得意気に語りまくる物造先輩の様子