

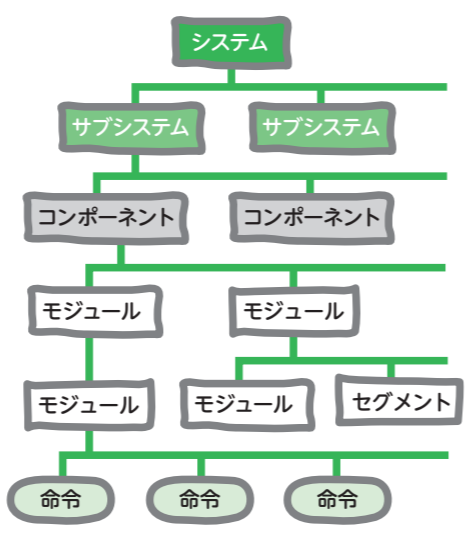
出る! テーマ2 モジュール分割

システム設計の作業は、各工程ごとに順次細かく分割していくこと。最も小さな単位が1つのプログラムになるイメージです。分割した単位は、「モジュール」のほか、一般に次のような名称でも呼ばれます。また分割方法は、システムの特性によって、さまざまな技法を用います。



モジュールの概念

- コンポーネント**
システムを構成する機能単位。座席予約や入金処理といった、関連する一連の作業を1つにまとめた単位です。
- モジュール(ユニット)**
独立してコンパイルできる単位。1つの機能単位に分割されたプログラムを指します。1つのモジュールをさらに分割して階層化する場合もあります。モジュール分割のメリットは、処理効率の向上、部品化、再利用が可能、保守の効率化と信頼性向上、などです。
- セグメント**
関数やサブルーチンなどの単位を指します。



モジュール分割技法

- STS (Source/Transform/Sink) 分割**
データ処理の流れに沿って、プログラム構造を入力、処理、出力の3つに分割し、それに基づいてモジュールを作成します。
- トランザクション (TR) 分割**
データに対応する処理 (トランザクション) の種類に応じて、処理モジュールを分割し、作成します。

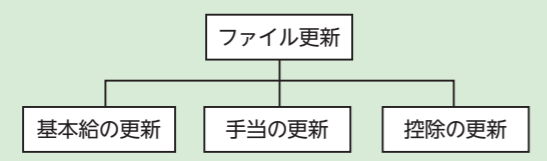


実際の問題で理解を深めよう



【判断問題】 3つのモジュールの処理内容に注目!

基本給の更新、手当の更新、控除の更新に関する伝票を個別に受け付け、給与計算用のファイルを更新するプログラムを、図のようにモジュール分割した。このモジュール分割の方法の名称はどれか。



- ア STS分割法
- イ ジャクソン法
- ウ トランザクション分割法
- エ ワーニ工法

→正解はウ。3つのモジュールは、関連のない独立した処理単位として分割されていることに注目しよう。

出る! テーマ3 モジュールの独立性

プログラムは開発した自分だけのものではありません。どんな人が関わってもいいように、内部構造もわかりやすくしておく必要があります。モジュールの独立性とは、モジュール内部の構造と外部とのつながりについての尺度です。直接的に処理内容に関わることはありませんが、独立性が保たれていると、メンテナンス性や拡張性に優れ、また再利用もしやすくなります。

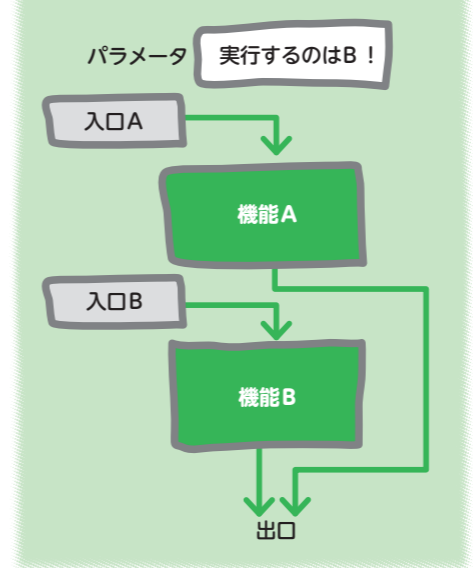


モジュールの強度とは

「強度」とは、分割された1つのモジュール内に、どんな要素が含まれているかによって決まる指標です。例えば、全く関連のない複数の機能が1つのモジュール内に含まれていれば、強度は「弱い」ことになり、1つの機能だけでモジュール内が成り立っていれば、強度は「強い」ものになります。下の表は、さまざまなケースによるモジュール強度の種類を取り上げたものですが、可能な限り、1モジュール1機能を実現する**機能的強度**を目標にしましょう。



論理的強度



強度	分類	説明
強	機能的強度	特定の機能を実行するために、モジュール内のすべての命令が関連している。
	情報的強度	特定の同一情報を扱う複数の機能が1つのモジュールにまとめられている形。
	連絡的強度	モジュール内でデータの受け渡し、または参照を行いつつ、複数の逐次的機能を実行するもの。
	手順的強度	複数の逐次的機能を実行する。例えば、流れ図の一部を無作為にモジュール化したような形。
	時間的強度	特定の時期に連続して複数の機能を実行するために同一モジュール化されたもの。機能間の関連性は弱い。
	論理的強度	関連したいくつかの機能を含み、機能を指定するパラメータにより、1つの機能を選択して実行するもの。
弱	暗合的強度	モジュールを大きさだけで分割し、たまたま重複した部分について共通モジュール化したもの。モジュール内の要素間に特別の関係がない。



モジュールの結合度とは

「結合度」は、分割されたモジュールどうしの結びつきを示す指標です。最もすっきりしているのは、互いに必要なデータのみをパラメータとして受け渡す**データ結合**です。例えば、 $A+B=C$ のような加算を行うモジュールを呼び出すのなら、A、B、Cの3つを変数をパラメータとして受け渡します。 $A+1$ 、 $B+2$ を入れて呼び出し、返ってきたCには3が入れているという形です。もし、呼び出すモジュールに、加減乗除の4つの機能が含まれていたなら、4番目のパラメータとして**計算の種類を指示する**必要があり、**制御結合**となります。さらに、1つのモジュール内に複数の機能が含まれることとなり、モジュール強度は「論理的強度」になります。