

強いプログラムを作るテクニックを学ぶ

第12回
(最終回)

モジュール内部の詳細設計～設計成果を振り返る～

酒井 郁子
館 伸幸

連載の総仕上げとして、これまで行ってきた設計の振り返りを行います。また、機能分割を行う上で定番の分割技法があるのでそれを最後に紹介します。

(編集部)

(馬具取部長)

さて、そろそろ野比さんのプログラムはでき上がってきたかい？

(野比さん)

はい、しっかり分析・設計した結果、コードのどこに何が実装されているか、すぐに説明できるプログラムができましたよ。

(物造先輩)

それはもう、僕が手取り、足取り指導した賜物ですからね。要求仕様から、一貫性の取れたプログラムの構造で作らせましたよ。

(馬具取部長)

ほ～、それは楽しみだね。ぜひ、野比さんにどんなプログラムか説明してもらいたいね。

(野比さん)

はい！！

馬具取部長にでき上がったプログラムの説明をする前に、野比さんが最後に行った設計工程があります。そこでまず、前回までを振り返り、残りの設計作業について説明しましょう。

前回、DFD (Data Flow Diagram) による分析結果から作成したモジュール構造図をたたき台として、実装条件を盛り込んだ構造を設計しました。これまでのステップで、次のような特徴を持つ設計ができ上がっています。

- (1) DFD分析により、主要な機能の特徴で分割した設計構造の指針
- (2) 階層化DFDをモジュール構造に変換したことにより、詳細を下位層に隠ぺいしたモジュール階層構造
- (3) DFDから変換したモジュール構造上に、動的な実装

条件の補正を行ったことで、(1)と(2)のよさを受け継いだ動的な処理分割

1. モジュール内部の詳細設計

思い出してみましょう、構造化分析・設計で目標とするプログラムは、

(1) 分かりやすい境界で、処理を分割する

複雑なシステムを理解しやすい塊に分割し、整理する。

(2) システムを階層構造で組み立てる

上位階層には下位層の処理を統括した大きな粒度のモジュールが、下位層には上位層の機能を分割した単機能モジュールとすることで、詳細が隠ぺいされた階層構造を作る。

(3) 各モジュールの独立性を高める

独立性を高めることで、モジュール内の処理を変更しても、ほかのモジュールに影響しない、ブラック・ボックスとして再利用できるようなモジュールを作る。

ということでした。ここまでの分析・設計により、この目標に沿った設計構造の幹はでき上がっています。分析・設計をプログラムとして実現するためには、この設計構造の枝葉の部分である各モジュールについて、その詳細設計を行うこととなります。

各モジュールの役割(何をする = What)は既に決まっています。そのモジュールを実装するためには、次の二つについて詳細設計の段階で作り込みます。

(1) モジュールのインターフェースを具体化する

モジュールの概略インターフェース(何が入力データとなり、出力データは何か)は、これまでの分析設計で決まっています。それをどうやって受け渡すかを、詳細設計では