txt040213 モジュール設計

① モジュールの設計とは

ⓐ 良いプログラム設計の要点

㋐ **複雑さの減少**
㋑ **分割**
㋒ **独立性**
㋓ **階層構造化**

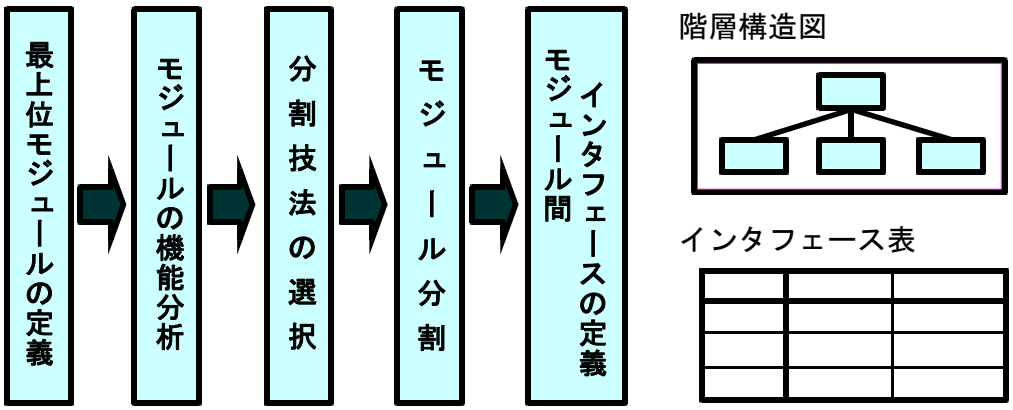
ⓑ プログラムの構造化設計

プログラム設計はプログラムをモジュールに分割し、階層構造化し、モジュール間のインタフェースを明確にし、階層構造図に表すことである。

プログラム構造化設計のポイントを次に示す。

㋐ **プログラムを適当な大きさのモジュールに分割する。**
㋑ **分割したモジュールを階層構造にする。**
㋒ **モジュール間のインタフェースを明確にする。**

ⓒ 構造化設計の手順



㋐ 最上位モジュールの定義

全体的な機能を定義する。カウンタ類の初期化、ファイルのオープン・クローズ、各機能の制御を行う。

㋑ モジュールの機能分析

上位モジュールで定義した機能を分析し、それを実行するために必要な下位の機能を明ら

かにする。機能の分割や統合を検討する。入力機能、入力データチェック機能、エラーデータの処理機能、ファイルの読込・書込機能、出力機能などを行う。

㋒　分割技法の選択

モジュールを分割するための最適な分割技法を選択する。分割技法しては、ＳＴＳ分割技法、ＴＲ分割技法、共通機能分割技法、ジャクソン法、ワーニエ法などが対象になる。

㋓　モジュール分割

選択した分割技法を用いてモジュール分割を行い、階層構造図を作成する。

㋔　モジュール間インタフェースの定義

上位モジュールと直接従属モジュールのインタフェースを明確にする。

㋕　分割すべきモジュールを探す。

更に分割できるモジュールがあれば分割を繰り返す。

②　モジュール分割技法

ⓐ　分割技法とは

データの流れに着目した分割技法とデータ構造に着目した分割技法がある。データの流れに着目した分割技法に、ＳＴＳ分割、トランザクション分割、共通機能分割などがあり、データ構造に着目した分割技法に、ジャクソン法、ワーニエ法などがある。

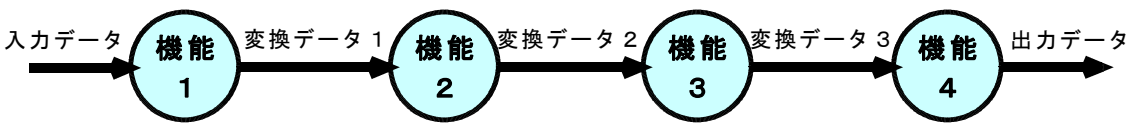
ⓑ　ＳＴＳ分割技法とは

プログラムのデータの流れに着目し、入力処理機能、変換処理機能、出力処理機能に分割していく方法である。

ⓒ　ＳＴＳ分割の手順

㋐　主要な機能を取り出し、問題構造を把握する。

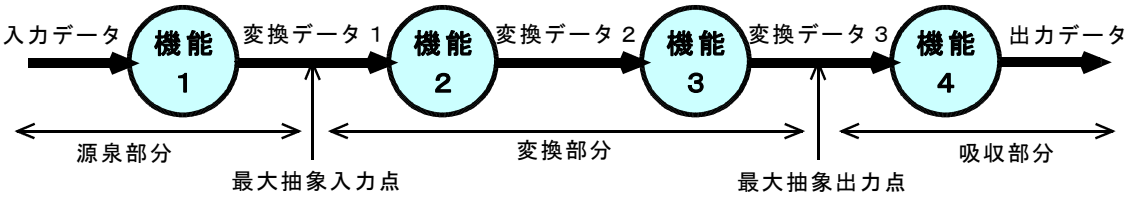
プログラム仕様書を分析し、主要なデータの流れに沿って処理機能を把握する。３～10個程度の機能を取り出し、問題の構造を明確にする。



㋑　主要なデータの流れを明確にし、機能と関連づける。

問題構造の中での、主要な入力データと出力データの流れを明確にする。ＤＦＤを用いてデータの流れを記述する。

㋒　データの変化する最大抽象入力点と最大抽象出力点を見つける。



主要なデータの流れを問題構造に沿って追いかけると、入力データが入力データの形態を無くする状態が現れる。この境界を最大抽象入力点という。主要なデータの流れを問題構造の終了点から逆に追っていくと、出力データが出力データの形態を無くする状態が現れる。この境界を最大抽象出力点という。

㋓　問題構造を最大抽象点を境に、源泉、変換、吸収の３つの主要な部分に分割し、直接従属モジュールの定義を行う。

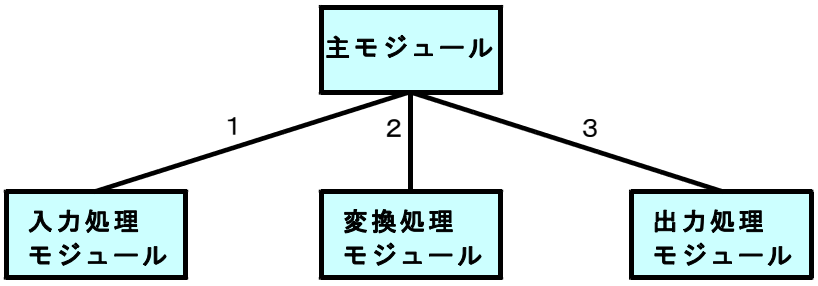
最大抽象入力点と最大抽象出力点の２点を境にして、入力側を入力処理(源泉)、出力側を出力処理(吸収)、その中間を変換処理(変換)に分け、それぞれを１つの機能としてモジュールを定義する。

㋔　上位モジュールとのインタフェースを定義する。

モジュール間の結合度が最小になるようにモジュールを分割し、階層構造図を作成する。モジュール間インタフェースを定義し、入出力するデータを明確にする。入出力インタフェース表を作成する。

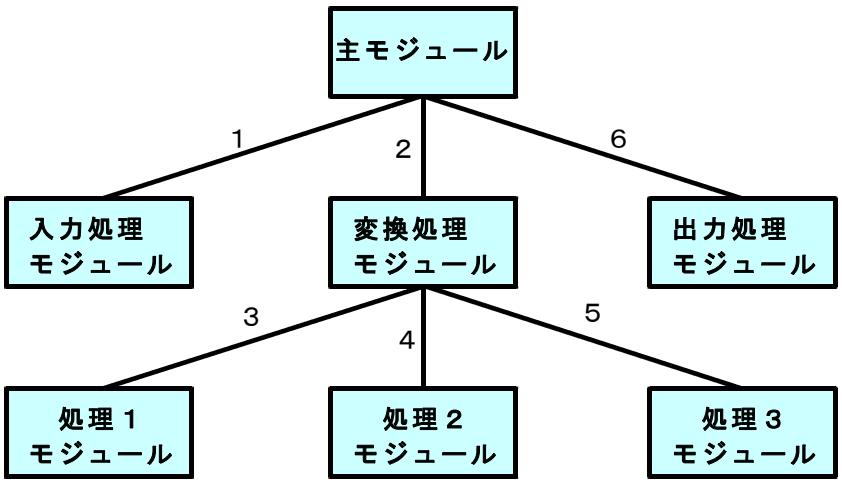
㋕　分割を繰り返す。

各モジュールを同様にして分析し、直接従属モジュールに分割する。すべてのモジュールを入力・変換・出力に分割し、インタフェースを定義する。モジュールの大きさは50～100ステップになる。



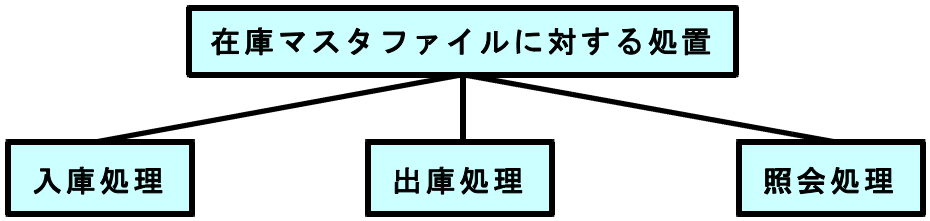
ⓓ　トランザクション（ＴＲ）分割技法とは

プログラムのトランザクションの種類に応じて、異なる処理機能をその機能単位にモジュールに分割していく方法である。



ⓔ　共通機能分割技法とは

共通の従属機能を取り出して、別個のモジュールとして定義する。同一のファイルに対する複数の機能をまとめる方法である。



③　ＳＴＳ分割の具体例

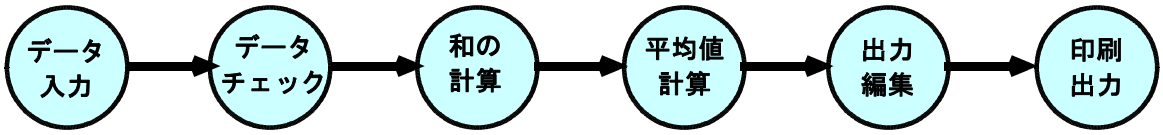
ⓐ　「データの平均値をもとめる」分割の手順

㋐　次の６つの機能に分割する。

❶　データを入力する。（データ入力）
❷　データをチェックする。（データチェック）
❸　入力データの和を求める。（和の計算）
❹　入力データの和をデータ数で割り平均値を求める。（平均値計算）
❺　結果のデータを出力様式に編集する。（出力編集）
❻　印刷出力する。（印刷出力）

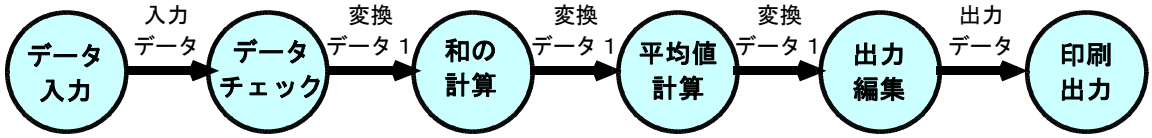
㋑　機能の再編成

主要なデータの流れとして、左から右に各機能を配置し、矢印で結ぶ。



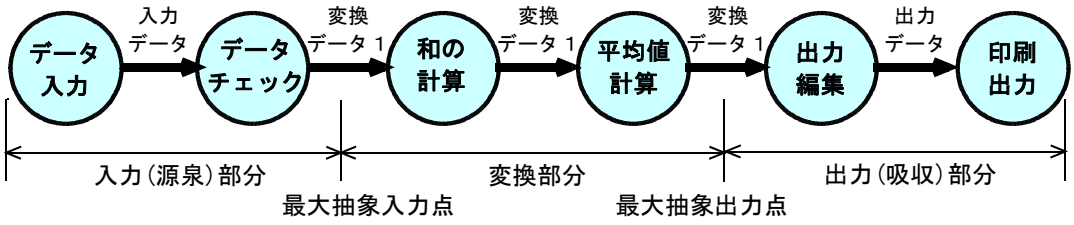
㋒　データの洗い出し

各機能間のデータに順次、入力データ、変換データ１、変換データ２、変換データ３、出力データの名前を付ける。変換データ１はチェック後のデータであり、変換データ２は入力データの和とデータの個数であり、変換データ３は平均値とその関連データである。



㋓　最大抽象点の決定

一連のデータの並びから、最大抽象入力点、最大抽象出力点を求める。最大抽象入力点は入力データが最大に抽象化される点であり、それ以降のデータはシステム内のデータとして取り扱われる。変換データ１以降はシステム内のデータとして取り扱われる。最大抽象入力点はデータをチェックすると入力データの和を求めるの間にある。最大抽象出力点は出力データが最大に抽象化される点であり、それ以前のデータはシステム内のデータとして取り扱われる。最大抽象出力点は平均値を求めると結果のデータを出力様式に編集するの間にある。



㋔　ＳＴＳに分割する。

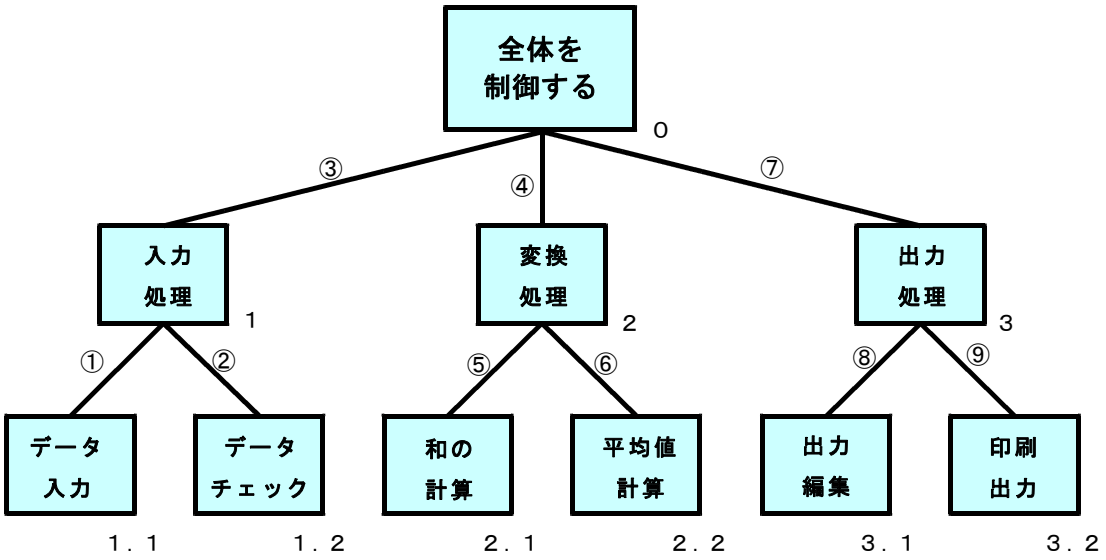
データ入力から最大抽象入力点までが源泉部分即ち入力部分になる。最大抽象入力点から最大抽象出力点までが変換部分になる。最大抽象出力点からデータ出力までが吸収部分即ち出力部分になる。

㋕　モジュール構造への変換

３つのモジュールの呼出を制御するために、全体を制御するモジュールを親モジュールとして設定する。入力部分、変換部分、出力部分を制御するモジュールを設定する。モジュール階層構造図を作成する。

㊩　モジュール間インタフェースの確定

モジュール間のインタフェース番号を設定する。データの流れをもとに各モジュール間の情報の受け渡しを決定する。インタフェース表を作成する。出力は階層構造の下位から上位に向かう方向になる。



インタフェース表

番号	入力	出力
①		入力データ
②	入力データ	変換データ１
③		変換データ１
④	変換データ１	変換データ３
⑤	変換データ１	変換データ２
⑥	変換データ２	変換データ３
⑦	変換データ３	
⑧	変換データ３	出力データ
⑨	出力データ	

④　ジャクソン法

ⓐ　ジャクソン法の図的表現法

㋐　基本

核となる構成要素であり、これ以上分割できないもので、１つのデータ項目、１つのステートメントに該当する。

㋑　連続

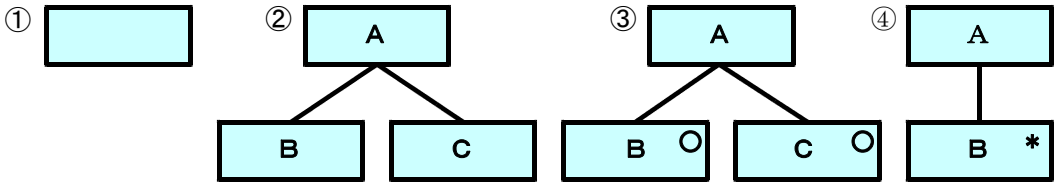
決まった順序で現れる２つ以上の相異なる基本要素で構成される構成要素で、複数のデータ項目を持つレコードに相当する。

㋒　選択

複数の部分のうち１つを選択するような構成要素で、条件によって判断を必要とする処理である。

㋓　繰返

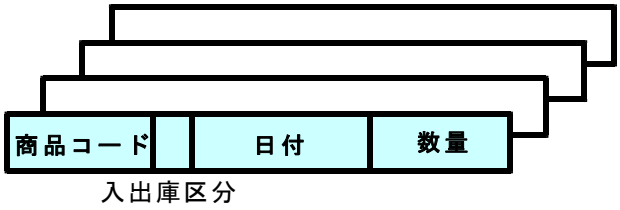
繰返しは同一の構成要素が繰り返し現れる構成要素で、配列や順次ファイルのレコードに相当する。



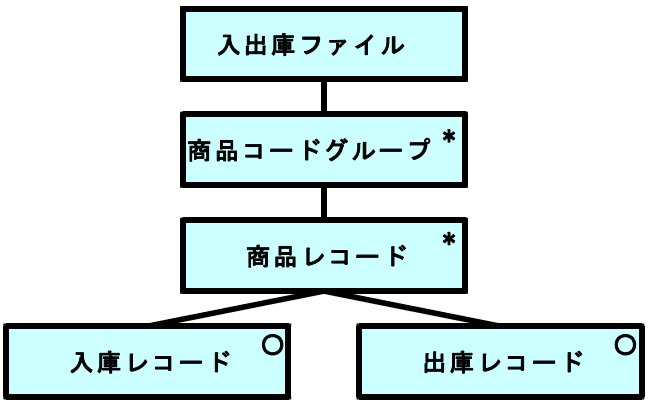
ⓑ　ジャクソン法の手順

㋐　データ構造の分析

入出庫ファイルの構造



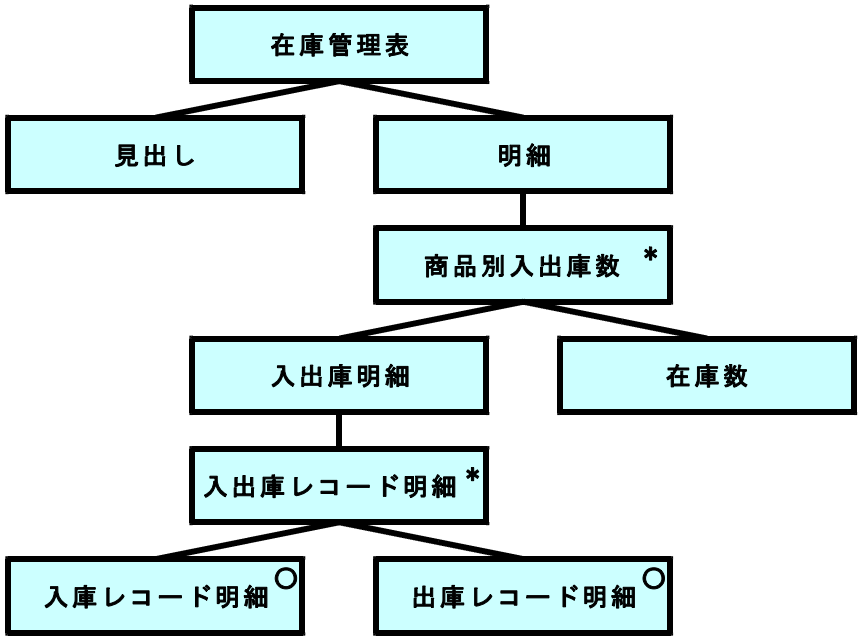
入出庫ファイルのデータ構造



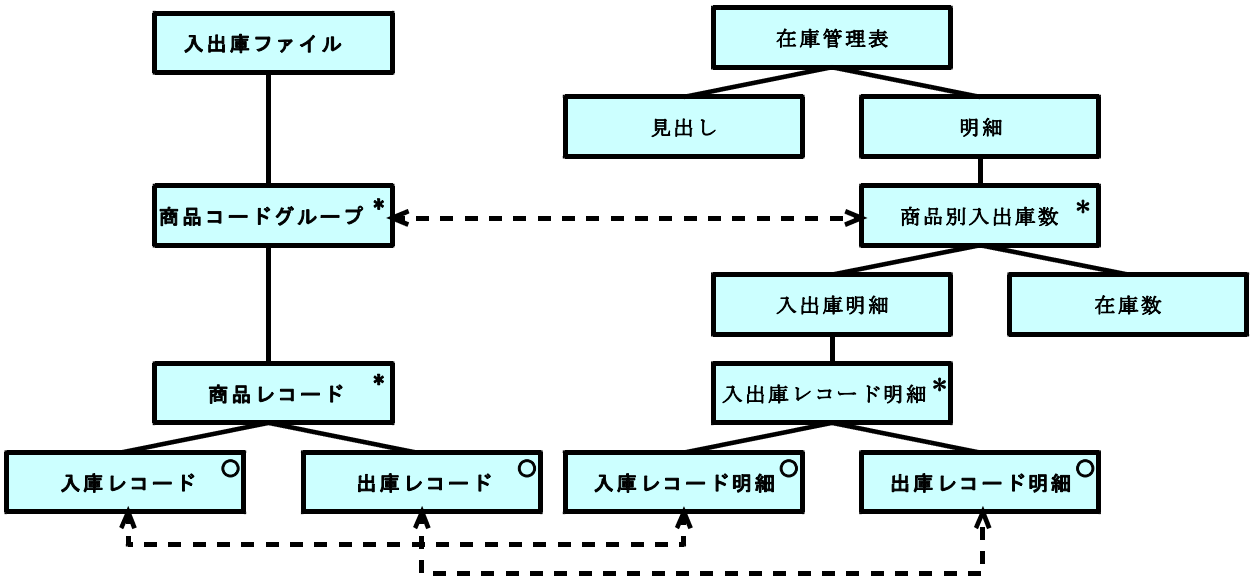
処理対象の入力と出力のデータを分析して、それぞれのデータ構造を明確にし、データ構造図を作成する

㋑　プログラム構造の作成

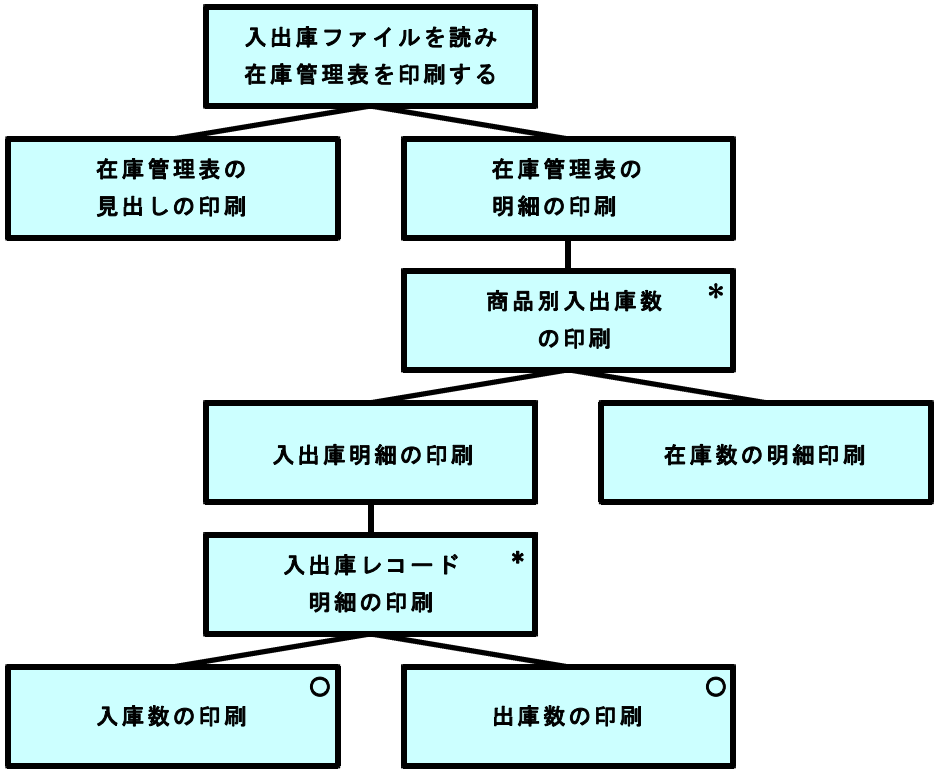
在庫管理表のデータ構造



定義した入力と出力のデータ構造の構成要素間の一対一の対応を探し、図式表示する。プログラムは入力から出力への変換であるから、一対一の対応を探すことによって、プログラムの機能を定義できる。図に示すように、商品コードグループに商品別入出庫数が対応する。入庫レコードに入庫レコード明細が対応する。出庫レコードに出庫レコード明細が対応する。商品コードグループの入力データから、商品別在庫数の結果を含む商品別入出庫数の出力データに変換するプログラム機能が必要になる。そのプログラム機能は、入庫レコードの入力データが入庫レコード明細の出力データに、出庫レコードの入力データから出庫レコード明細の出力データに変換する機能を含んだものとして構成される。



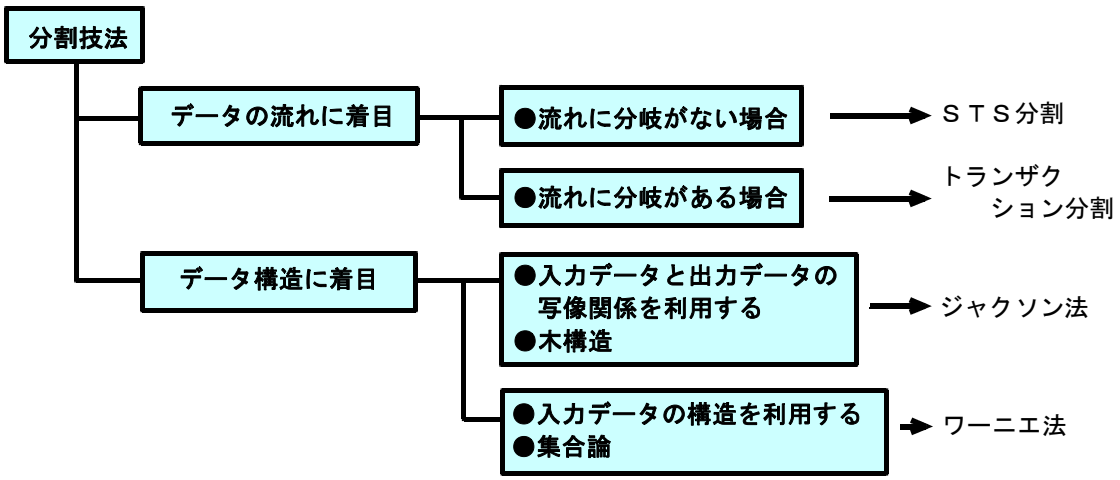
㋒　作業の定義と命令の割り当て



プログラム構造図を作成する。プログラム構造図は出力データ構造図に対応したものである。出力データ構造の繰り返し部分を作り出すにはプログラムに繰り返し構造が必要である。出力データ構造の選択の部分を作り出すにはプログラムに選択の構造が必要である。プログラム構造を表現する構成要素はデータ構造を表現する構成要素を使用する。プログラム構造ができるとプログラム構造の各構成要素に基本的な命令を割り当てる。入出力のデータ構造が複雑な場合、中間のデータ構造を作成し一対一の対応を見出す必要がある。

⑤　分割技法の選択法

ⓐ　分割技法の選択基準の流れ



⑥　モジュールの独立性

ⓐ　モジュールの特徴

㋐　複数のステートメント群が語彙としてまとまっていて、境界識別子で区切られ、単独の名前が付いている。

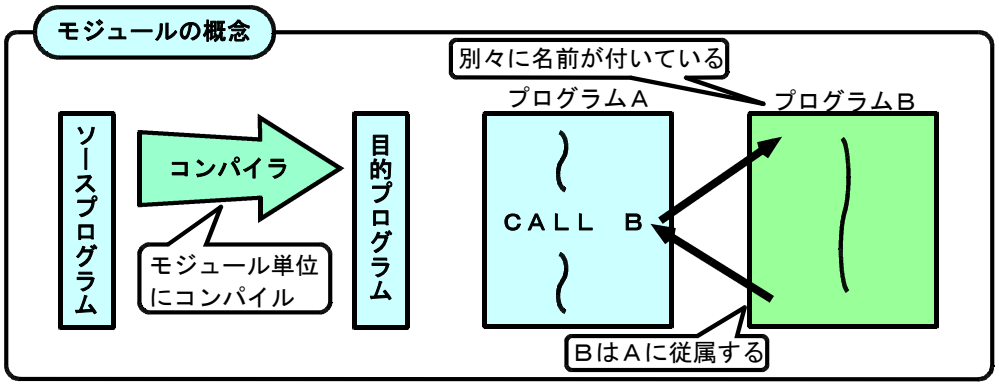
㋑　コンパイルが別々にできる。

㋒　その他のモジュールで呼び出し可能である。

㋓　呼び出したモジュールの実行は、呼び出されたモジュールの実行が終わるまで待たされる。

㋔　呼び出したモジュールの実行が終わると、実行は呼び出したモジュールに再び戻る。

ⓑ　モジュールの独立性を高める考え方



㋐　モジュール内部での関連性を最大になるようにする。

個々のモジュール内部の関連性を表す概念をモジュール強度という。モジュール強度を最大にする

㋑　モジュール間の関連性が最小になるようにする。

モジュール間の関連性を表現する概念をモジュール結合度という。モジュール結合度を最小にする。

⑦　モジュールの強度

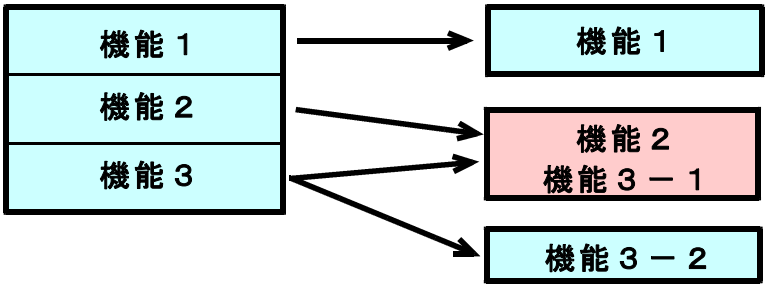
ⓐ　モジュール強度とは

モジュール強度の大きいモジュールほど独立性の高いモジュールである。従って、モジュール強度の大きいモジュールが望ましいモジュールである。モジュールの強度は、機能的強度、情報的強度、連絡的強度、手順的強度、時間的強度、論理的強度、暗号的強度の順に強度が大きい。モジュール強度が最も大きいのは機能的強度である。

ⓑ　モジュール強度の内容

㋐　暗号的強度

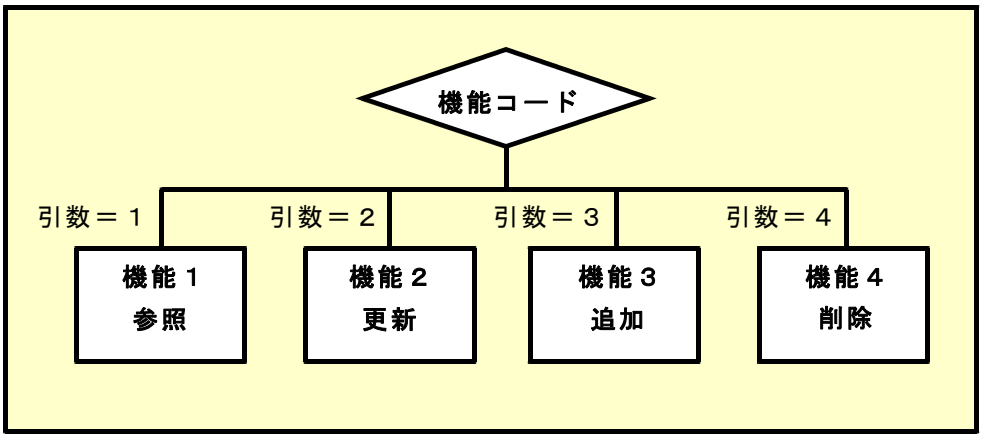
モジュール内の要素間に特別の関係が認められないモジュール化である。主記憶域の制限のために、既存のモジュールを単純に複数個に分割した場合や重複したステートメント・パターンを統合してモジュール化した場合が該当する。そのモジュールの機能を定義することができない。複数の全く関係のない機能あるいは機能の一部を実行する。モジュール内の各要素は、自モジュール内の他の要素との関連性が弱い反面、他のモジュール内の要素との強い関連を持つ傾向にある。他のモジュールの変更にも大きな影響を受けることが多く、保守がやりにくい。特定の機能が定義できないということは、再使用できる可能性が小さい。



㋑　論理的強度

ある機能を論理的にとらえてモジュール化したものである。入出力操作をまとめてすべて扱うモジュールを作る場合やすべてのデータを編集するモジュールをつくる場合が該当する。モジュール内の論理は、条件によって実行パスが異なる。関連したいくつかの機能を含み、

そのうちの１つが呼び出したモジュールによって識別され、実行される。機能的には共通するものなので、一部の論理は共有できる。モジュールが呼び出されたときモジュール内のすべてのステートメントが実行されるわけではなく、それだけ強度は弱くなる。呼び出すモジュールとのインタフェースにパラメータの取り扱いが必要になり、ミスを誘いやすくなる。特定のデータ処理を１つのモジュールに局所化できる効果はある。



㋒　時間的強度

複数の逐次的な機能を実行するが、実行する機能間に強い関連性がない。初期設定モジュールや終了処理モジュールが該当する。テーブルの初期処理は該当モジュールで実行するが、テーブルの実際の処理機能は、他のモジュールで行う。同じテーブルを介して、他のモジュールとの強い関連性を持つことになる。テーブルの変更が発生したとき、複数のモジュールが影響を受ける。

㋓　手順的強度

問題を処理するために関係している複数個の機能のうちのいくつかを実行するようなモジュールである。複数機能の中の１つだけを実行する場合は、選択機能が必要になり、論理的強度に似たものになる。大きな機能の一部の手順をモジュール化した場合、フロー・チャートの一部をモジュール化した場合が該当する。

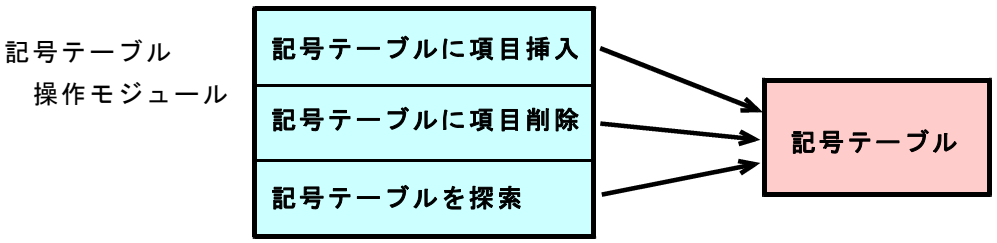
㋔　連絡的強度

手順的強度の性質をもち、その上モジュール内要素間でデータの受け渡しがあったり、同じデータを参照しているようなモジュールである。モジュール内要素がデータに関してつながっているだけ手順強度より強度は強い。

㋕　情報的強度

特定のデータ構造を扱う複数の機能を一つのモジュールにまとめたものである。同一データ構造は、特定のモジュールだけでアクセスしようという発想である。データの修正時の影響をこのモジュール内に限定することができる。データの機密保護の面からも有利である。論理的強度モジュールは１つの入り口点を持ち、実行機能の選択はパラメータを利用してい

る。情報的強度モジュールは複数個の入り口点を持ち、各入り口点は単一の固有の機能を実行する。



㋖　機能的強度

モジュール内のすべての要素が、単一機能を実行するために関連しあっている。強度としては最も強い。機能的強度を持ったモジュールの変更は、そのモジュールだけで処理でき、他のモジュールへの影響は他の強度より小さい。

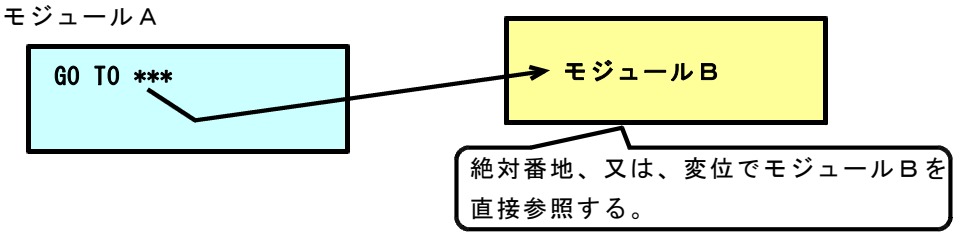
⑧　モジュールの結合度

ⓐ　モジュール結合度とは

モジュール結合度の小さいモジュールほど独立性の高いモジュールである。従って、モジュール結合度の小さいモジュールが望ましいモジュールである。モジュールの結合度は、データ結合、スタンプ結合、制御結合、外部結合、共通結合、内容結合の順に、結合度は小さい。データ結合がモジュール結合度が最も小さいモジュールである。

ⓑ　モジュール結合度の内容

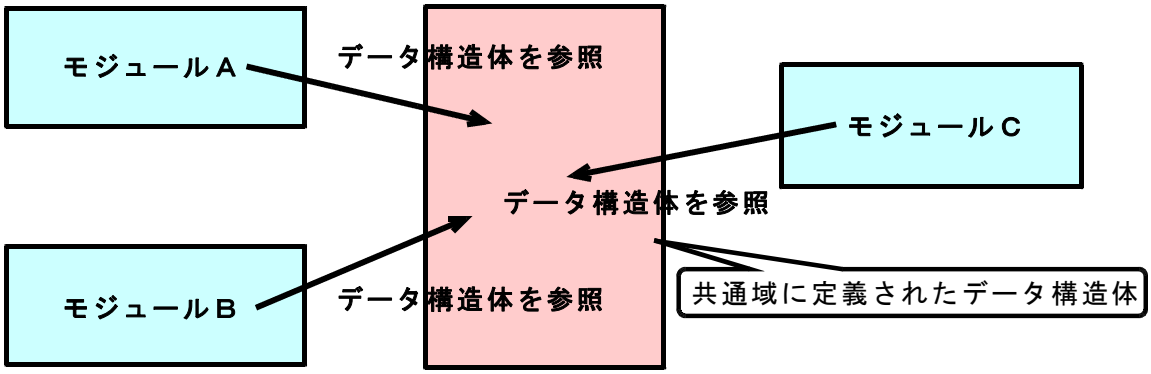
㋐　内容結合



あるモジュールと他のモジュールが一部を共有するモジュールの結合の仕方である。他のモジュール内の外部宣言していないデータを直接参照する場合や命令の一部を共有する場合が該当する。アセンブラ言語使用のモジュールに発生する。一方のモジュールの変更が他のモジュールに影響を及ぼすことになる。他のモジュールのことまで考えて変更する必要がある。

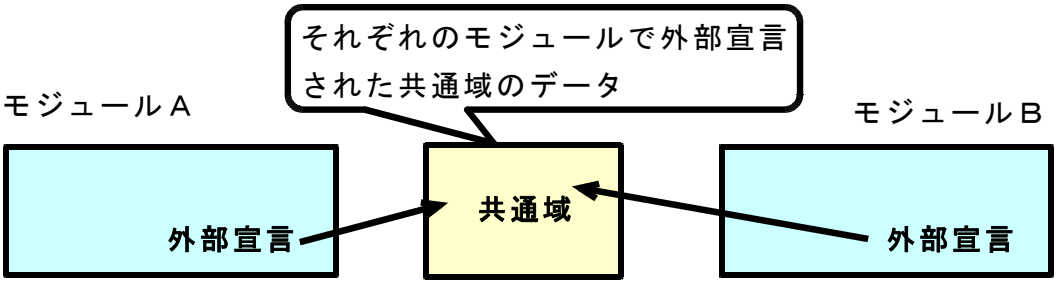
㋑　共通結合

共通域に定義したデータ構造体を複数のモジュールが共有するような結合の方法である。共通域の一部のデータ構造体の修正により、共通域参照の全モジュールのコンパイルのし直しが必要になったり、共通域のデータは、もともとそのデータに無関係なモジュールでもその気になれば使えるため、結合度が高いモジュールといえる。共通結合しているモジュールを他のモジュールが使用したいとき、使いにくい。また、共通域のデータはモジュール間のインタフェース上に現れないので、ソースコードによるモジュール機能の解読を難しくする。モジュールＡの都合でデータＸの長さを変更したとき、本来その変更に無関係なモジュールＢ、Ｃも再コンパイルの対象になる。モジュールＡの設計者がデータＸがモジュールＣで使用されていることを知らない場合には思わぬトラブルが発生する可能性がある。



㋒　外部結合

外部宣言したデータを共有する場合である。共有結合とよく似ているが、必要なデータだけを外部宣言するので不必要なデータまで共有することがないのでそれだけ結合度が弱くなる。



㋓　制御結合

呼び出すモジュールが、呼び出されるモジュールの制御を指示するデータをパラメータとして渡すやり方である。スイッチとかインデックスをパラメータとして渡す。呼び出し側は呼び出されるモジュールの論理を知っている必要がある。相手をブラックボックス化できないだけ、結合度が強くなる。呼び出されるモジュールの強度は論理的強度になる。データの共有といったことがないので、共通結合や外部結合よりも結合度は弱くなる。

㊔　スタンプ結合

共通域にないデータ構造を２つのモジュールで受け渡しするような場合である。データ構造を受け渡ししながら、構造内の一部データを使用しない場合である。不必要なデータまで受け渡しする点がデータ結合よりも結合度が強くなる。データ構造を受け渡ししても、構造のすべてのデータが処理されるならば、データ結合と見なせる。

㊕　データ結合

モジュール間のインタフェースとして、スカラ型のデータ要素だけをパラメータとして受け渡す。相手のモジュールをブラックボックスとして扱うことが可能となり、結合度が一番弱くなる。複合設計では結合度の一番弱いデータ結合を良しとしている。モジュール間の結合は単純に明確化されたパラメータでデータを受け渡しする型が一番良いのである。モジュールＡは入力としてＸをモジュールＢに渡し、出力としてＹをもらうことがわかっておればそれで良いのである。モジュールＢがＸをどのようにしてＹに変換するのか、その手順はＡには必要ないことである。モジュールＢの手順変更によってモジュールＡが影響を受けることはない。モジュールＡがモジュールＢのデータを直接参照する内容結合のようなときは、モジュールＢの変更がモジュールＡにそのまま影響を及ぼすことになる。このような場合は保守がやりにくく、トラブル発生の原因になる。プログラムのモジュール化をはかる場合、モジュールを機能的にまとめ、モジュール間のインタフェースはデータ結合で行うようにすれば、モジュールの独立性が高くなり、信頼性の高いプログラムが設計できることになる。

⑨　ソフトウェア設計と複合設計

ⓐ　ソフトウェア設計とは

ソフトウェアの設計は、要求されたコンピュータシステムに対して、ソフトウェアを実装するために、段階的に具体化していく作業のことである。ソフトウェアの要求を、プログラム構造とデータ構造に変換していく工程であり、分割と統合という考え方を利用して、ソフトウェアの機能を階層的にいくつかの機能に分割していくと、階層化されたある機能は、いくつかの独立した小さな機能単位から構成されるようになる。

ソフトウェアを階層的に構成する個々の機能は、システム、サブシステム、プログラム、モジュールに相当する。システムからモジュールへと段階的に詳細化していくプロセスがソフトウェア設計に相当する。

ⓑ　複合設計

プロセスを設計する技法に、処理するプロセスに着目しながら進める考え方とデータに着目しながら進める考え方、データとプロセスを一体化して進める考え方がある。それぞれ、プロセス指向設計、データ指向設計、オブジェクト指向設計という。

複合設計は、構造化設計技法を取り入れたもので、モジュール構造とそのインタフェースを設計する段階で、モジュール強度を最大に、モジュール強度を最小にすることによって、モジ

ュールの独立性を高める考え方である。モジュールの独立性が高いと言うことは、個々のモジュールが独自の機能だけで実現しているとともに、他のモジュールへの波及効果が少ないことになる。これによって、保守性を高めることが可能になる。具体的には、構造化設計技法で使用されるＳＴＳ分割法、ＴＲ分割法が用いられる。

ⓒ　複合設計の手順

㋐　問題の構造の概観を記述する。

データの流れを基本にして、３～10個の機能で問題の記述を行う。

㋑　問題の中の主要な入力データと出力データの流れを明らかにする。

㋒　主要な入力データの流れを問題の構造に沿ってトレースする。

入力データは最初はっきり入力の形をとっているが、徐々に形を変え、そのうちに、最早入力データと言えなくなる点に到達する。これが最大抽象入力点である。

㋓　同様の分析を主要出力データの流れについて行う。

最大抽象出力点を求める。

㋔　最大抽象入力点と最大抽象出力点を基準に、問題構造を三つの主要部分に分割する。

源泉部分、中央変換部分、吸収部分に分かれる。

㋕　モジュール間のインタフェースを定義する。

インタフェースはデータ結合になるように配慮する。

㋖　更に、下位のモジュールについても同様な分析をする。

ⓓ　独立性の評価基準

〔独立性〕

モジュール強度		↓	モジュール結合度	
暗合的強度	弱	低	強	内容結合
論理的強度	↓	↓	↓	共通結合
時間的強度				外部結合
手順的強度				制御結合
連絡的強度				スタンプ結合
情報的強度				データ結合
機能的強度	強	高	高	非直接結合

例題演習

プログラムのモジュール化に関する記述として、適切なものはどれか。

ア　異なったモジュール間の情報の受け渡しは、ファイルを仲介にして行う。
イ　主記憶装置の容量を許す限り、モジュールを大きくし、その個数を少なくすべきである。
ウ　プログラミングは容易になるが、保守は難しくなる。
エ　モジュール単位にコーディング、デバッグが可能になるので、プログラム開発、テストの生産性向上に役立つ。

解答解説

モジュール化に関する問題である。

アのモジュール間の情報の受け渡しは、データで行う。ファイルで行うのはプロセスフローにおける処理プログラム間の場合である。

イのモジュールの大きさは、可能な限り小さくしモジュール間の結合度を弱くするのが適切な処理であり、大きくして、個数を少なくするのはモジュールの独立性から問題がある。

ウの適切なモジュール化は、モジュール強度が大きく、モジュール結合度が弱くなるため、モジュールの独立性が高くなり、保守を容易にする。

エのモジュール単位のコーディング、デバッグが可能になり、開発やテストの生産性が向上するは適切な記述である。求める答えはエである。

例題演習

モジュール分割技法の中で，データの流れに沿って，入力処理機能，変換機能，出力処理機能へと分割する技法はどれか。

ア　ＳＴＳ分割　　イ　共通機能分割
ウ　ジャクソン法　　エ　トランザクション分割

解答解説

モジュール分割技法のＳＴＳ分割に関する問題である。

アのＳＴＳ分割は、プログラムを入力データ処理、入力データから出力データへの変換処理、出力データ処理の３つのモジュールに分割する手法である。求める答えはアである。

イの共通機能分割は、共通的な機能を取り出して、複数のモジュールで実行できる独立したモジュールとして定義する方法である。

ウのジャクソン法は、データ構造からプログラム構造を決めるデータ中心の構造化技法で、基本、連続、選択、繰り返しの４つの基本図式を用いるジャクソン図を使って、データ構造とプログラム構造を階層構造で表す。

エのトランザクション分割は、オンライントランザクション処理などのプログラムで、トランザクションの種類ごとにモジュールを分割する手法である。

例題演習

次に示すプログラム構造化設計の手順のうち、正しい手順はどれか。

Ａ　分割技法の選択　　Ｂ　インタフェースの定義
Ｃ　最上位モジュールの定義　　Ｄ　モジュールの機能分析
Ｅ　モジュール分割　　Ｆ　分割すべき他のモジュールの検討

ア　Ａ→Ｄ→Ｃ→Ｅ→Ｂ→Ｆ　　イ　Ａ→Ｃ→Ｄ→Ｅ→Ｂ→Ｆ
ウ　Ｃ→Ｄ→Ａ→Ｅ→Ｂ→Ｆ　　エ　Ｃ→Ａ→Ｄ→Ｅ→Ｂ→Ｆ

解答解説

プログラムの構造化設計の手順に関する問題である。

プログラム構造化設計の手順は、最上位モジュールの定義（Ｃ）、モジュールの機能分析（Ｄ）、分割技法の選択（Ａ）、モジュールの分割（Ｅ）、インタフェースの定義（Ｂ）、分割すべき他のモジュールの検討（Ｆ）の順に行う。Ｃ→Ｄ→Ａ→Ｅ→Ｂ→Ｆの順になり、求める答えはウとなる。

例題演習

“データを読み込み、数字データだけを選んでその平均値を表示する”プログラムをＳＴＳ分割したとき、それぞれの機能は吸収、源泉、変換のどの部分に分類されるか。

	機　　能			
	データ入力	数字選択	平均値算出	表示
ア	吸収	吸収	源泉	変換
イ	吸収	源泉	源泉	変換
ウ	源泉	源泉	変換	吸収
エ	源泉	変換	変換	吸収

解答解説

平均値を求めるプログラムのＳＴＳ分割に関する問題である。

源泉は入力処理部分、吸収は出力処理部分であるから、データ入力は源泉、表示は吸収になる。平均値算出は変換に相当する。数字選択は入力のデータ構造で実行されるものであり、源泉に相当する。従って、源泉・源泉・変換・吸収になる。求める答えはウとなる。

例題演習

オンラインリアルタイム処理のように、入力トランザクションの種類に応じて処理が異なる場合に有効な分割技法はどれか。

ア　ＳＴＳ分割　　イ　ジャクソン法　　ウ　共通機能分割　　エ　ＴＲ分割

解答解説

トランザクション分割に関する問題である。

アのＳＴＳ分割は、プログラムを入力データ処理、入力データから出力データへの変換処理、出力データ処理の３つのモジュールに分割する手法である。

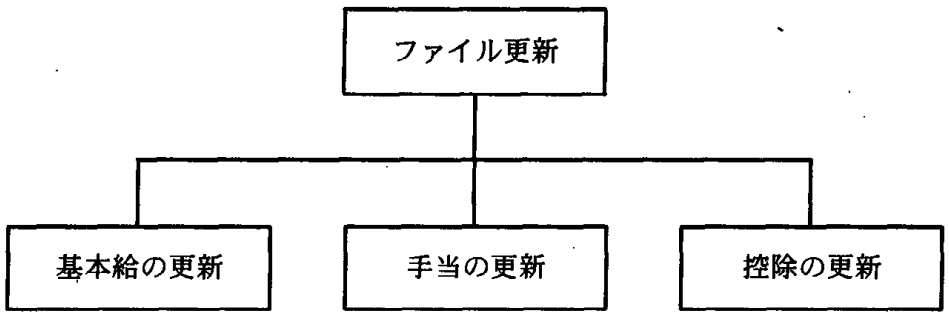
イのジャクソン法は、データ構造からプログラム構造を決めるデータ中心の構造化技法で、基本、連続、選択、繰り返しの４つの基本図式を用いるジャクソン図を使って、データ構造とプログラム構造を階層構造で表す。

ウの共通機能分割は、個々のプログラムで共通する処理を、独立したモジュールとするモジュール分割法である。

エのトランザクション分割は、データの流れに着目した分割技法で、データの流れに分岐がある場合に用いる。入力トランザクションの種類に応じて処理が異なる場合の手法であるから、トランザクション分割であり、求める答えはエとなる。

例題演習

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　ワーニエ法

解答解説

トランザクション分割に関する問題である。

アのＳＴＳ分割は、プログラムを入力データ処理、入力データから出力データへの変換処理、出力データ処理の３つのモジュールに分割する手法である。

イのジャクソン法は、データ構造からプログラム構造を決めるデータ中心の構造化技法で、基本、連続、選択、繰り返しの４つの基本図式を用いるジャクソン図を使って、データ構造とプログラム構造を階層構造で表す。

ウのトランザクション分割は、オンライントランザクション処理などのプログラムで、トランザクションの種類ごとにモジュールを分割する手法である。

エのワーニエ法は、入出力データを集合としてとらえる事務処理向きの構造化設計手法で、順次、選択、繰り返しでデータ構造を表現する。

階層構造図では、基本給の更新、手当の更新、控除の更新とモジュールが処理機能別に分けられている。これはトランザクション分割である。求める答えはウとなる。

例題演習

構造化分析で作成したデータフロー図から、構造化設計における構造化チャートへの変換する技法はどれか。

ア　ＫＪ法　　イ　ＯＭＴ法
ウ　ジャクソン法　　エ　トランザクション分割法

解答解説

構造化分析手法のトランザクション分割に関する問題である。

アのＫＪ法は、断片的な情報を整理するための手法で、名刺大のカードに発言を要約し、カードの内容の親近性によって分類し、図解し、文章化する手順で情報の整理を行う手法である。

イのＯＭＴ法は、オブジェクト指向を使ってシステム分析／設計開発方法論の一つである。システムをオブジェクトモデル、動的モデル、機能モデルを使って表現する。

ウのジャクソン法は、データ構造からプログラム構造を決めるデータ中心の構造化技法で、基本、連続、選択、繰り返しの４つの基本図式を用いるジャクソン図を使って、データ構造とプログラム構造を階層構造で表す。

エのトランザクション分割は、データの流れに着目した分割技法で、データの流れに分岐がある場合に用いる。トランザクションを入力するモジュール、トランザクションの属性ごとに振り分けるモジュール、それぞれのトランザクションごとに変換するモジュールに分ける。

ＤＦＤを利用して、階層構造図の構造化チャートに変換する技法はトランザクション分割であり、求める答えはエとなる。

例題演習

データ構造に着目したモジュール分割技法はどれか。

ア　共通機能分割　　イ　源泉／変換／吸収分割（ＳＴＳ分割）
ウ　ジャクソン法　　エ　トランザクション分割（ＴＲ分割）

解答解説

データ構造に着目したモジュール分割技法に関する問題である。

アの共通機能分割は、データの流れに着目した分割技法で、個々のプログラムで共通する処理を、独立したモジュールとするモジュール分割法である。

イのＳＴＳ分割は、データの流れに着目した分割技法で、プログラムを入力データ処理、入力データから出力データへの変換処理、出力データ処理の３つのモジュールに分割する手法である。

ウのジャクソン法は、データの構造に着目して、入力データ、出力データの構造、構成要素のデータ項目に対応した形で分割を行う。求める答えはウとなる。

エのトランザクション分割は、データの流れに着目した分割技法で、データの流れに分岐がある場合に用いる。

例題演習

プログラムの構造化設計におけるワーニエ法の説明として、適切なものはどれか。

ア　扱うデータの構造に着目して、入出力データの構造図を作成し、次に入力データの構造図をもとにプログラム構造図を作成する方法である。
イ　扱うデータの流れに着目し、データフロー上の機能を源泉(source)、変換(transform)、吸収(sink) にグループ分けする方法である。
ウ　ソフトウェアをデータとプロセスの集合としてとらえ、一本化することによってモジュールとしての独立性を高める方法である。
エ　プログラムの制御構造に着目し、呼出し関係を表す制御フローに基づいて論理設計を行う方法である。

解答解説

ワーニエ法に関する問題である。

ワーニエ法は、１９７０年代の初めにフランスのワーニエによって確立された構造的プログラミング技法である。主として事務処理分野での利用を目指している。ファイルやデータの構造の分析を行い、これをもとにプログラムの構造的分析であるワーニエ図を作成する。作成手順は、問題の分析、データ構造の分析、データ構造からプログラム構造への変換、ワーニエ図からプログラム流れ図への手順で進める。データ分析では、入力データの構造と出力データの構造を集合論を利用して分析する。ファイルを一つの集合としてとらえ、部分集合に細分化していく。

アはワーニエ法、イはＳＴＳ分割法、ウはオブジェクト指向分析設計法、エは構造化チャートを利用した構造化手法である。求める答えはアとなる。

例題演習

システム設計およびプログラム構造の作成に集合論を適用しているのを特徴としているのはどれか。

ア　ジャクソン法　　イ　ワーニエ法
ウ　段階的詳細化技法　　エ　ブラックボックス法

解答解説

データ構造に着目したワーニエ法に関する問題である。

アのジャクソン法は、データ構造からプログラム構造を決めるデータ中心の構造化技法で、基本、連続、選択、繰り返しの４つの基本図式を用いるジャクソン図を使って、データ構造とプログラム構造を階層構造で表す。木構造を利用した分割技法である。

イのワーニエ法は、入出力データを集合としてとらえる事務処理向きの構造化設計手法で、

順次、選択、繰り返しでデータ構造を表現する。データ構造に着目した分割技法である。求める答えはイとなる。

ウの段階的詳細化は、上位から下位へ段階的に大きな機能を小さい機能に分割しながら設計を進めていく方法である。

エのブラックボックス法は、モジュールをブラックボックスと見なし、機能仕様に基づき作成したテストデータでテストを行う手法である。

例題演習

プログラムのモジュール化設計においてモジュールの分割を行う際には、それぞれの独立性に留意することが大切である。モジュールの独立性を評価するための尺度に関する概念はどれか。

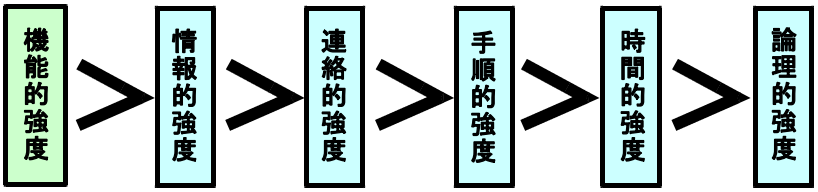
ア　モジュール間結合度　　イ　モジュール深度
ウ　モジュール精度　　エ　モジュール難易度

解答解説

モジュールの独立性の評価に関する問題である。

モジュール強度は、モジュール強度は独立性を高めるための要素の１つで、強度の高いほど独立性が高いモジュールである。強度の高低の分類は、部分の機能の自己完結度や凝縮度で分けることができる。

モジュール強度は、機能的強度、情報的強度、連絡的強度、手順的強度、時間的強度、論理的強度、暗号的強度に分類され、機能的強度が最も独立性が高い。



モジュールの独立性を評価する尺度には、モジュール強度とモジュール間結合度がある。この問題ではモジュール間結合度が該当する。求める答えはアとなる。

例題演習

モジュール内部の関連性を示す尺度にモジュールの強度がある。次のうち、強度の強い順に並んでいるのはどれか。

ア　暗号的強度、情報的強度、手順的強度、連絡的強度、時間的強度
イ　機能的強度、手順的強度、情報的強度、連絡的強度、論理的強度
ウ　機能的強度、情報的強度、連絡的強度、手順的強度、時間的強度
エ　情報的強度、連絡的強度、手順的強度、時間的強度、暗号的強度

解答解説

モジュール強度に関する問題である。

モジュール内部の関連性を表すモジュール強度の種類

① 暗号的強度は、モジュール内の要素間に特別な関係が認められない場合で、既存モジュールを単純に分解したり、重複したコーディングを統合したりする場合である。

② 論理的強度は、関連したいくつかの機能を含み、そのうちの一つが別のモジュールによって選択される。モジュール内のすべてのステートメントが実行されるわけではない。

③ 時間的強度は、複数の逐次的な機能を実行するが、実行する機能間にはあまり強い関連性がない。

④ 手順的強度は、問題を処理するために関係している複数個の機能のうちいくつかを実行するようなモジュールである。

⑤ 連絡的強度は、手順的強度の性質を持ち、その上、モジュール内要素間でデータの受け渡しがあったり、同じデータを参照したりするモジュールである。

⑥ 情報的強度は、特定のデータ構造を扱う複数の機能を一つのモジュールにまとめたもの。

⑦ 機能的強度は、すべての要素が一つの機能を実行するために関連し合っているモジュールで、一番強い強度のモジュールである。

上の説明の①～⑦はモジュール強度が弱い順に並んでいる。最も強度が強いのは機能的強度である。従って、求める答えは、機能的強度、情報的強度、連絡的強度、手順的強度、時間的強度の順になる。求める答えはウとなる。

例題演習

モジュール間の関係を示す尺度のモジュールの結合度の大きい順に並んでいるのはどれか。

ア　スタンプ結合、データ結合、内容結合、外部結合、制御結合

イ　内容結合、外部結合、制御結合、スタンプ結合、データ結合

ウ　共通結合、外部結合、制御結合、データ結合、スタンプ結合

エ　データ結合、スタンプ結合、制御結合、外部結合、共通結合

解答解説

モジュール結合度に関する問題である。

モジュール間の関連性を表すモジュール結合度の種類

① 内容結合は、他のモジュール内の外部宣言していないデータを直接参照する。依存度が非常に高く、変更が他に影響しやすい。

② 共通結合は、共通域に定義したデータをいくつかのモジュールが共有するような結合の方法である。

③ 外部結合は、外部宣言したデータの共有である。必要なデータだけを外部宣言するので、不必要なデータまで共有することはない。結合度は弱くなる。

④ 制御結合は、呼び出すモジュールが、呼び出されるモジュールの制御を指示するデータをパラメータとして渡すやり方である。スイッチとかインデックスをパラメータとして渡す。データ結合に比べて結合度が強くなる。データの共有がないので外部結合より結合度

は弱い。

⑤　スタンプ結合は、共通域にないデータ構造を二つのモジュールで受け渡しする場合。不必要なデータまで受け渡しする点がデータ結合より結合度が強くなる。

⑥　データ結合は、モジュール間のインタフェースとして、スカラ型のデータ要素だけをパラメータとして受け渡す。相手のモジュールをブラックボックスとして扱うことが可能になり、結合度は一番弱い。

モジュールの結合度の大きい順に並べると、

内容結合、共通結合、外部結合、制御結合、スタンプ結合、データ結合

となり、求める答えはイとなる。

例題演習

モジュールの独立性の尺度であるモジュール結合度は，弱いほど独立性が高くなる。次のうち，モジュールの独立性が最も高い結合度をもつものはどれか。

ア　共通結合　　イ　スタンプ結合　　ウ　データ結合　　エ　内容結合

解答解説

モジュール結合度に関する問題である。

アの共通結合は、プログラムの共有領域に定義したデータを関係する各モジュールがそのデータを共有する方式である。共有するデータの構造が変更になった場合、共通領域のデータ定義部分と各モジュールのデータ定義部分の修正が必要となる。結合度が強くモジュールの独立性が高くない。

イのスタンプ結合は、モジュール間の引数としてデータ構造名を渡してモジュール間でデータを共有する。共通領域にデータ領域を定義しないので、モジュール間の関連性は弱くなる。

ウのデータ結合は、モジュール間はデータ要素のみ引数で受け渡す。モジュールを修正しても影響を及ぼす範囲が少ない。データ結合はモジュールの独立性が最も高い。

エの内容結合は、他のモジュールのある部分を直接参照する方式で、結合度が最も強い。

独立性が最も高い結合度はデータ結合で、求める答えはウとなる。

例題演習

ソフトウェアのモジュール設計において、信頼性、保守性を向上させるためのアプローチとして、望ましいものはどれか。

ア　モジュール強度を高く、結合度を高くする。
イ　モジュール強度を高く、結合度を低くする。
ウ　モジュール強度を低く、結合度を高くする。
エ　モジュール強度を低く、結合度を低くする。

解答解説

モジュール設計の独立性に関する問題であ。

モジュールの信頼性、保守性を高めるためにはモジュールの独立性を確保する必要があり、独立性を確保するためにはモジュールの強度を高くし、結合度を低くする必要がある。求める答えはイである。

例題演習

モジュール結合度が最も弱いモジュールはどれか。

ア　単一のデータ項目を大域的データで受け渡すモジュール
イ　単一のデータ項目を引数で受け渡すモジュール
ウ　データ構造を大域的データで受け渡すモジュール
エ　データ構造を引数で受け渡すモジュール

解答解説

モジュール結合度に関する問題である。

モジュール結合度の小さいモジュールほど独立性の高いモジュールである。従って、モジュール結合度の小さいモジュールが望ましいモジュールである。モジュールの結合度は、データ結合、スタンプ結合、制御結合、外部結合、共通結合、内容結合の順に、結合度は小さい。データ結合がモジュール結合度が最も小さいモジュールである。

アは外部結合、イはデータ結合、ウは共通結合、エはスタンプ結合であり、求める答えはイとなる。

例題演習

モジュール強度が最も高いものはどれか。

ア　あるデータを対象として逐次的に複数の機能を実行するモジュール
イ　異なる入力媒体からのデータを処理するモジュール
ウ　単一の機能を実行するモジュール
エ　特定の時点で必要とされる作業のすべてを含んでいるモジュール

解答解説

モジュール強度に関する問題である。

モジュール強度の大きいモジュールほど独立性の高いモジュールである。従って、モジュール強度の大きいモジュールが望ましいモジュールである。モジュールの強度は、機能的強度、情報的強度、連絡的強度、手順的強度、時間的強度、論理的強度、暗号的強度の順に強度が大きい。モジュール強度が最も大きいのは機能的強度である。

機能的強度は、モジュール内のすべての要素が、単一機能を実行するために関連しあっている。強度としては最も強い。機能的強度を持ったモジュールの変更は、そのモジュールだけで処理でき、他のモジュールへの影響は他の強度より小さい。

アは手順的強度、イは論理的強度、ウは機能的強度、エは時間的強度である。求める答えはウとなる。

例題演習

モジュールの独立性を高めるにはモジュール結合度を弱くする。モジュール間の情報の受渡しに関する記述のうち、モジュール結合度が最も弱いものはどれか。

ア　共通域に定義したデータを、関係するモジュールが参照する。
イ　制御パラメタを引数として渡し、モジュールの実行順序を制御する。
ウ　データ項目だけをモジュール間の引数として渡す。
エ　必要なデータだけを外部宣言して共有する。

解答解説

モジュール結合度に関する問題である。
結合度の弱い順は、データ結合、制御結合、外部結合、共通結合の順になる。
アは共通結合、イは制御結合、ウはデータ結合、エは外部結合である。
結合度が最も低いのはデータ結合であり、求める答えはウとなる。

例題演習

図は，五つの関数Ａ～Ｅからなるプログラムの構成を表したものである。表に，これらの関数間のインタフェースを示す。図中の番号は関数間のインタフェースを示し，表中の項目“Ｎｏ”の列と対応している。このほかに，関数Ａ，Ｄ，Ｅ は，特定のデータ領域を参照するという関係がある。関数ＡとＥの間のモジュール結合関係はどれか。

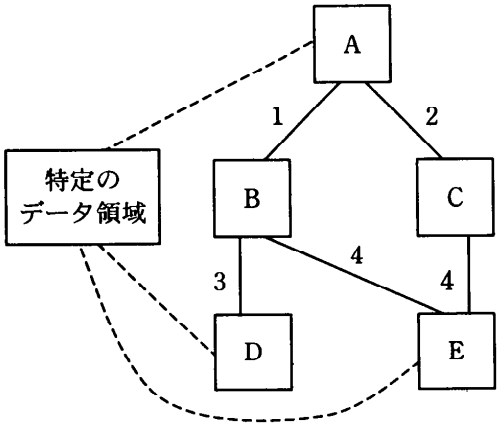


図　プログラムの構成

表　関数間のインタフェース

No	引数	戻り値
1	データ項目の内容	データ項目の内容
2	データ項目の内容	データ項目の内容
3	機能コード	なし
4	なし	リスト

ア　共通結合　　イ　データ結合
ウ　内容結合　　エ　なし(非直接結合)

解答解説

複合設計のモジュール結合に関する問題である。
アの共通結合は、共通域に定義したデータをいくつかのモジュールが共有するような結合の方法である。
イのデータ結合は、モジュール間のインタフェースとして、スカラ型のデータ要素だけをパ

ラメータとして受け渡す。

ウの内容結合は、他のモジュール内の外部宣言していないデータを直接参照する。

ＡとＥは特定のデータ領域を共に参照しているため共通結合であり、求める答えはアとなる。

例題演習

モジュール設計書を基にモジュール強度を評価した。適切な評価はどれか。

〔モジュール設計書（抜粋）〕

上位モジュールから渡される処理コードに対応した処理をする。処理コードが“Ｉ”のときは挿入処理，処理コードが“Ｕ”のときは更新処理，処理コードが“Ｄ”のときは削除処理である。

ア　これは“暗合的強度”のモジュールである。モジュール内の機能間に特別な関係はなく，むしろ他のモジュールとの強い関係性をもつ可能性が高いので，モジュール分割をやり直した方がよい。

イ　これは“情報的強度”のモジュールである。同一の情報を扱う複数の機能を，一つのモジュールにまとめている。モジュール内に各処理の入口点を設けているので，制御の結びつきがなく，これ以上のモジュール分割は不要である。

ウ　これは“連絡的強度”のモジュールである。モジュール内でデータの受渡し又は参照を行いながら，複数の機能を逐次的に実行している。再度見直しを図り，必要に応じて更にモジュール分割を行った方がよい。

エ　これは“論理的強度”のモジュールである。関連した幾つかの機能を含み，パラメタによっていずれかの機能を選択して実行している。現状では大きな問題となっていないとしても，仕様変更に伴うパラメタの変更による影響を最小限に抑えるために，機能ごとにモジュールを分割するか，機能ごとの入口点を設ける方がよい。

解答解説

モジュールの論理的強度に関する問題である。

モジュール設計書によると、挿入処理、更新処理、削除処理の機能がコードとして、上位モジュールから渡されて、そのコードに基づいてモジュール内の処理が決まるモジュールである。複数の機能をパラメータによって制御する。

アの暗号的強度は、モジュール内の機能間に関係はなく、他のモジュールとの強い関係を持っている。

イの情報的強度は、複数の機能を持っているが、各処理に個別の入り口があるため、制御の結びつきがない。

ウの連絡的強度は、モジュール内の複数の機能を逐次的に実行する。

エの論理的強度は、モジュール内に複数の機能を持ち、パラメータにより機能を選択し実行するモジュールである。パラメータにより機能が制御される。求める答えはエとなる。