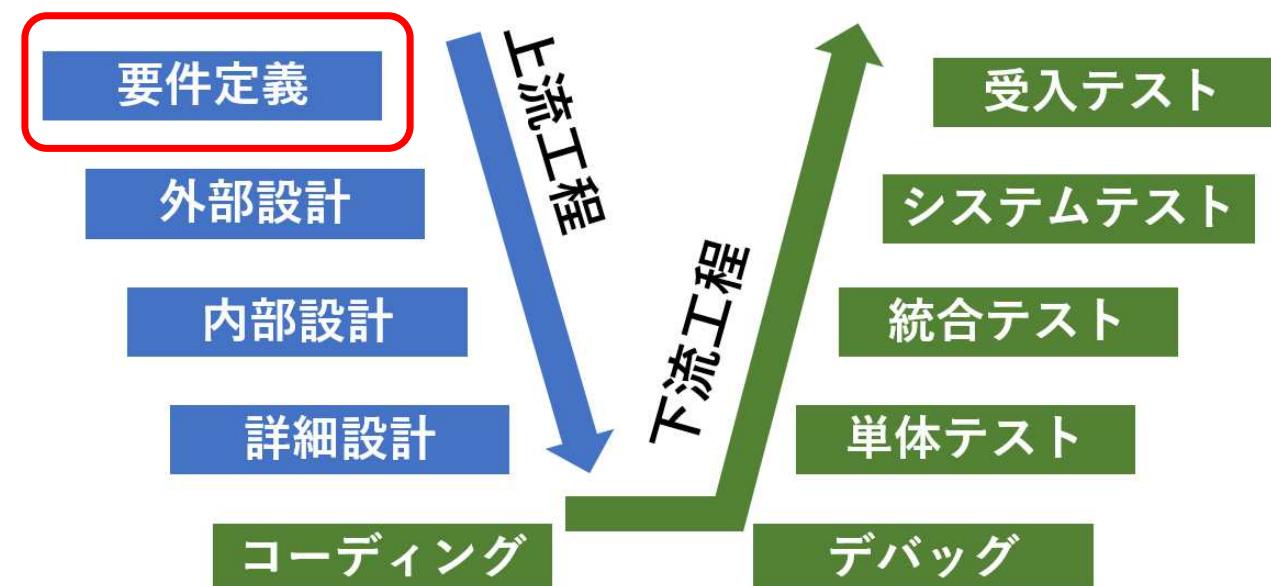


要件定義

システム化する範囲と機能を整理

- 要件定義
- 非機能要件

利害関係者の識別



システム方式設計

- ・ ソフトウェア、ハードウェア構成
- ・ ネットワーク構成
- ・ 機能の構成
- ・ 外部システムとの連携方式
- ・ 開発言語の

システム方式設計

非機能要件に注意



要件定義

外部設計

内部設計

詳細設計

コーディング

上流工程

下流工程

受入テスト

システムテスト

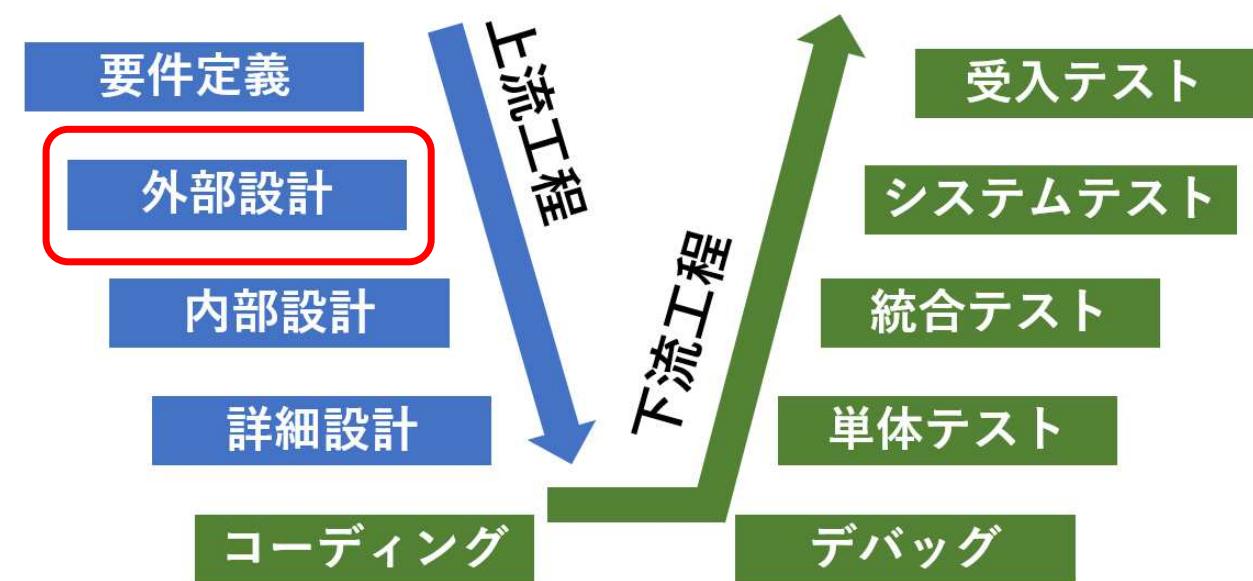
統合テスト

単体テスト

デバッグ

外部設計

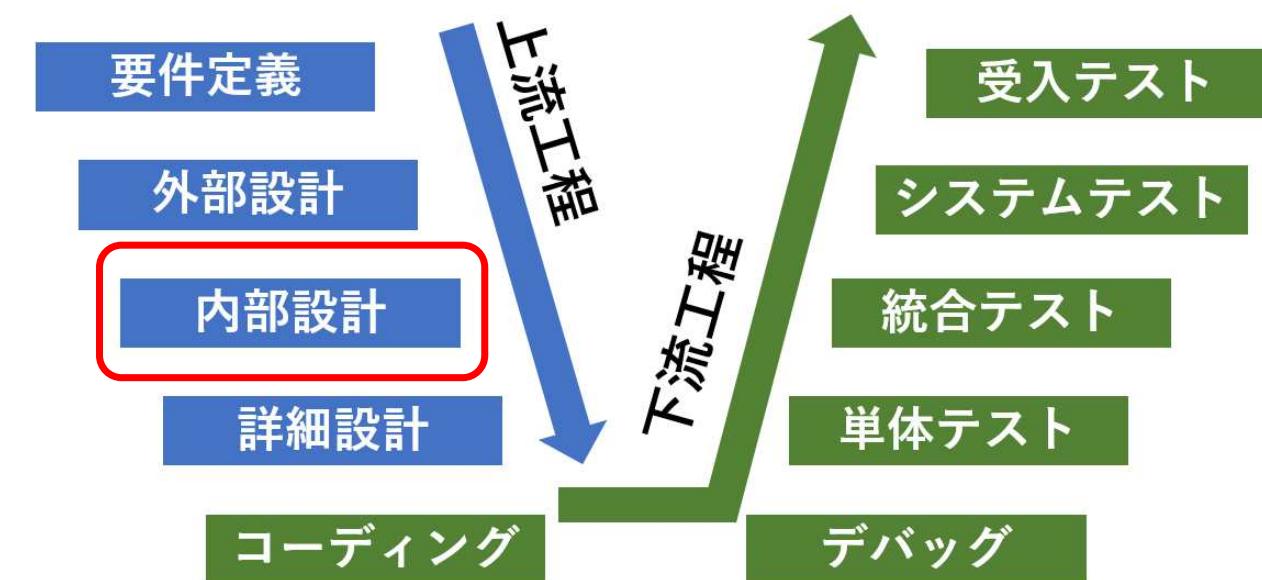
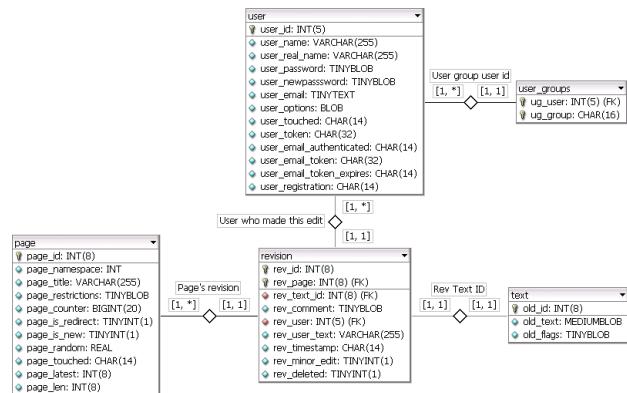
利用者から見た設計



内部設計

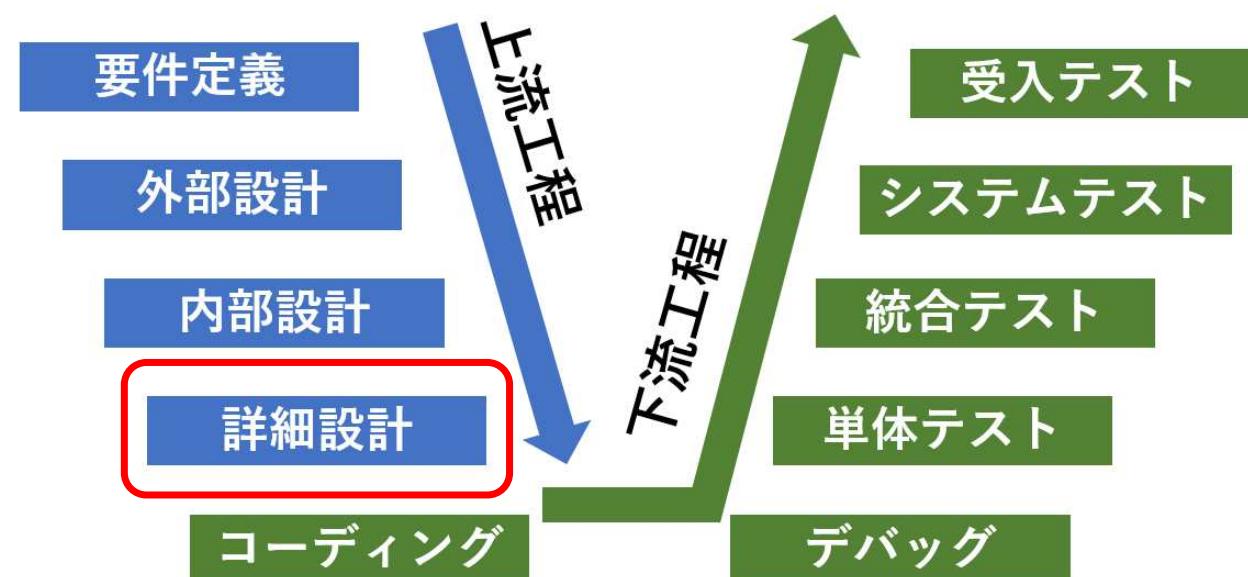
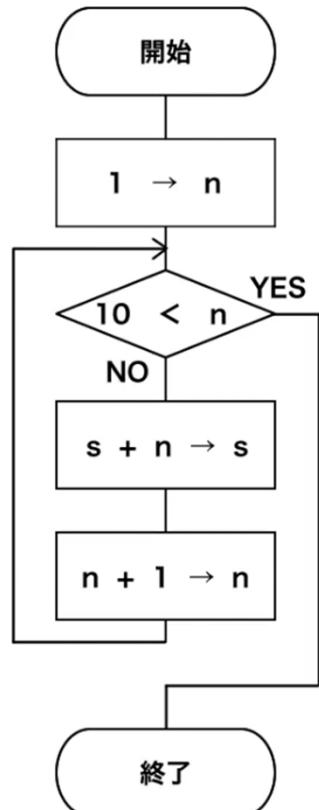
開発者から見た設計

```
Article
- name : String
- contents : String
+ PAGENAME_SUFFIX : String
+ getName() : String
+ setName(newName : String) : void
+ getContents() : String
+ setContents(newContents : String) : void
```



詳細設計

プログラマから見た設計



問65 共通フレーム 2013 によれば、要件定義プロセスで行うことはどれか。

ア システム化計画の立案

ウ ソフトウェア詳細設計

イ システム方式設計

エ 利害関係者の識別

問66 社内の業務システムの要件定義の承認を担う責任者に含まれるのはどれか。

- ア 開発要員を派遣している派遣元の責任者
- イ システムの運用・保守を担当している部門の責任者
- ウ 当システムの RFP 発行先の全てのベンダの責任者
- エ プロジェクト管理で使用する進捗管理ツールの提供元の責任者

問65 非機能要件項目はどれか。

- ア 新しい業務の在り方や運用に関わる業務手順，入出力情報，組織，責任，権限，業務上の制約などの項目
- イ 新しい業務の遂行に必要なアプリケーションシステムに関わる利用者の作業，システム機能の実現範囲，機能間の情報の流れなどの項目
- ウ 経営戦略や情報戦略に関わる経営上のニーズ，システム化・システム改善を必要とする業務上の課題，求められる成果・目標などの項目
- エ システム基盤に関わる可用性，性能，拡張性，運用性，保守性，移行性，セキュリティ，システム環境などの項目

問65 要件定義段階において、要求品質の向上のために発注者が留意すべきことはどれか。

- ア 現行システムと同じ機能の要求であっても、現行システムの機能や使われ方を調査して、要件定義を実施する。
- イ ビジネス要求の視点よりも、現行業務で使用されている機能が盛り込まれているか否かの視点で、要件定義の妥当性を検証する。
- ウ 要件定義書はあくまでも利用者ニーズの大枠を定めたものとして、実際には設計段階以降に、受注者と議論して具体的な要件を確定していく。
- エ 要件定義段階では業務要件を整理し、システムの移行方法・運用方法など非機能要件は、システム稼働前に洗い出す。

問42 システム開発の外部設計工程で行う作業はどれか。

ア 業務分析

イ 帳票設計

ウ 物理データ設計

エ プログラム設計

問44 内部設計書のデザインレビューを実施する目的として、最も適切なものはどれか。

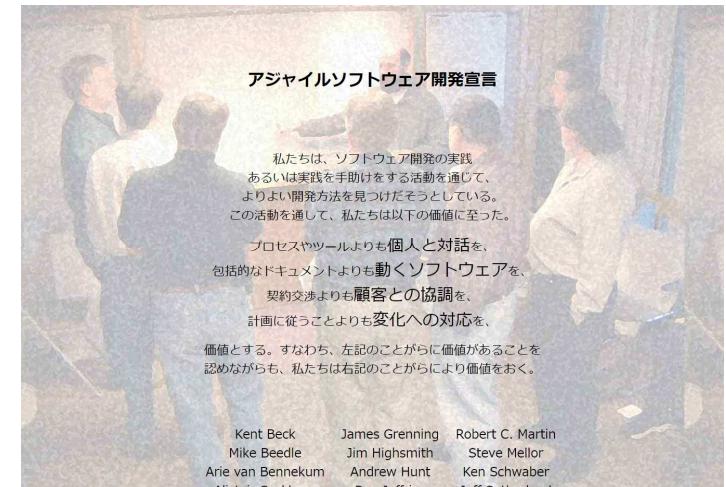
- ア 外部設計書との一貫性の検証と要件定義の内容を満たしていることの確認
- イ 設計記述規約の遵守性の評価と設計記述に関する標準化の見直し
- ウ 要件定義の内容に関する妥当性の評価と外部設計指針の見直し
- エ 論理データ設計で洗い出されたデータ項目の確認と物理データ構造の決定

迅速に無駄なく開発を行う手法

アジャイルソフトウェア開発宣言

- プロセスやツールよりも**個人と対話を**、
- 包括的なドキュメントよりも**動くソフトウェアを**、
- 契約交渉よりも**顧客との協調を**、
- 計画に従うことよりも**変化への対応を**、

価値とする。



XP (エクストリーム プログラミング)

計画よりも柔軟性を重視する

・イテレーション

アジャイル開発を繰り返す単位

スクラム

開発チームが一体となりプロジェクトを進めていく

・スプリント

アジャイル開発を繰り返す単位

・プロダクトオーナー

完成した製品の責任を持つ人

・スクラムマスター

プロジェクトの推進に責任を持つ人

・リファクタリング

外部から見た動作は変えず、プログラムだけを改善する。

・テスト駆動開発

プログラミングの前にテストを作成し常にテストにパスする
ようにプログラミングをしていく。

問49 エクストリームプログラミング（XP）におけるリファクタリングの説明はどれか。

- ア 外部から見た動作を変えずにプログラムをより良く作り直すこと
- イ コーチがチームメンバの意識を高めること
- ウ プログラミングとテストを繰り返し行うこと
- エ プログラムを作成するよりも先にテストケースを考えること

問50 エクストリームプログラミング（XP : eXtreme Programming）における“テスト駆動開発”の説明はどれか。

- ア 最初のテストでバグを抽出すること
- イ テストケースを順次改善すること
- ウ テストでのカバレージを優先すること
- エ プログラムを書く前にテストケースを作成すること

問49 アジャイル開発手法の説明のうち、スクラムのものはどれか。

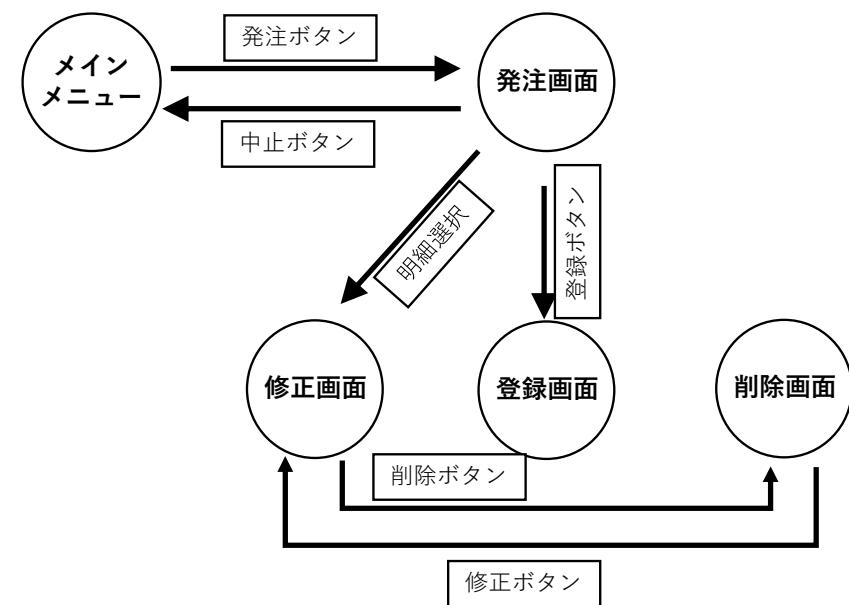
- ア コミュニケーション、シンプル、フィードバック、勇気、尊重の五つの価値を基礎とし、テスト駆動型開発、ペアプログラミング、リファクタリングなどのプラクティスを推奨する。
- イ 推測（プロジェクト立て上げ、適応的サイクル計画）、協調（並行コンポーネント開発）、学習（品質レビュー、最終 QA／リリース）のライフサイクルをもつ。
- ウ プロダクトオーナなどの役割、スプリントレビューなどのイベント、プロダクトバックログなどの作成物、及びルールから成るソフトウェア開発のフレームワークである。
- エ モデルの全体像を作成した上で、優先度を付けた詳細なフィーチャリストを作成し、フィーチャを単位として計画し、フィーチャ単位に設計と構築を繰り返す。

問50 アジャイル開発手法の一つであるスクラムでは、プロダクトオーナ、スクラムマスター、開発者でスクラムチームを構成する。スクラムマスターが行うこととして、最も適切なものはどれか。

- ア 各スプリントの終わりにプロダクトインクリメントのリリースの可否を判断する。
- イ スクラムの理論とプラクティスを全員が理解するように支援する。
- ウ プロダクトバックログアイテムを明確に表現する。
- エ プロダクトバックログの優先順位を決定する。

状態遷移図

状態
遷移



データフローダイアグラム

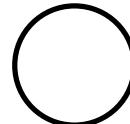
図式化



源泉/吸収



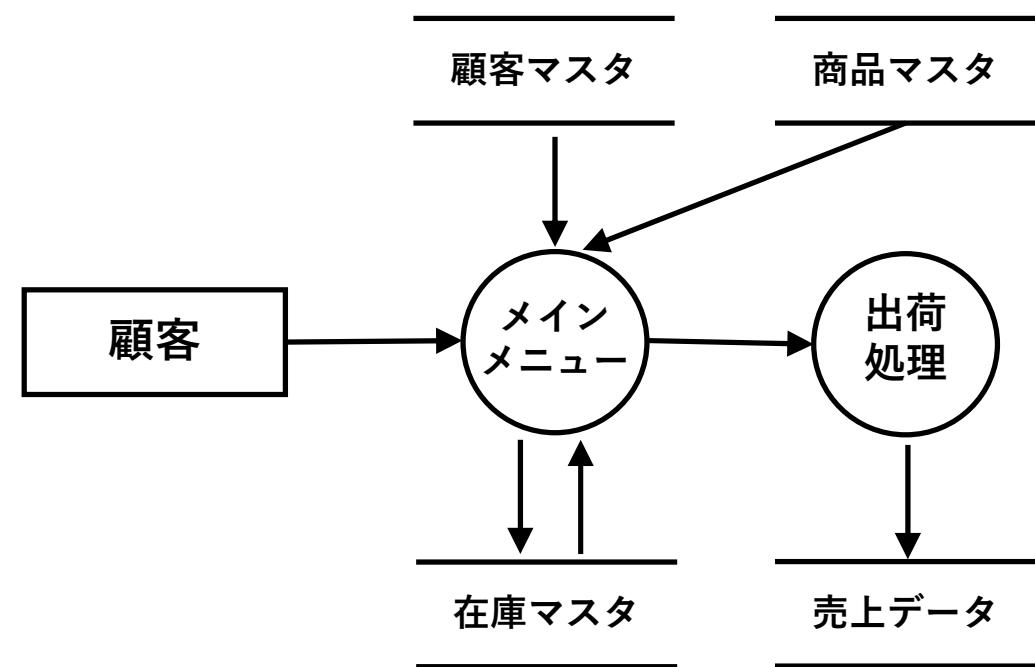
データストア



プロセス



データフロー

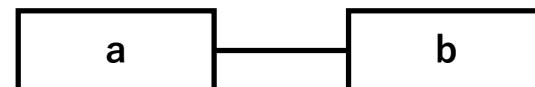




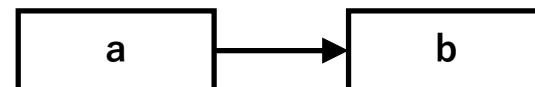
エンティティ



リレーションシップ



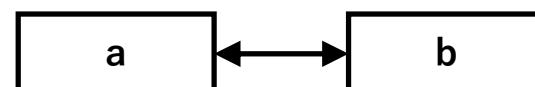
1対1の関係



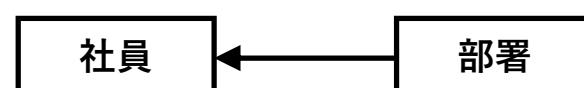
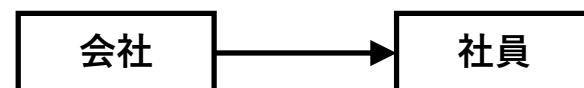
1対多の関係



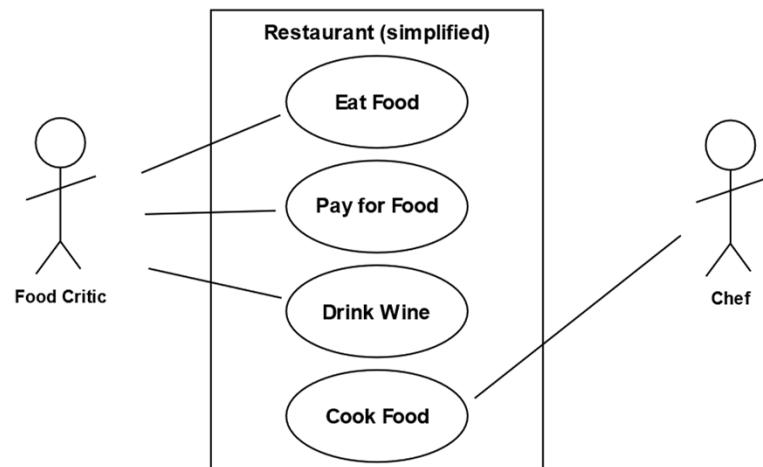
多対1の関係



多対多の関係

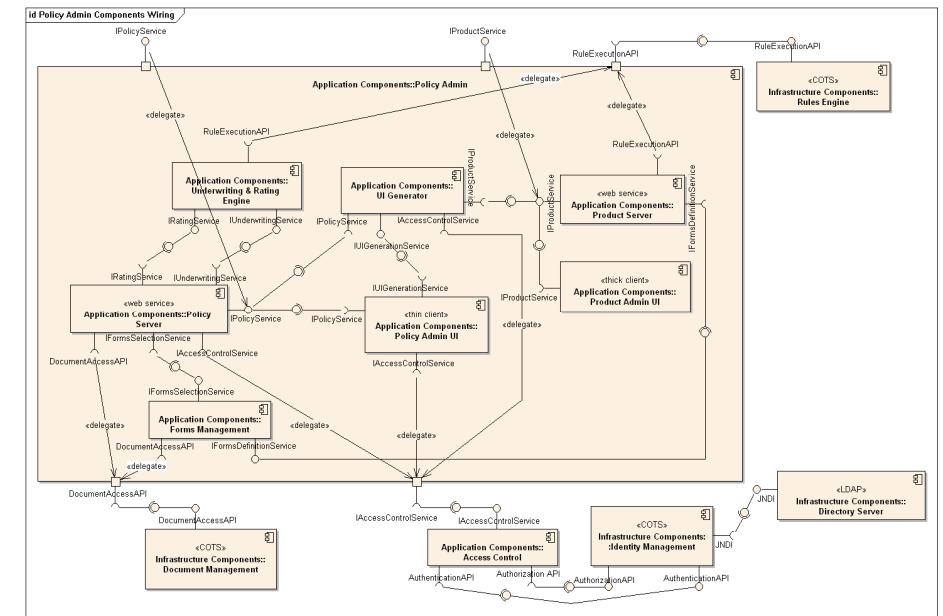


ユースケース図



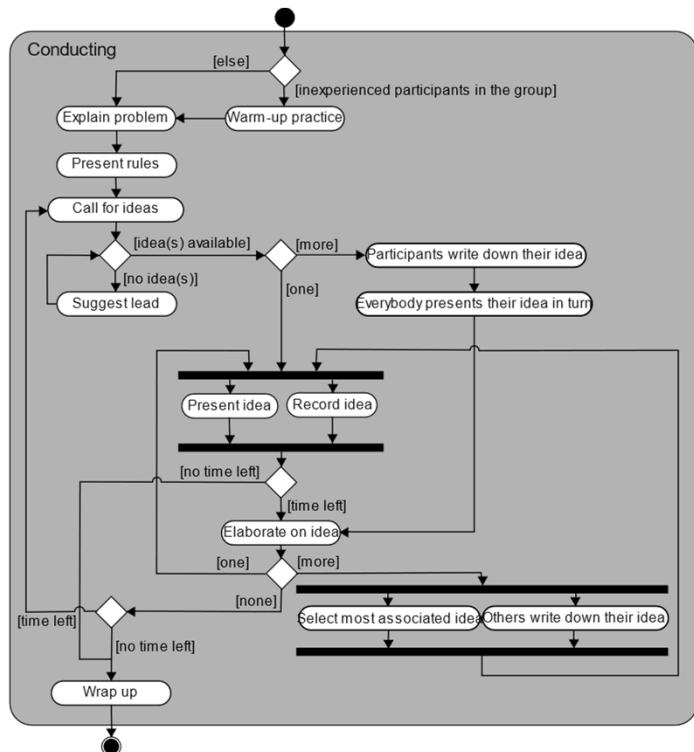
機能と利用者（または外部のシステム）との関係

コンポーネント図

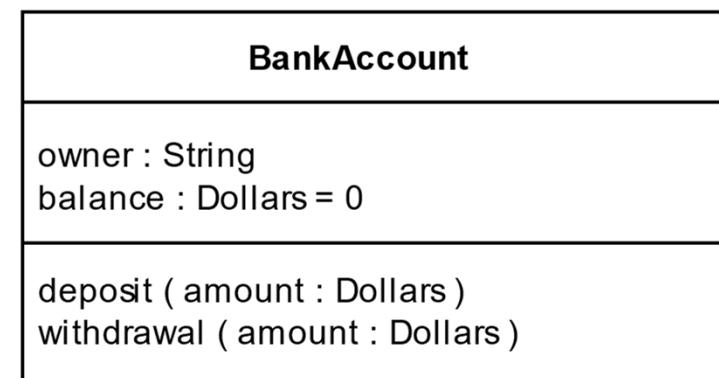


コンポーネント(部品)とインターフェースの関連

アクティビティ図



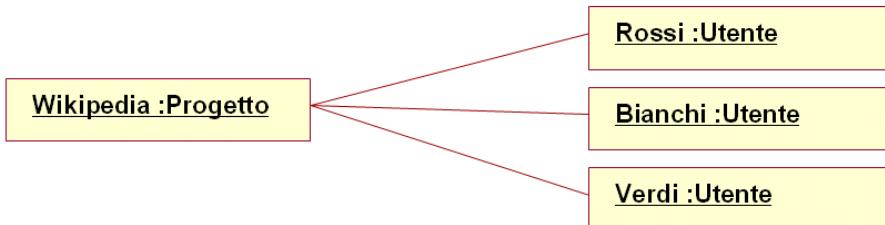
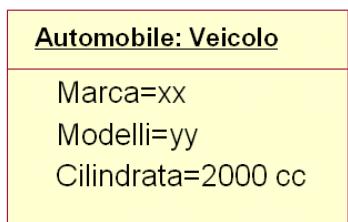
クラス図



クラスの仕様とクラス間の関連
ER図の発展形

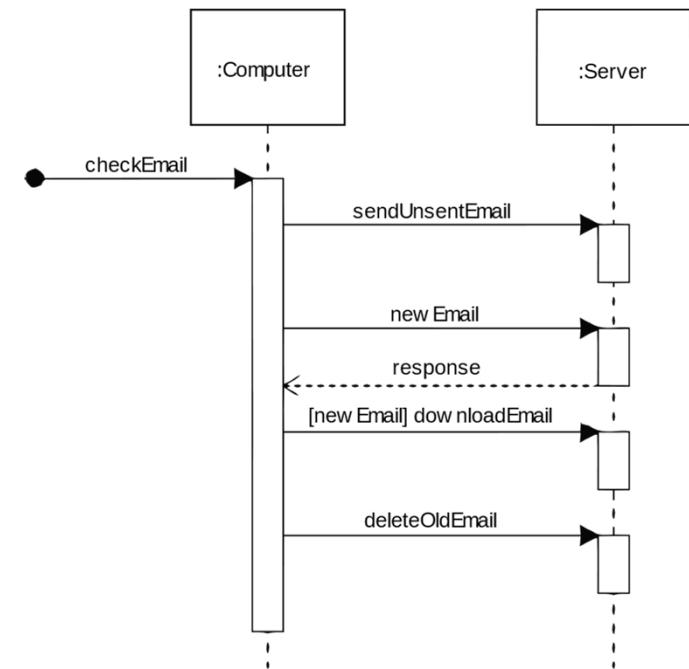
制御の流れを表現(フローチャートの発展)

オブジェクト図



特定のタイミングでのオブジェクトの状態と関連を表現

シーケンス図



インスタンス間の相互作用を時系列で表現

問48 プロセス制御などの事象駆動（イベントドリブン）による処理の仕様を表現する方法として、適切なものはどれか。

ア DFD

イ E-R 図

ウ クラス図

エ 状態遷移図

問29 顧客は一般に複数の銀行に預金するものとして、顧客と銀行の関連を、E-R 図で次のように表現する。このモデルを関係データベース上に“銀行”表，“口座”表，“顧客”表として実装する場合の記述として、適切なものはどれか。



- ア “銀行” 表から “口座” 表へのカーディナリティは多対 1 である。
- イ “銀行” 表中に参照制約を課した外部キーがある。
- ウ “口座” 表から “顧客” 表へのカーディナリティは 1 対多である。
- エ “口座” 表には二つ以上の外部キーがある。

問45 E-R 図の解釈として、適切なものはどれか。ここで、＊＊は多対多の関連を表し、自己参照は除くものとする。



- ア ある組織の親組織の数が、子組織の数より多い可能性がある。
- イ 全ての組織は必ず子組織をもつ。
- ウ 組織は2段階の階層構造である。
- エ 組織はネットワーク構造になっていない。

問44 新システムのモデル化を行う場合の DFD 作成の手順として、適切なものはどれか。

- ア 現物理モデル → 現論理モデル → 新物理モデル → 新論理モデル
- イ 現物理モデル → 現論理モデル → 新論理モデル → 新物理モデル
- ウ 現論理モデル → 現物理モデル → 新物理モデル → 新論理モデル
- エ 現論理モデル → 現物理モデル → 新論理モデル → 新物理モデル

問47 UMLにおける振る舞い図の説明のうち、アクティビティ図のものはどれか。

- ア ある振る舞いから次の振る舞いへの制御の流れを表現する。
- イ オブジェクト間の相互作用を時系列で表現する。
- ウ システムが外部に提供する機能と、それを利用する者や外部システムとの関係を表現する。
- エ 一つのオブジェクトの状態がイベントの発生や時間の経過とともにどのように変化するかを表現する。

問44 UML 2.0において、オブジェクト間の相互作用を時間の経過に注目して記述するものはどれか。

ア アクティビティ図

ウ シーケンス図

イ コミュニケーション図

エ ユースケース図

1 . 暗号

機能間の**関連を無視し**複数の機能を1つのモジュールにまとめた。

2 . 論理

関連した、いくつかの機能を1つのモジュールにまとめ、どの機能を使うかを**引数で指定**する。

3 . 時間

ある**時点**に連續して使用する機能（関連がなくても）を1つのモジュールにまとめた。

4 . 手順

必ず**順番（逐次）**に実行される機能を1つのモジュールにまとめた。別のデータを使う。

5 . 連絡

「手順的強度」に加え、同じデータを使う。

6 . 情報

特定のデータを扱う機能を1つのモジュールにまとめ**隠ぺい**した。

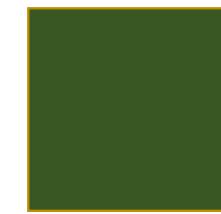
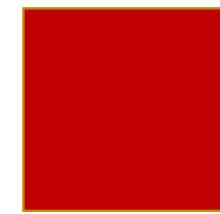
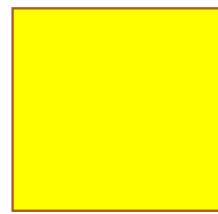
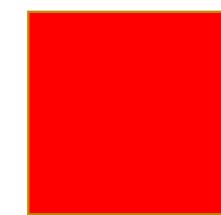
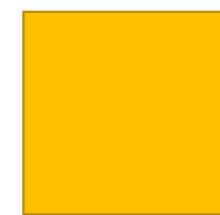
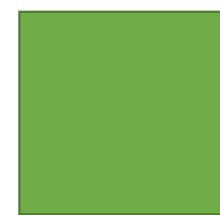
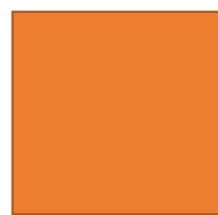
7 . 機能

モジュールが**1つの機能**のみを提供する



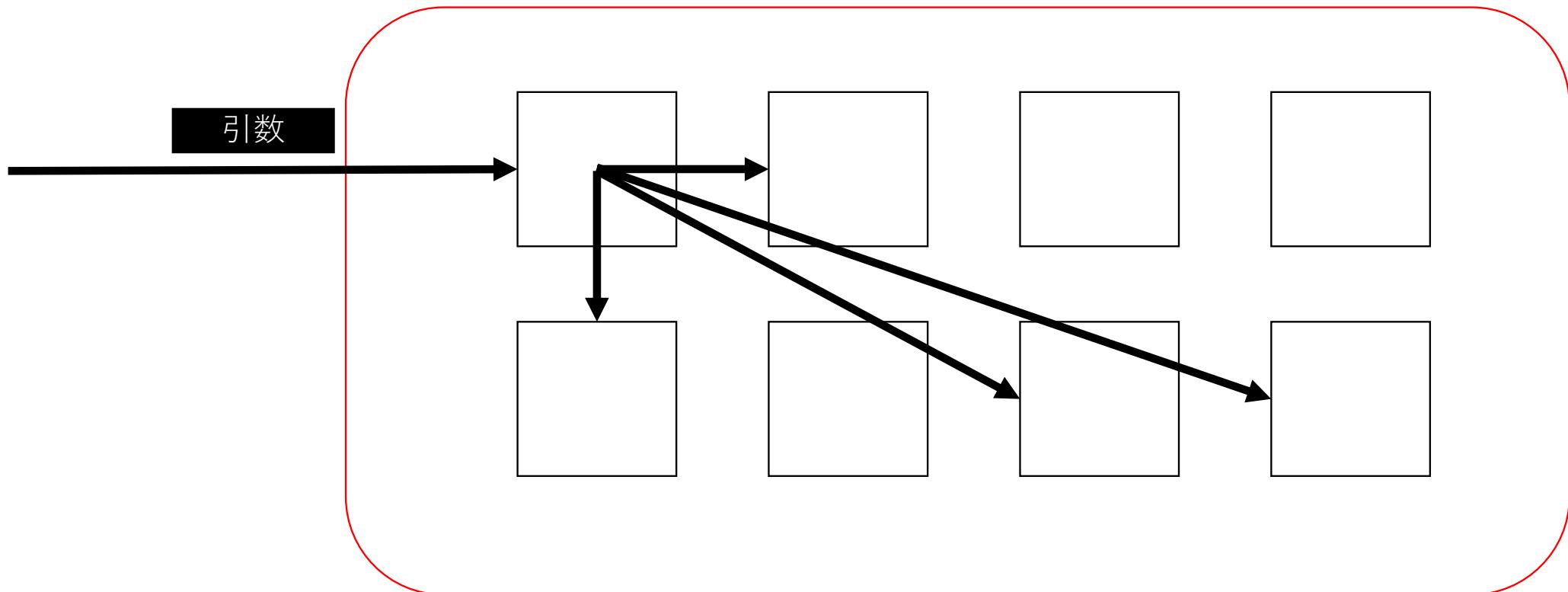
1. 暗号

関連を無視して複数の機能をまとめた。



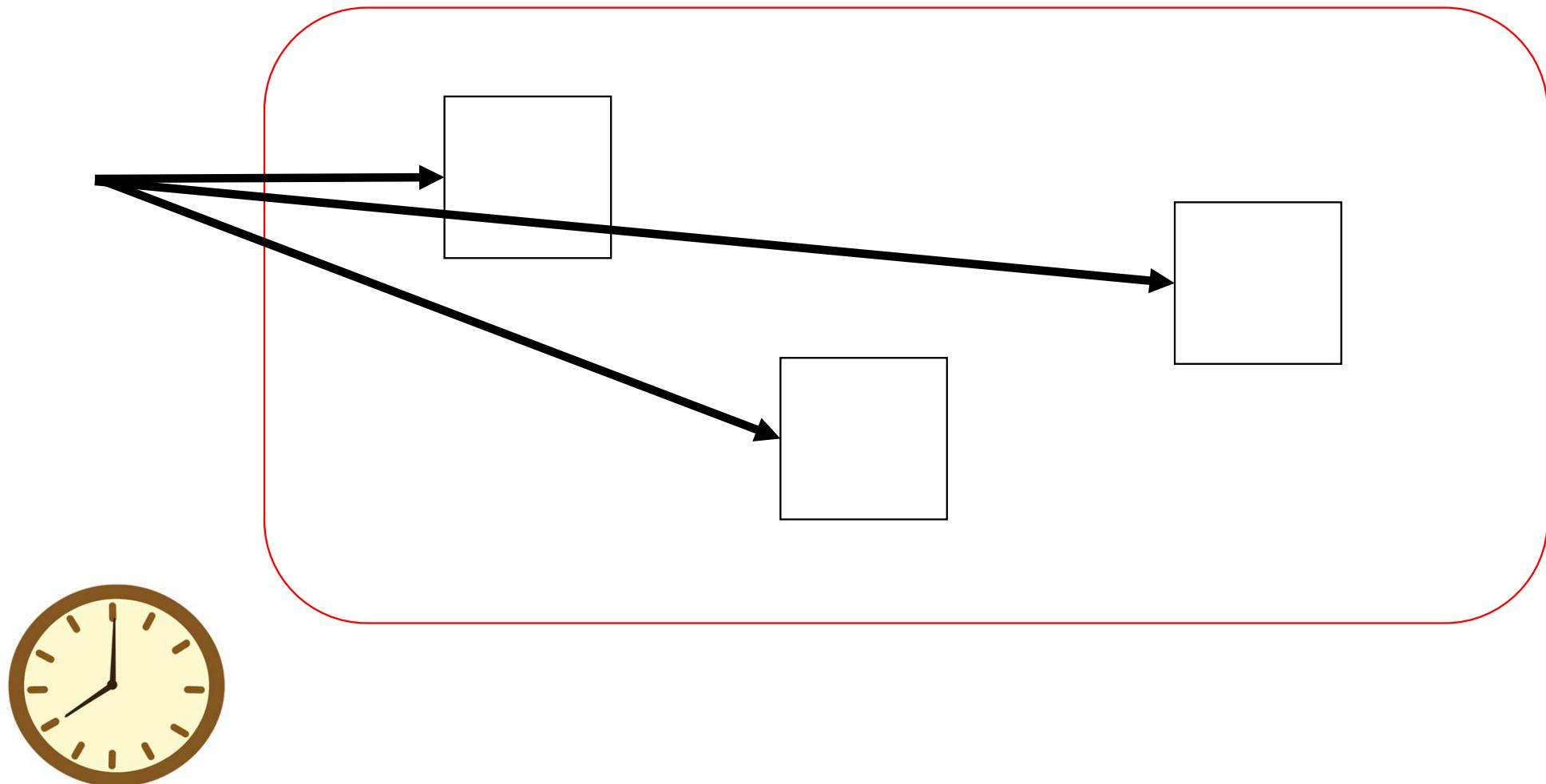
2.論理

関連した機能をまとめる。どれを使うか
は引数で指定する。



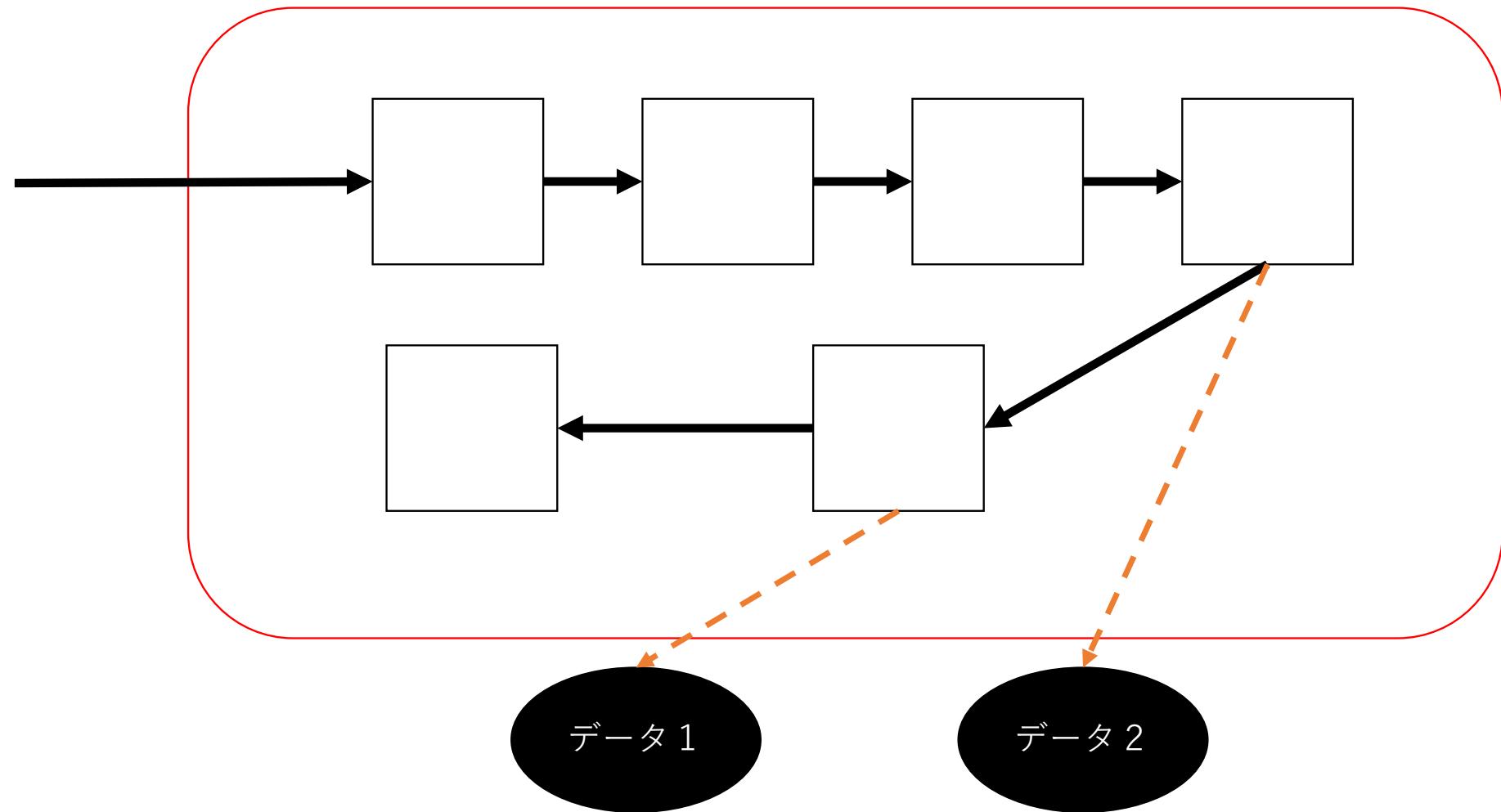
3. 時間

ある時点に連続して使用する機能をまとめた。



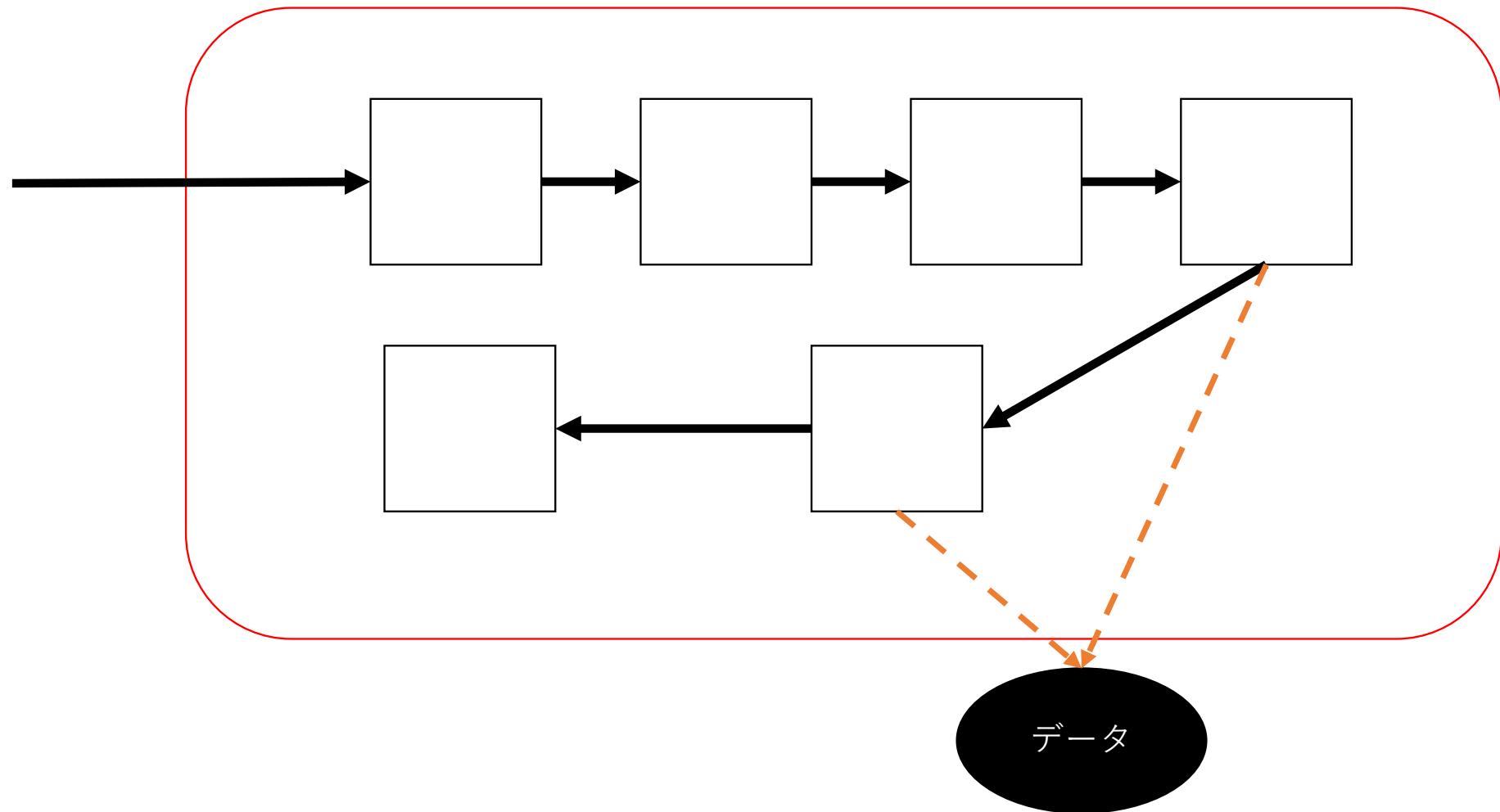
4.手順

順番に実行される機能をまとめた。
別のデータを使う



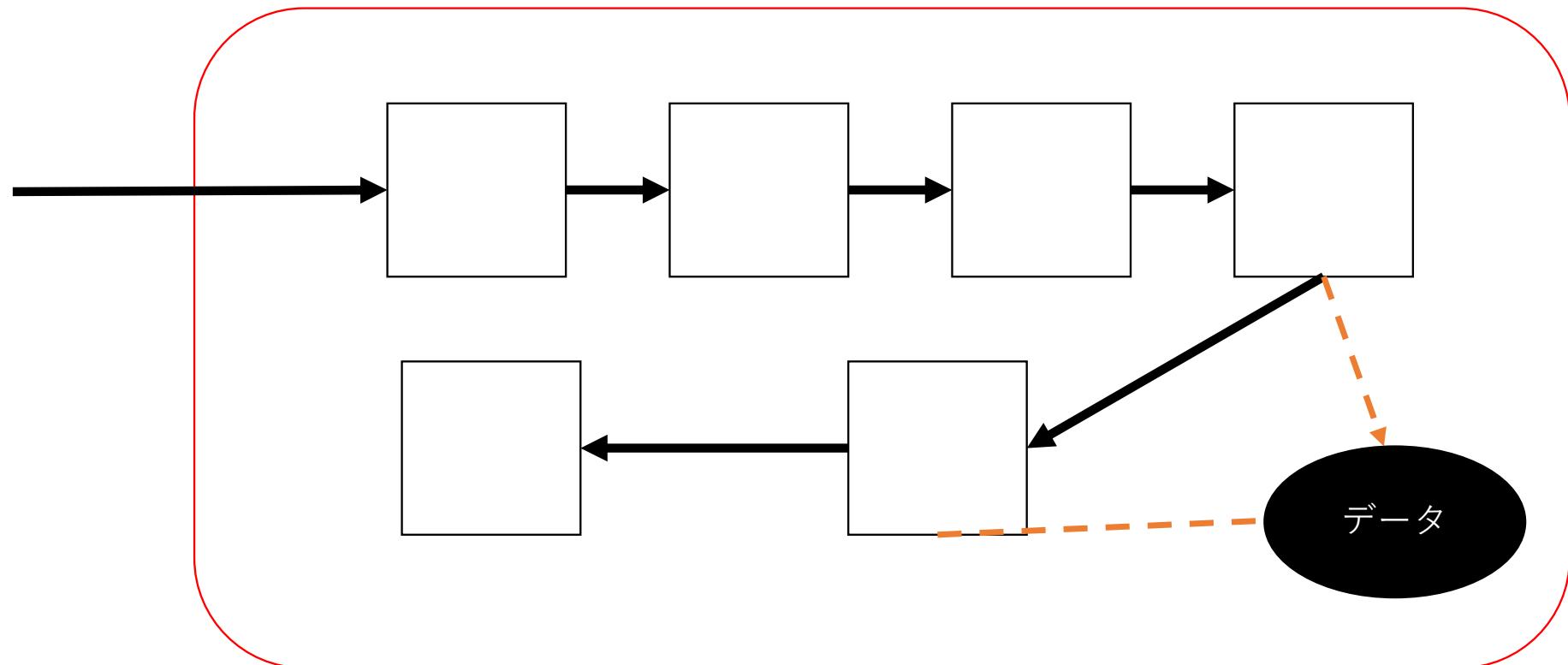
5. 連絡

順番に実行される機能をまとめた。
同じデータを使う。



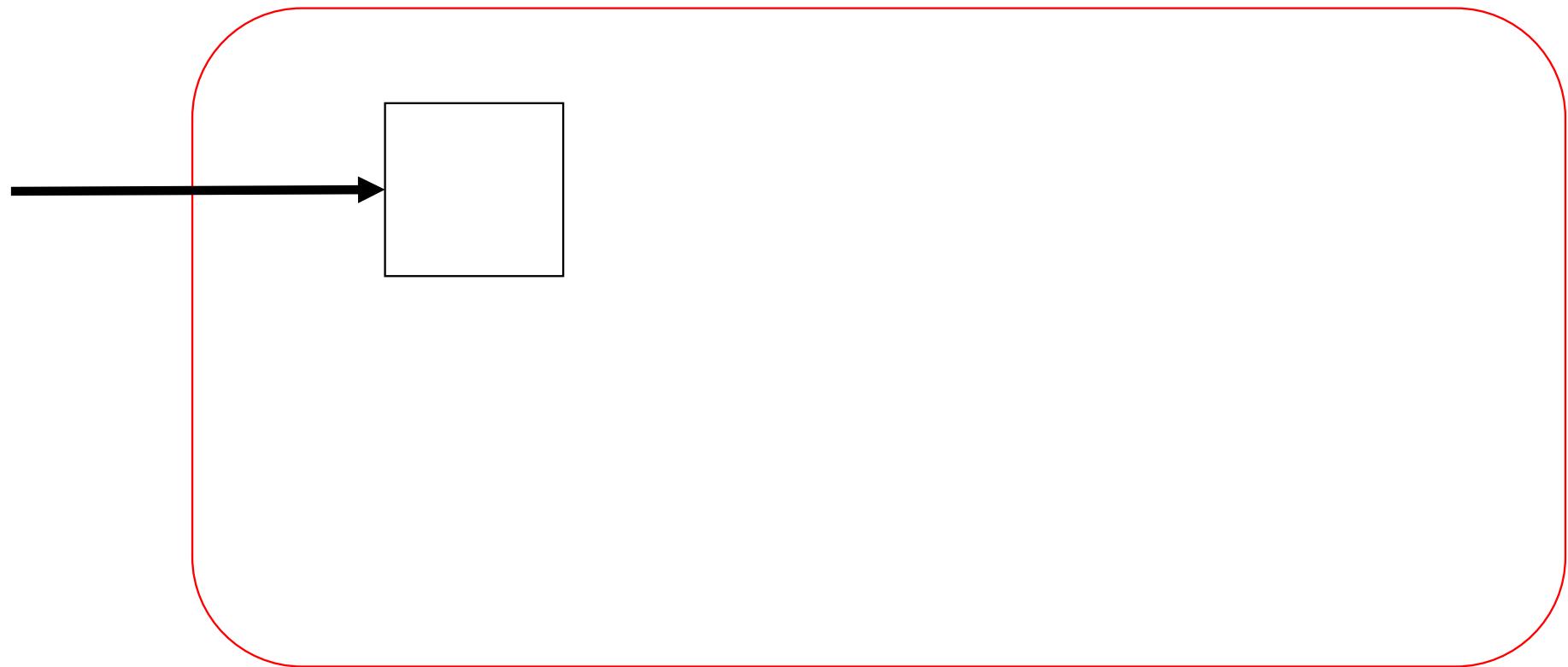
6.情報

同じデータを扱う機能を1つにまとめ
外部から隠した。



7. 機能

1つの機能だけが存在する。



機能の関連性に着目

1.暗号

機能間の**関連を無視し**複数の機能を1つのモジュールにまとめた。

2.論理

関連した、いくつかの機能を1つのモジュールにまとめ、どの機能を使うかを**引数で指定**する。

3.時間

ある**時点**に連續して使用する機能（関連がなくても）を1つのモジュールにまとめた。

順番に実行

モジュール強度

4.手順

必ず**順番（逐次）**に実行される機能を1つのモジュールにまとめた。別のデータを使う。

5.連絡

「手順的強度」に加え、**同じデータ**を使う。

6.情報

特定のデータを扱う機能を1つのモジュールにまとめ**隠ぺい**した。

7.機能

モジュールが**1つの機能**のみを提供する



問45 モジュールの結束性（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

問46 モジュール設計に関する記述のうち、モジュール強度（結束性）が最も強いものはどれか。

- ア ある木構造データを扱う機能をこのデータとともに一つにまとめ、木構造データをモジュールの外から見えないようにした。
- イ 複数の機能のそれぞれに必要な初期設定の操作が、ある時点で一括して実行できるので、一つのモジュールにまとめた。
- ウ 二つの機能 A, B のコードは重複する部分が多いので、A, B を一つのモジュールにまとめ、A, B の機能を使い分けるための引数を設けた。
- エ 二つの機能 A, B は必ず A, B の順番に実行され、しかも A で計算した結果を B で使うことがあるので、一つのモジュールにまとめた。

問45 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”的モジュールである。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”的モジュールである。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”的モジュールである。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”的モジュールである。関連した幾つかの機能を含み、パラメタによっていす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。

1. 内容

他のモジュールの内容を **直接参照** している。

4. 制御

他のモジュールを制御するために、**制御用のパラメータを引数** として受け渡している。

2. 共通

他のモジュールと **グローバル領域のデータ構造** を使ってつながっている。

5. スタンプ

データ構造を引数 として受け渡している。

3. 外部

他のモジュールと **グローバル領域のデータ** を使ってつながっている。

6. データ

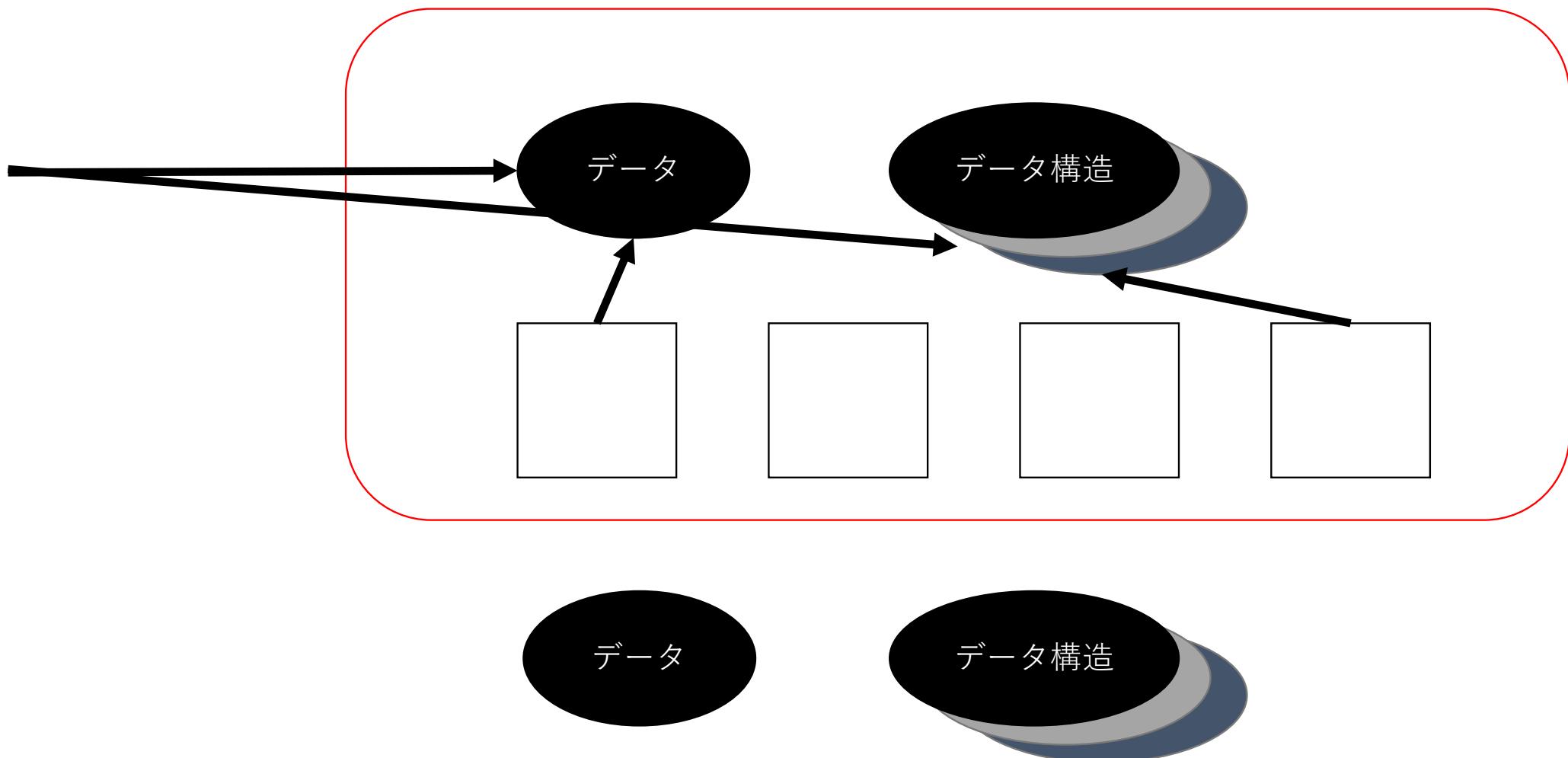
データを引数 として受け渡している。



1. 内容

モジュール内部の内容（データやデータ構造）
を直接参照

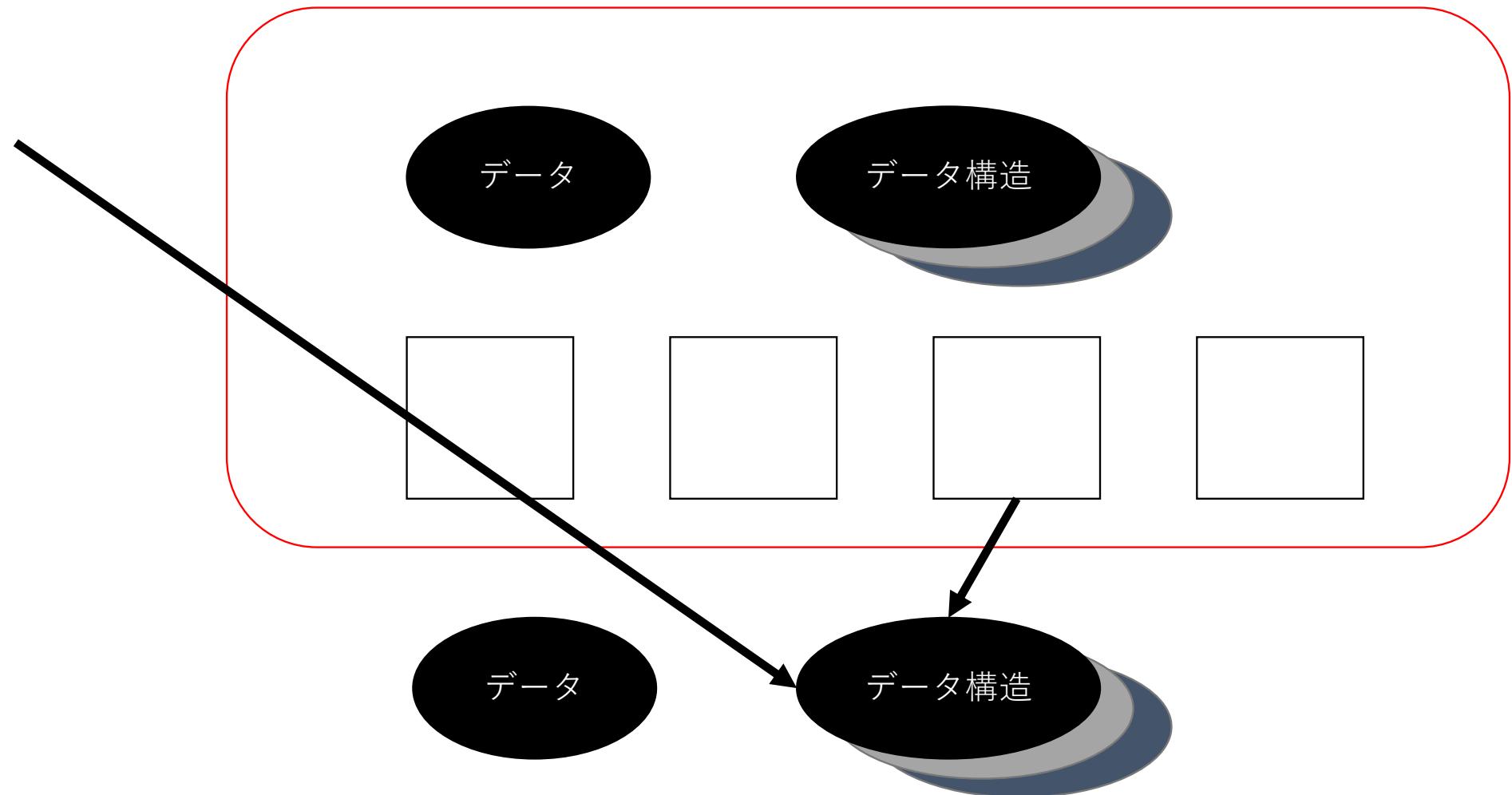
モジュール強度



2. 共通

グローバル領域のデータ構造を参照している。

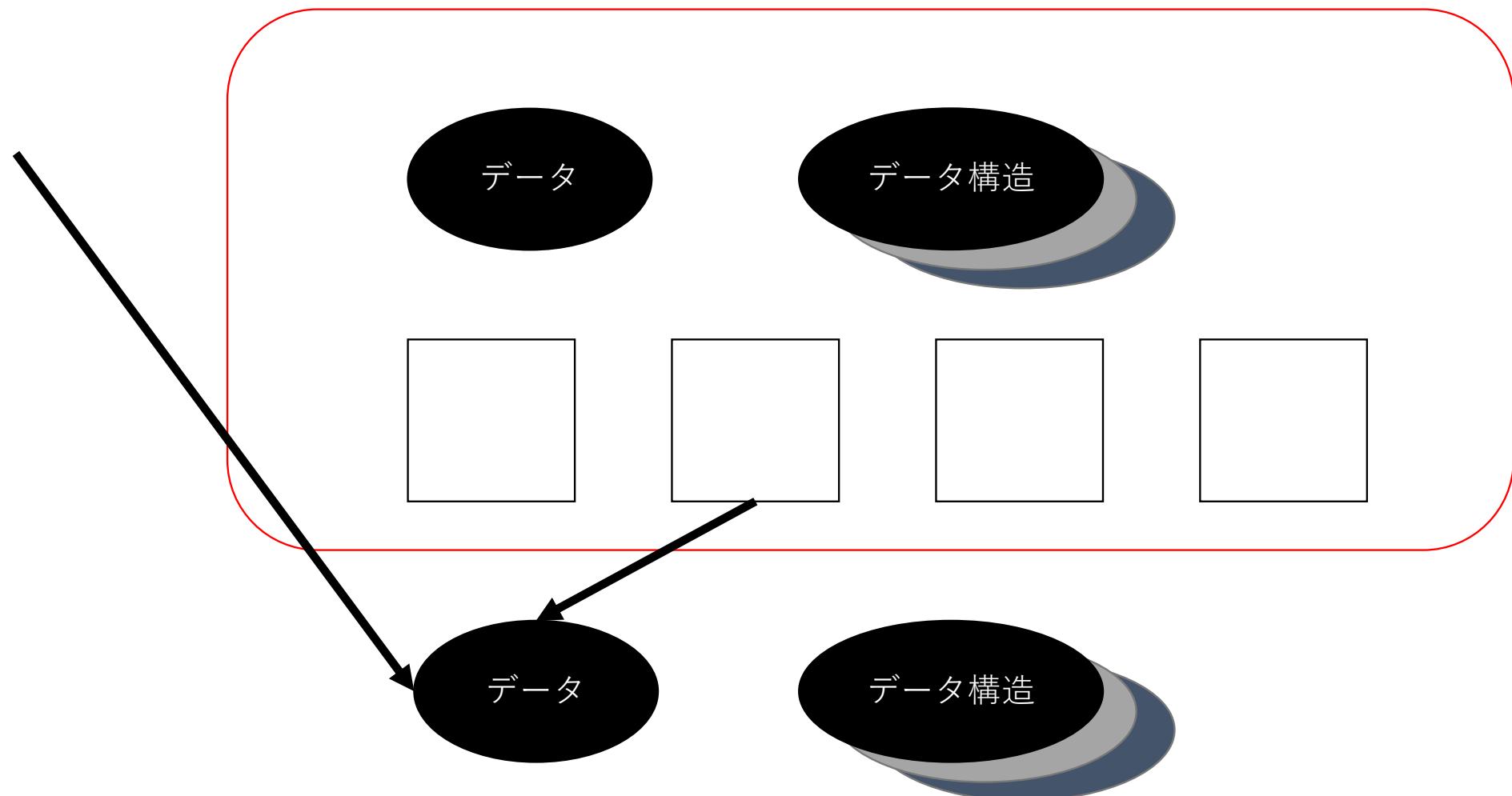
モジュール結合度



3.外部

グローバル領域のデータを参照している。

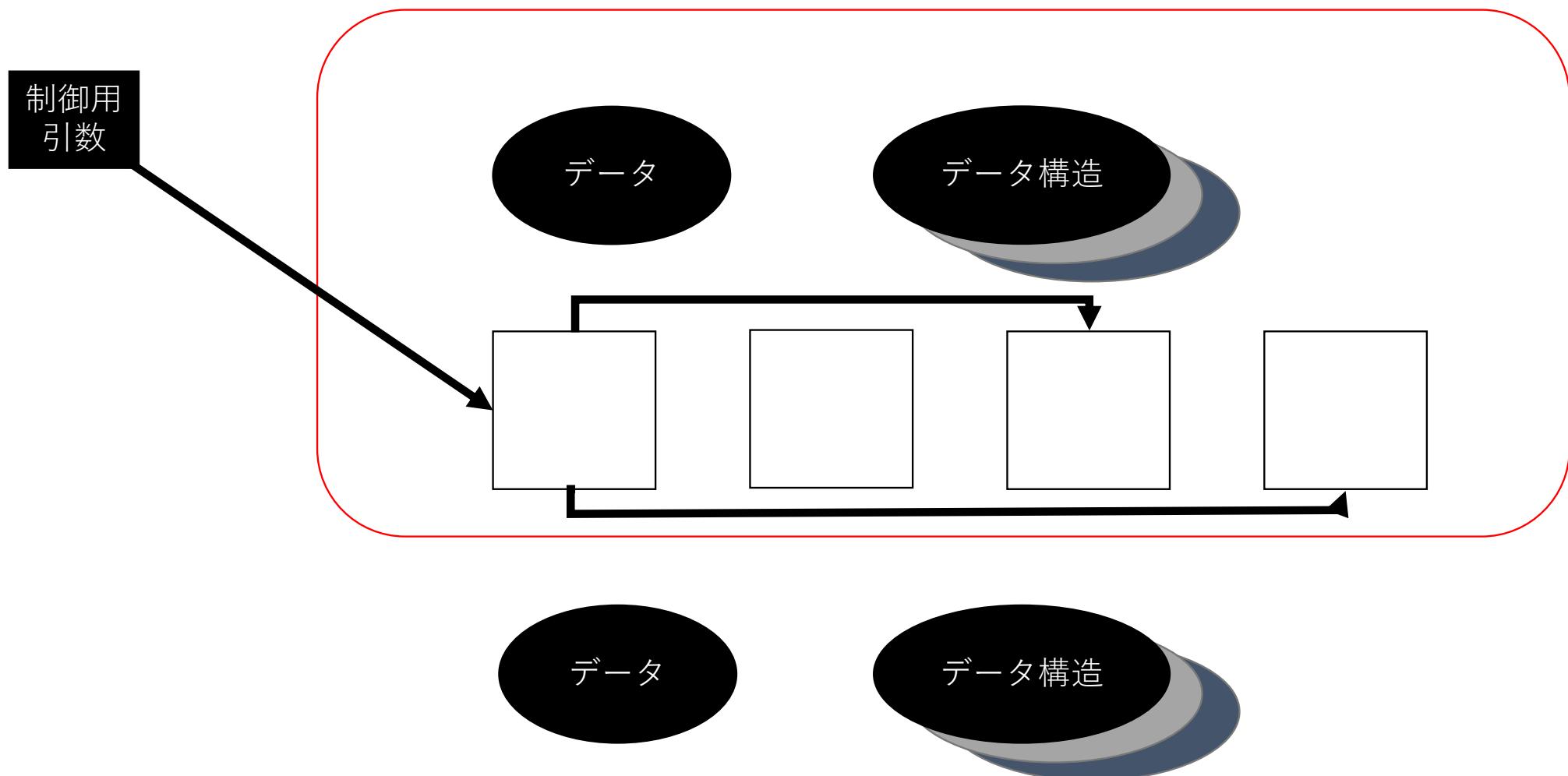
モジュール結合度



4. 制御

制御用パラメータを引数として渡し
実行順序を制御する。

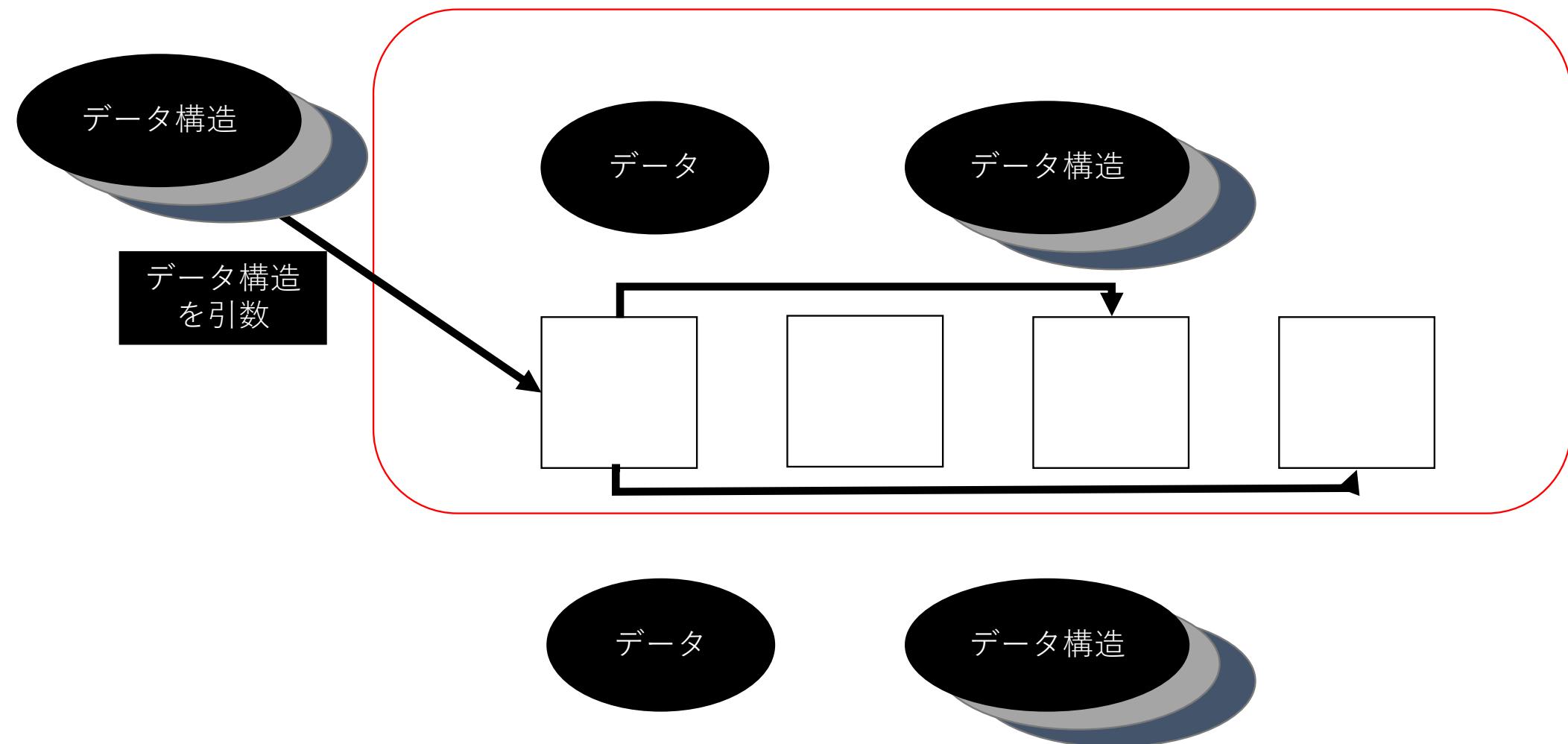
モジュール結合度



5.スタンプ

必要なデータ構造を引数として渡す。

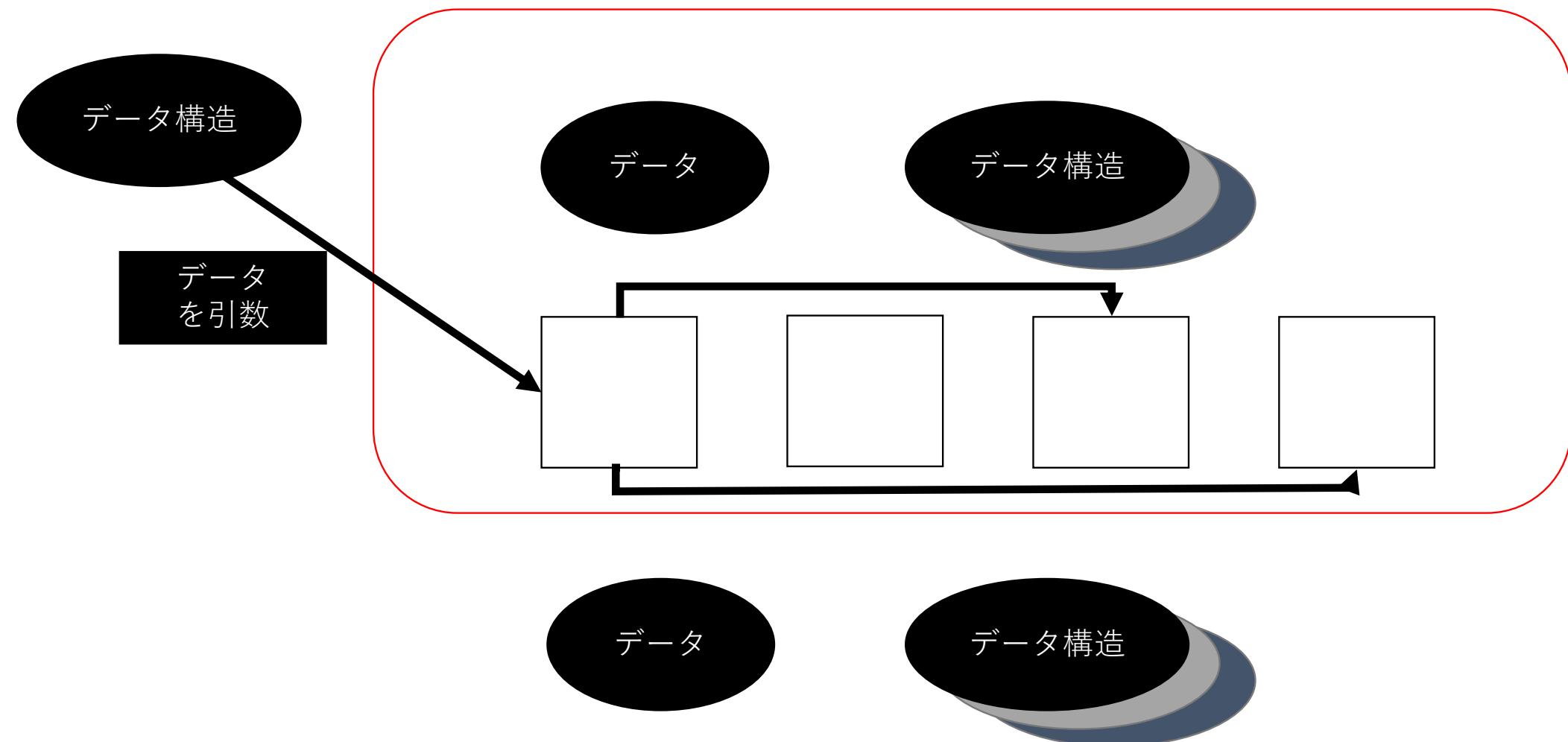
モジュール結合度



6. データ

必要なデータ構造を引数として渡す。

モジュール結合度



1. 内容

他のモジュールの内容を**直接参照**している。

4. 制御

他のモジュールを制御するために、**制御用のパラメータを引数**として受け渡している。

2. 共通

他のモジュールと**グローバル領域のデータ構造**を使ってつながっている。

3. 外部

他のモジュールと**グローバル領域のデータ**を使ってつながっている。

グローバル領域を使用

5. スタンプ

データ構造を引数として受け渡している。

6. データ

データを引数として受け渡している。

引数として渡す



問48 モジュール間の情報の受渡しがパラメタだけで行われる、結合度が最も弱いモジュール結合はどれか。

- ア 共通結合 イ 制御結合 ウ データ結合 エ 内容結合

問47 モジュールの結合度が最も低い、データの受渡し方法はどれか。

- ア 単一のデータ項目を大域的データで受け渡す。
- イ 単一のデータ項目を引数で受け渡す。
- ウ データ構造を大域的データで受け渡す。
- エ データ構造を引数で受け渡す。

問46 モジュール分割の良否を、モジュール結合度の視点から評価する場合、最も適切な記述はどれか。

- ア 共通データ領域は、全てのモジュールからアクセスできるようになっていることが望ましい。
- イ ソフトウェア全体のモジュール分割の良否は、モジュール間の結合度のうちで最も強いものがどのように分布しているかで判断するのが望ましい。
- ウ 直接の呼び出し関係になっていないモジュール間で情報を交換するには、共通データ領域を用いるのが最も望ましい。
- エ 呼び出す側と呼び出される側のモジュール間のデータの受渡しは、引数としてデータ項目を列挙するのが最も望ましい。

要件定義

外部設計

内部設計

詳細設計

コーディング

上流工程

下流工程

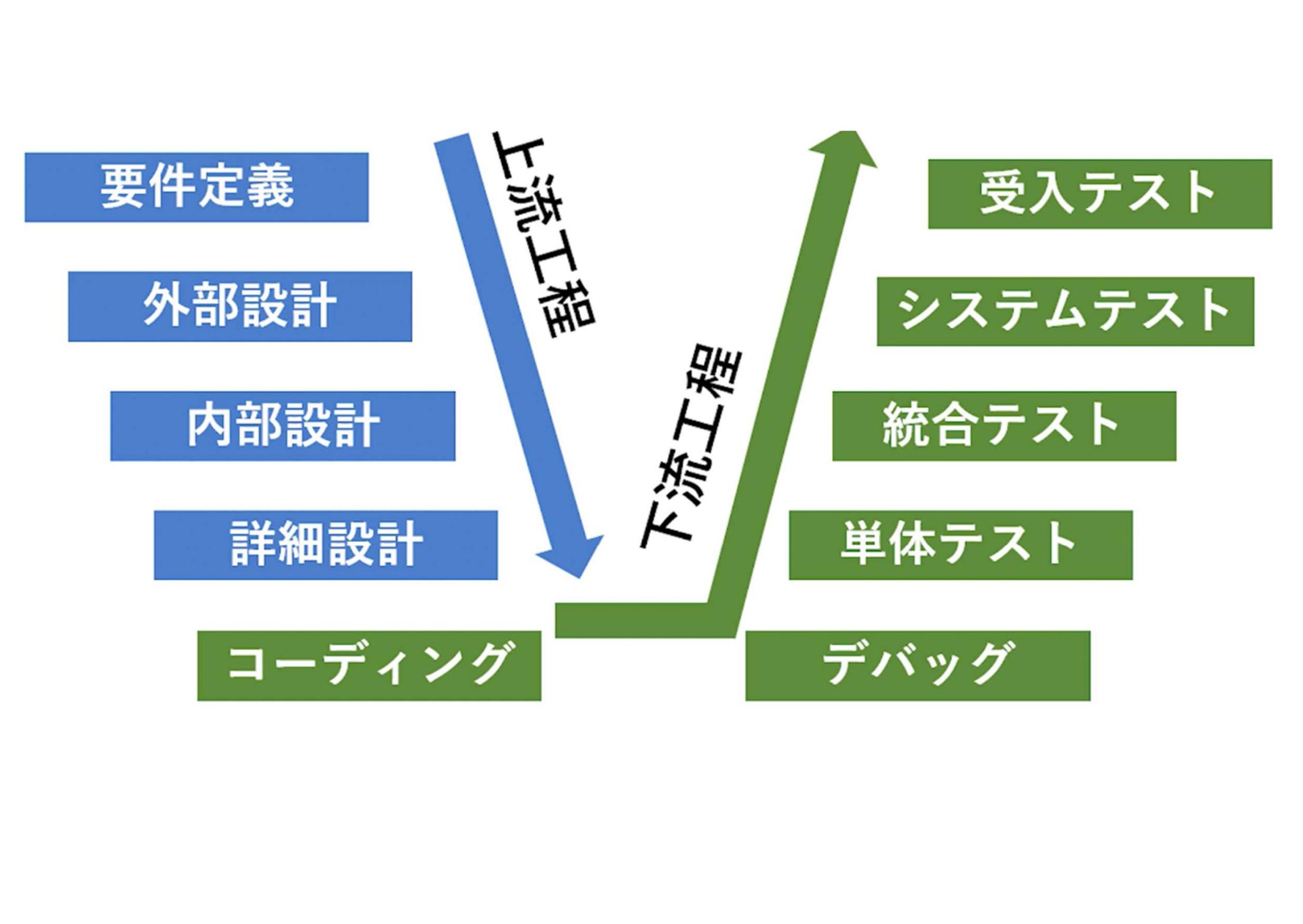
受入テスト

システムテスト

統合テスト

単体テスト

デバッグ



ブラックボックステスト

単体テスト

内部の動きは確認せず、入力に対して正しい結果が返ってくるかをテストする

2 1

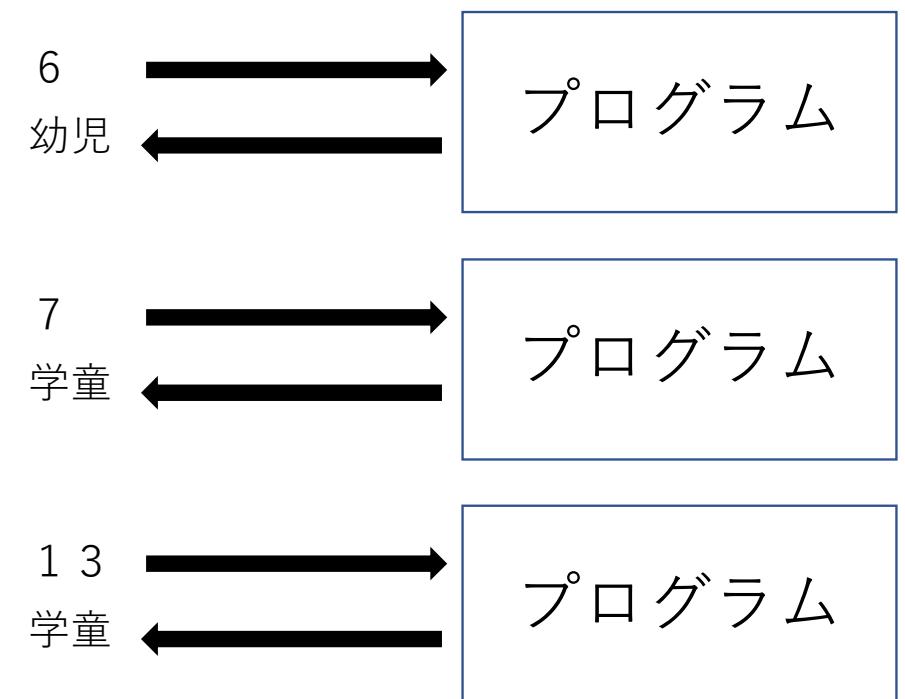
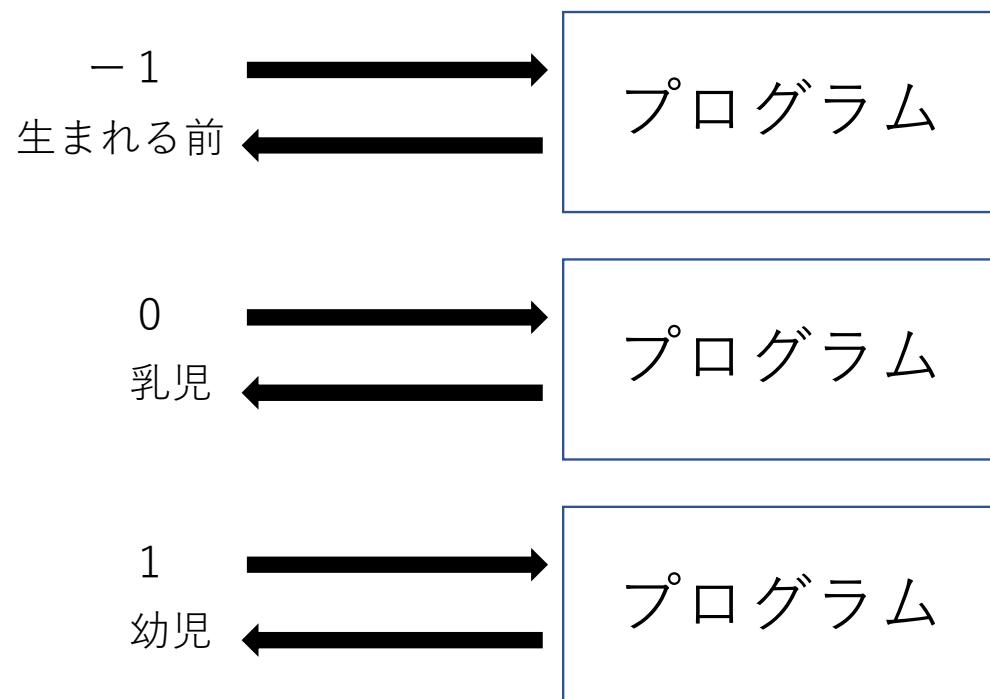


青年

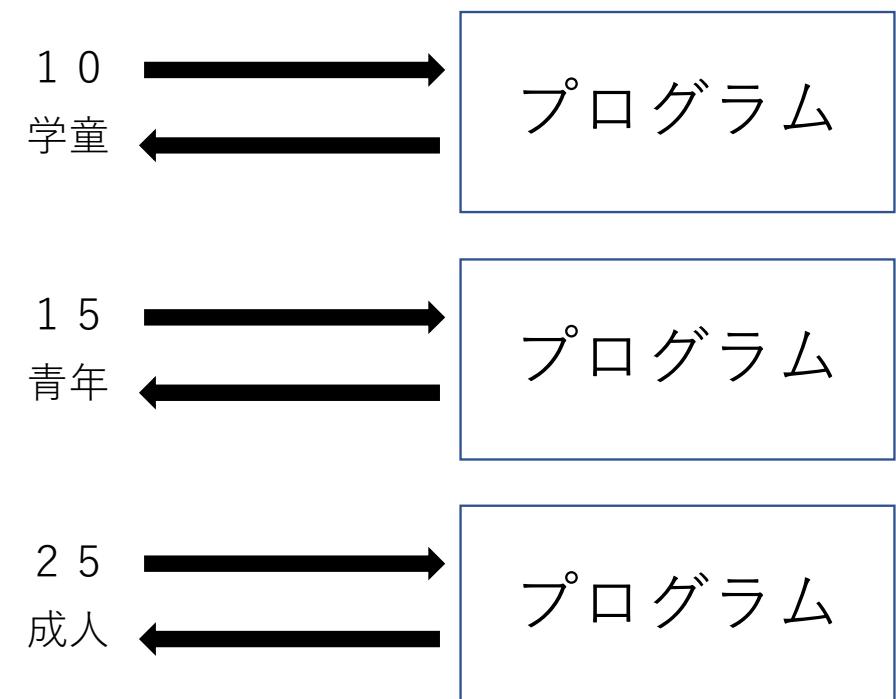
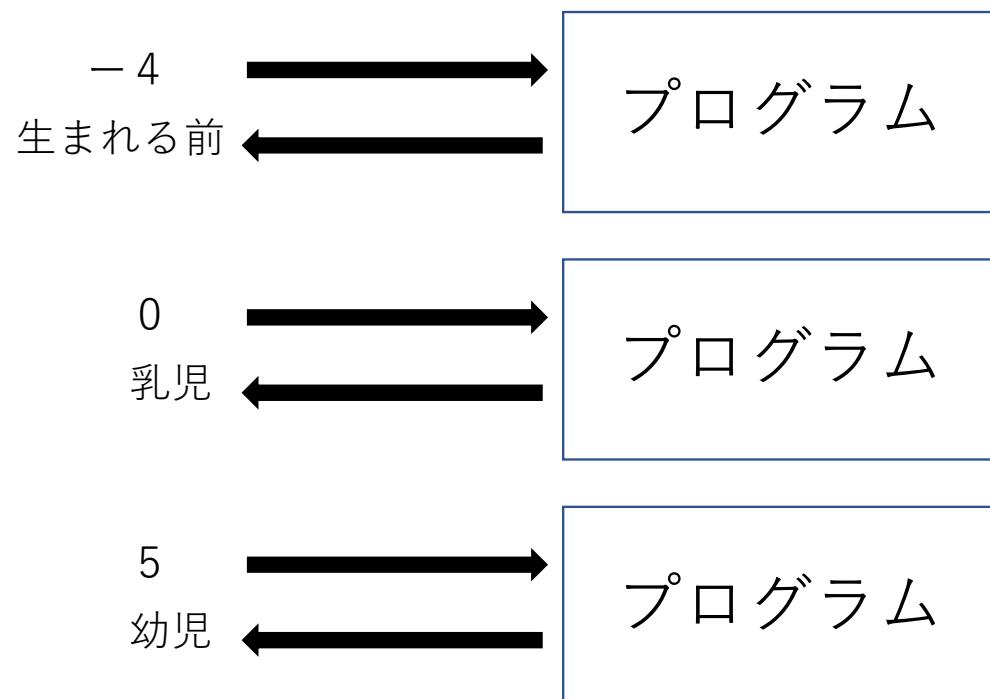


プログラム

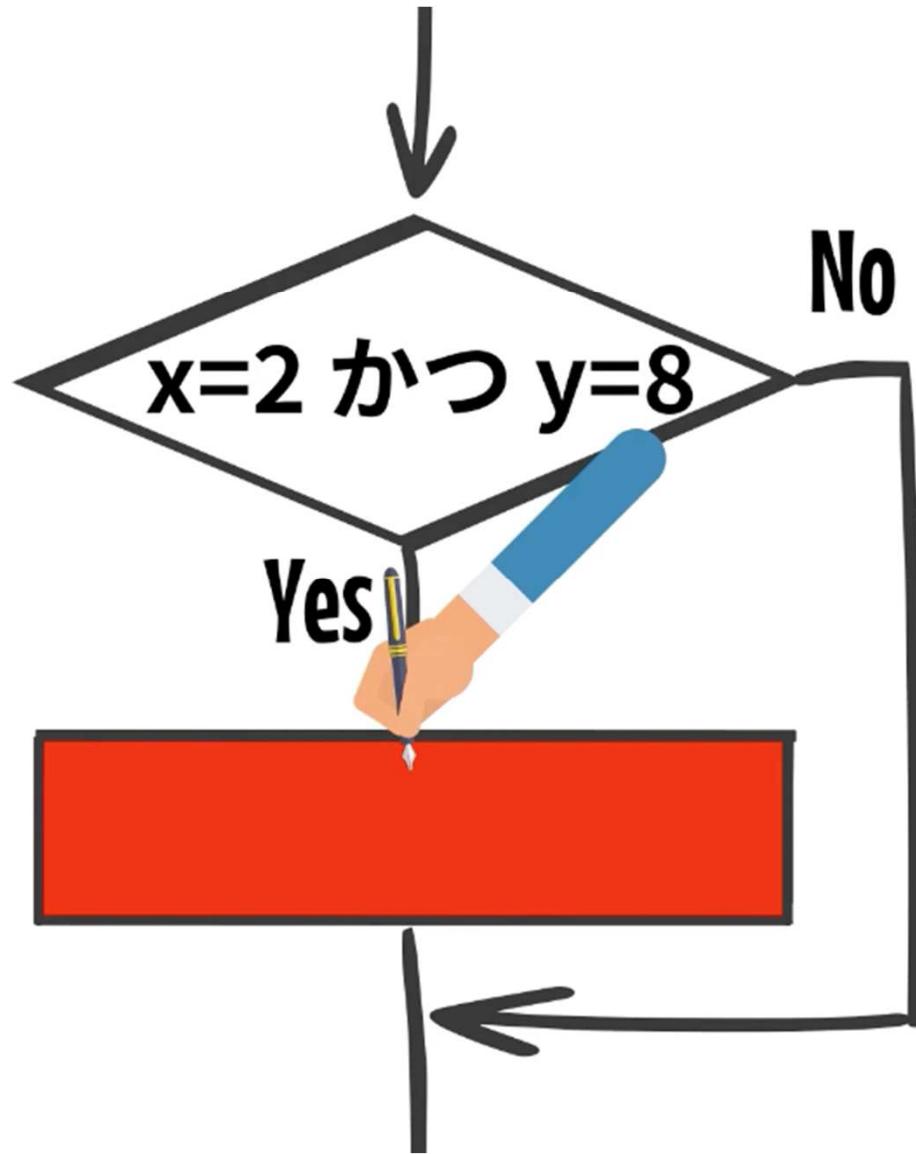
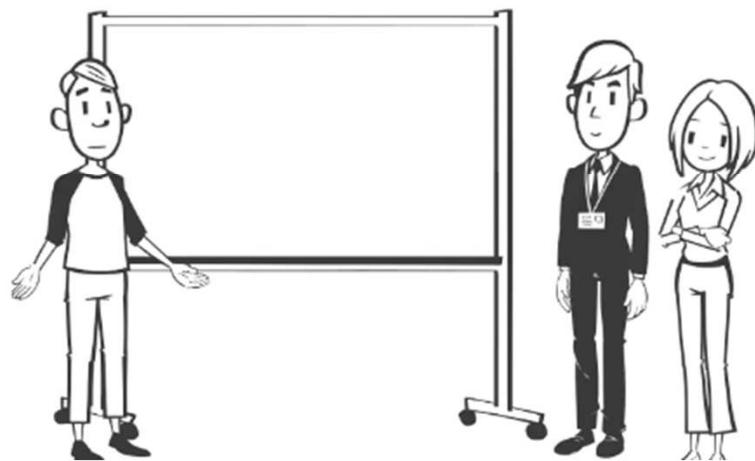
グループ分けし、境目のデータを重点的にテストする



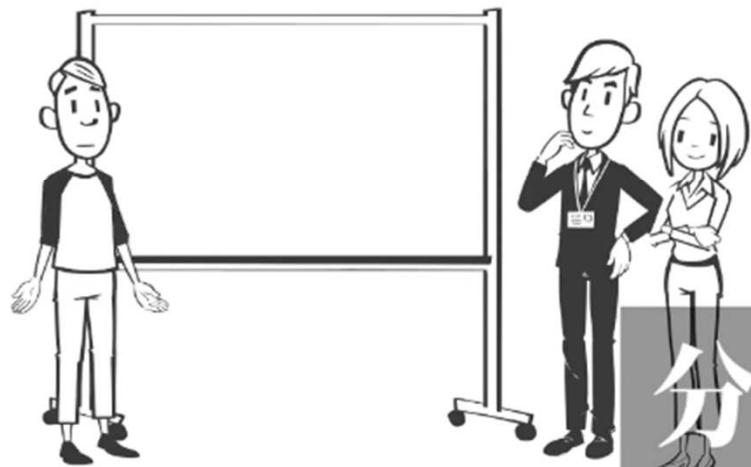
グループ分けし、それぞれから代表値を選ぶ



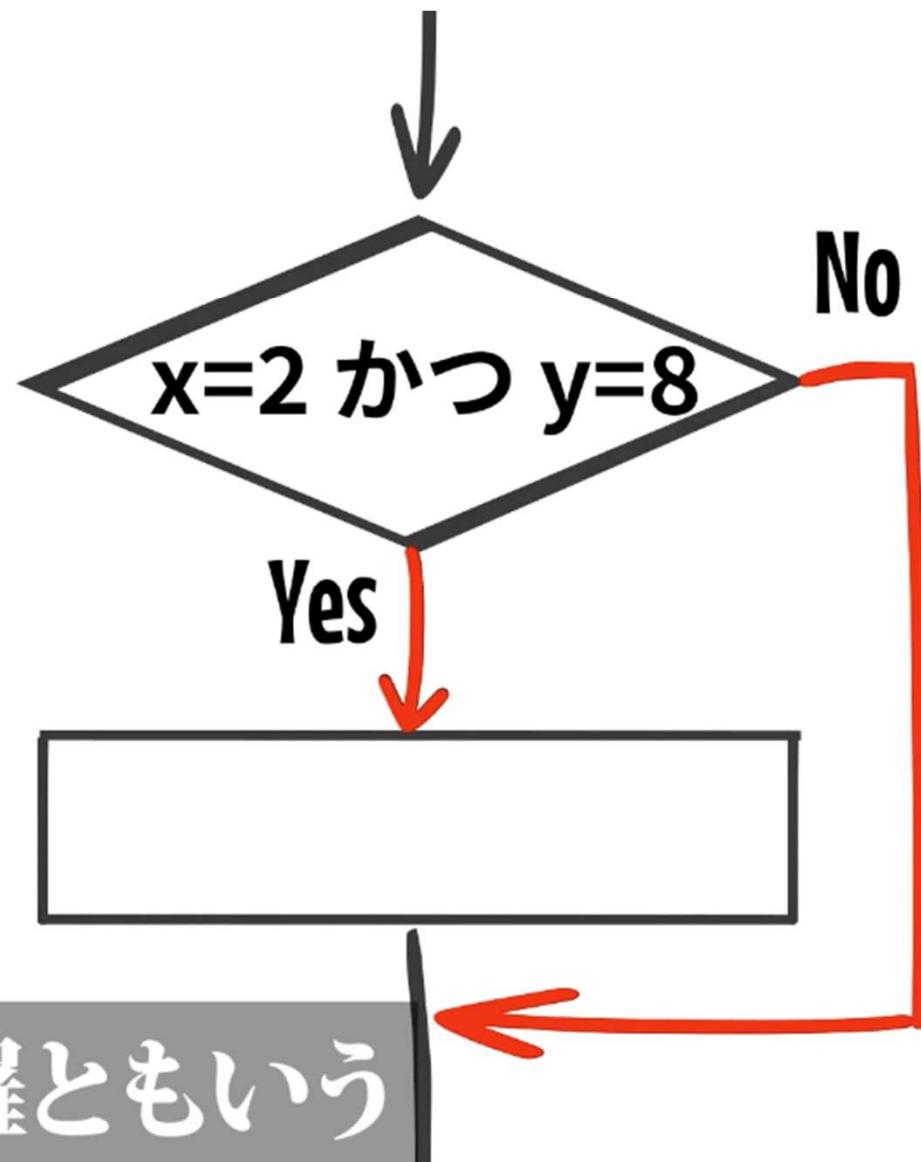
今回は
命令網羅
でお願ひします



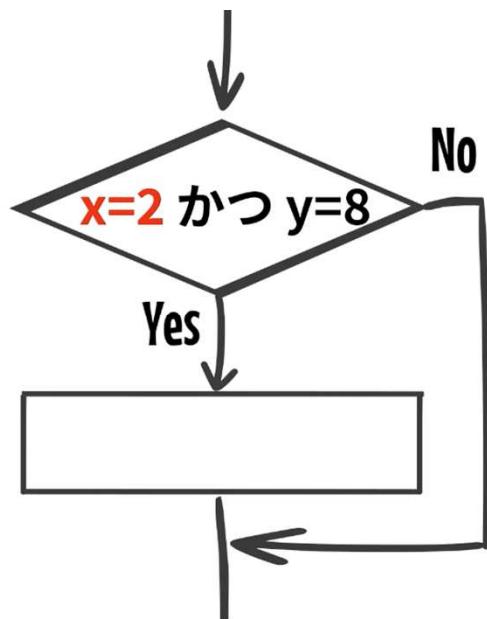
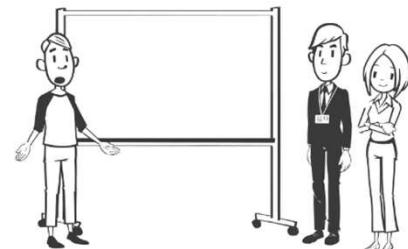
今回は
判定条件網羅
でお願いします



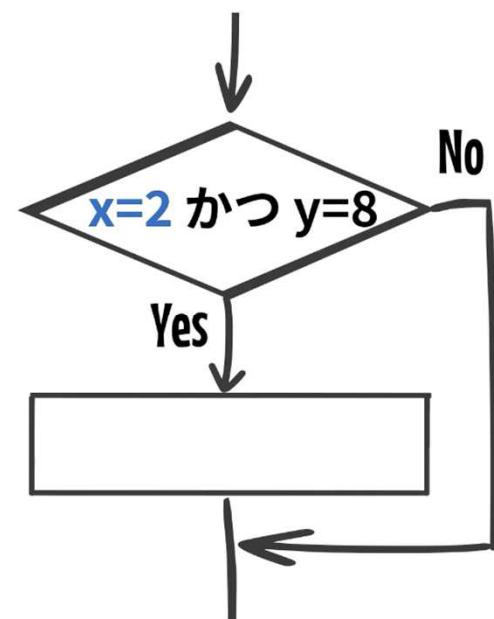
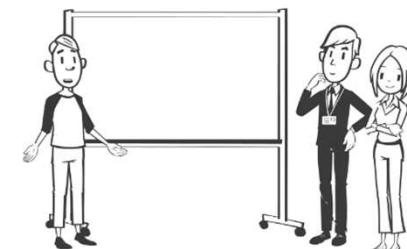
分岐網羅ともいう



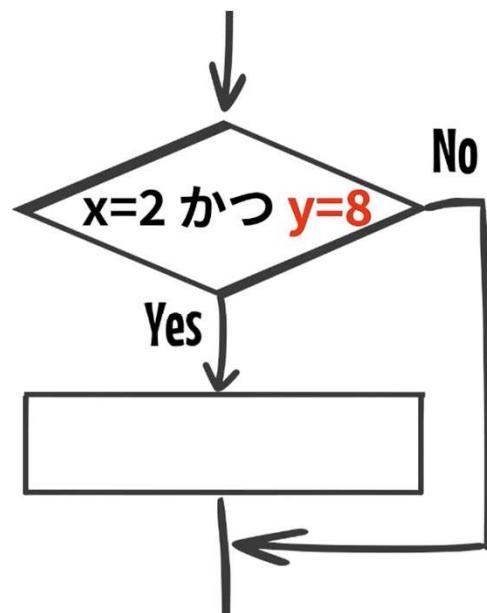
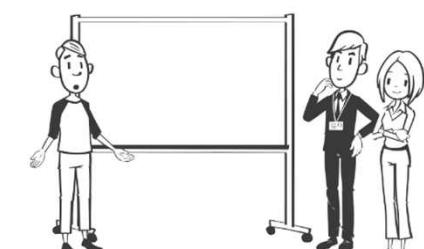
今回は
条件網羅
でお願いします



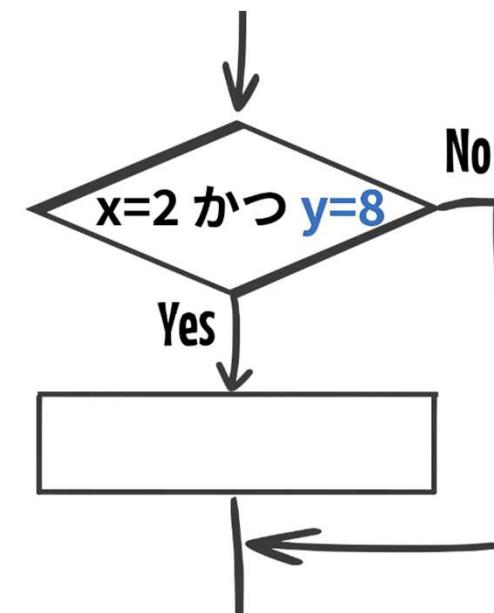
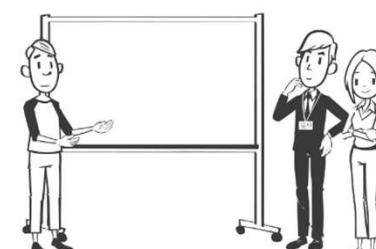
今回は
条件網羅
でお願いします



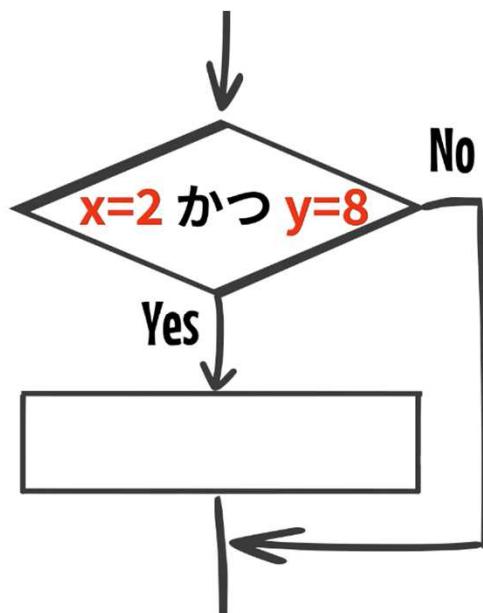
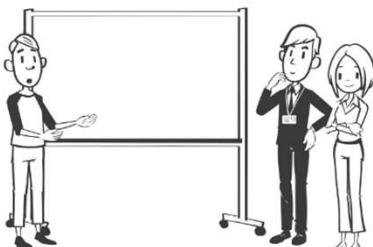
今回は
条件網羅
でお願いします



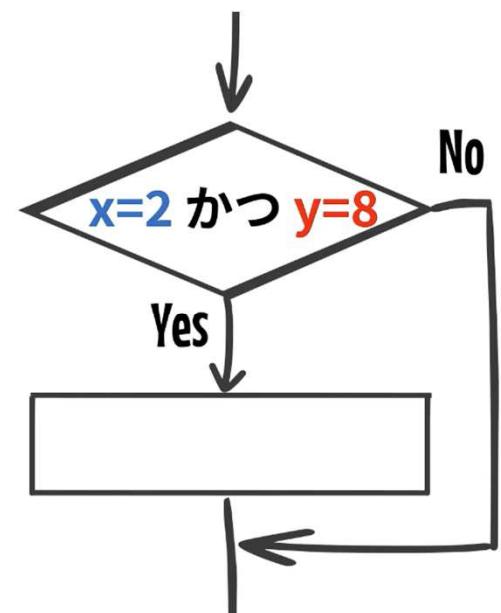
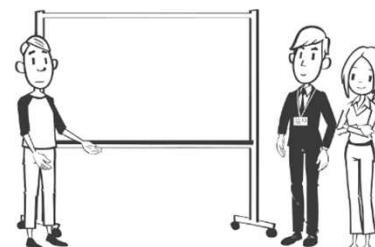
今回は
条件網羅
でお願いします



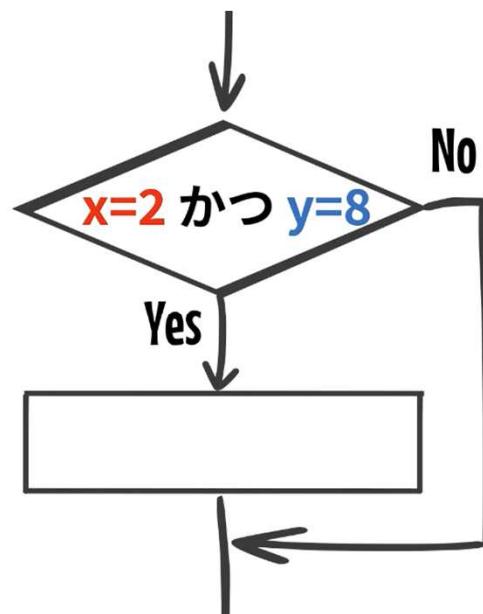
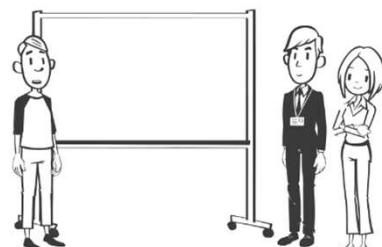
今回は
複数条件網羅
でお願いします



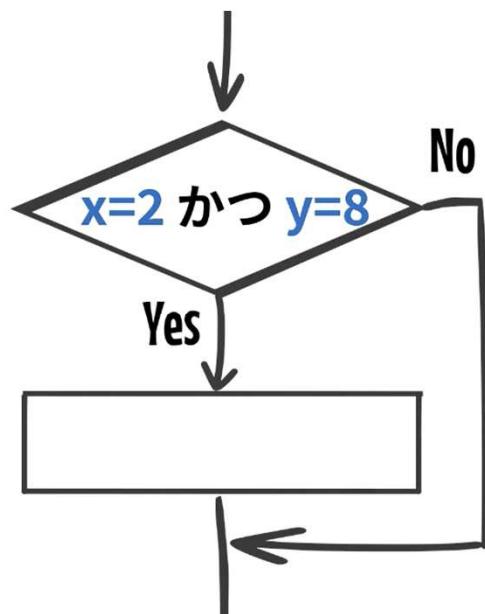
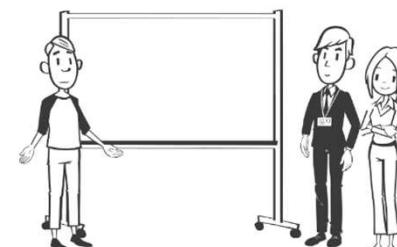
今回は
複数条件網羅
でお願いします



今回は
複数条件網羅
でお願いします



今回は
複数条件網羅
でお願いします



問18 ホワイトボックステストにおいて、プログラムの実行された部分の割合を測定するのに使うものはどれか。

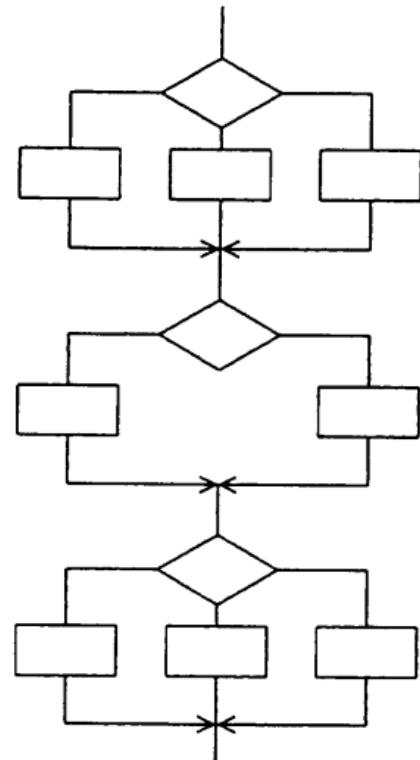
ア アサーションチェッカ

ウ 静的コード解析ツール

イ シミュレータ

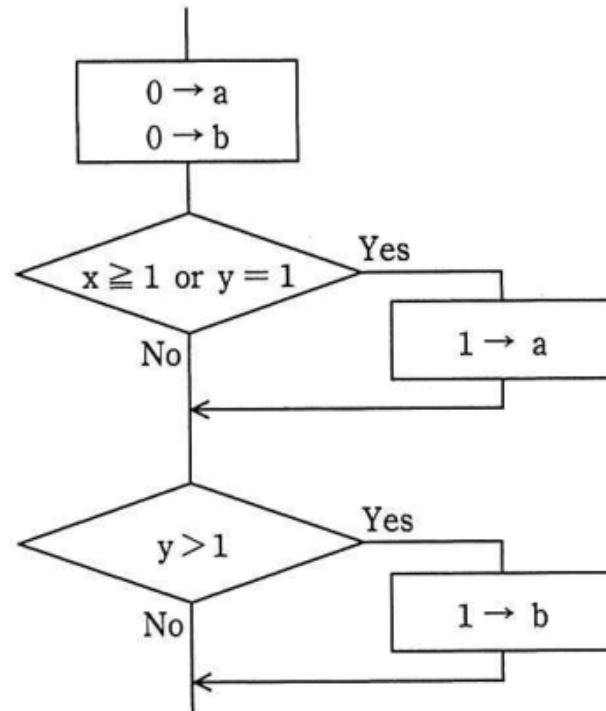
エ テストカバレージ分析ツール

問44 あるプログラムについて、流れ図で示される部分に関するテストを、命令網羅で実施する。この場合、考えられる最小のテストケース数はどれか。ここで、それぞれの判定条件は互いに独立であるとする。



ソフトウェア開発技術者平成18年秋期 午前問44

問49 流れ図において、分岐網羅を満たし、かつ、条件網羅を満たすテストデータの組は
どれか。



入力（テストデータ）		
	x	y
ア	2	2
	1	2
イ	1	2
	0	0
ウ	1	2
	1	1
エ	0	1
	1	2
オ	0	1
	0	2

問48 ブラックボックステストのテストデータの作成方法のうち、最も適切なものはどれか。

- ア 稼働中のシステムから実データを無作為に抽出し、テストデータを作成する。
- イ 機能仕様から同値クラスや限界値を識別し、テストデータを作成する。
- ウ 業務で発生するデータの発生頻度を分析し、テストデータを作成する。
- エ プログラムの流れ図から、分岐条件に基づいたテストデータを作成する。

問49 プログラムの誤りの一つに、繰返し処理の終了条件として $A \geq a$ とすべきところを $A > a$ とコーディングしたことに起因するものがある。このような誤りを見つけるために有効なテストケース設計技法はどれか。ここで、A は変数、a は定数とする。

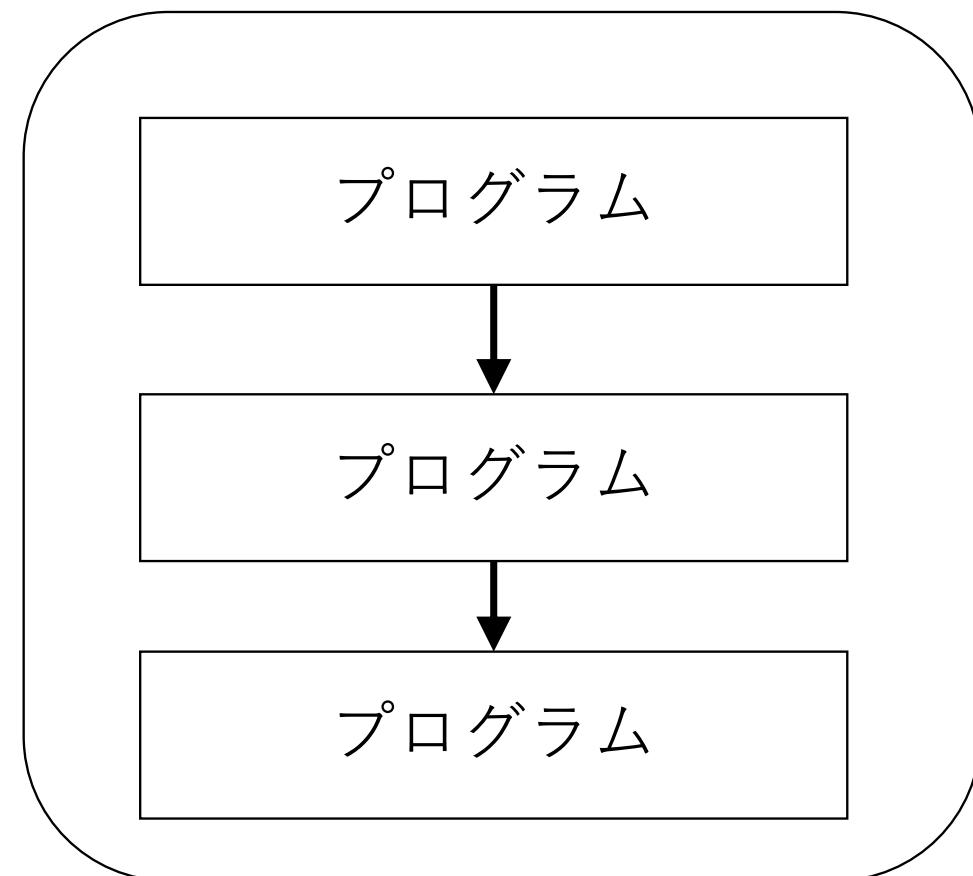
ア 限界値分析

イ 条件網羅

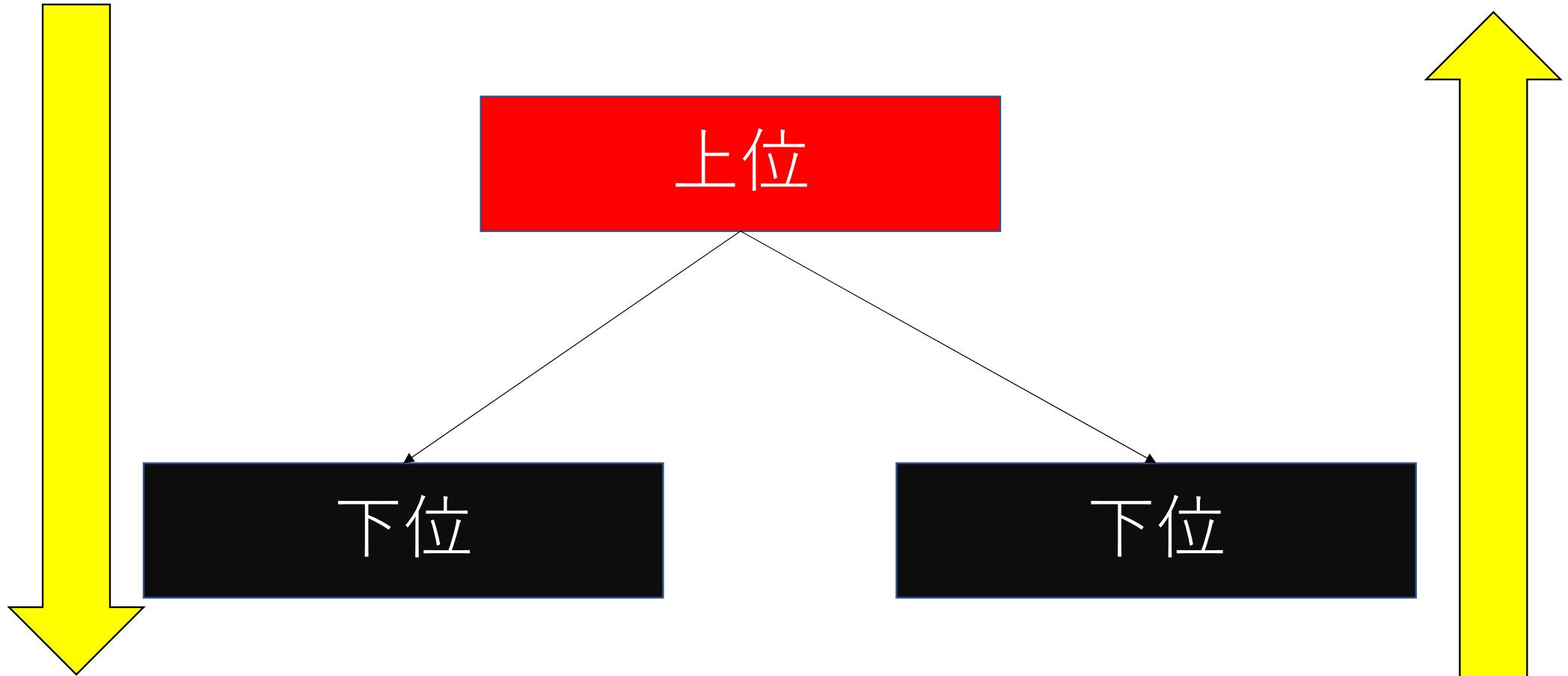
ウ 同値分割

エ 分岐網羅

ある程度まとめた単位でのテスト

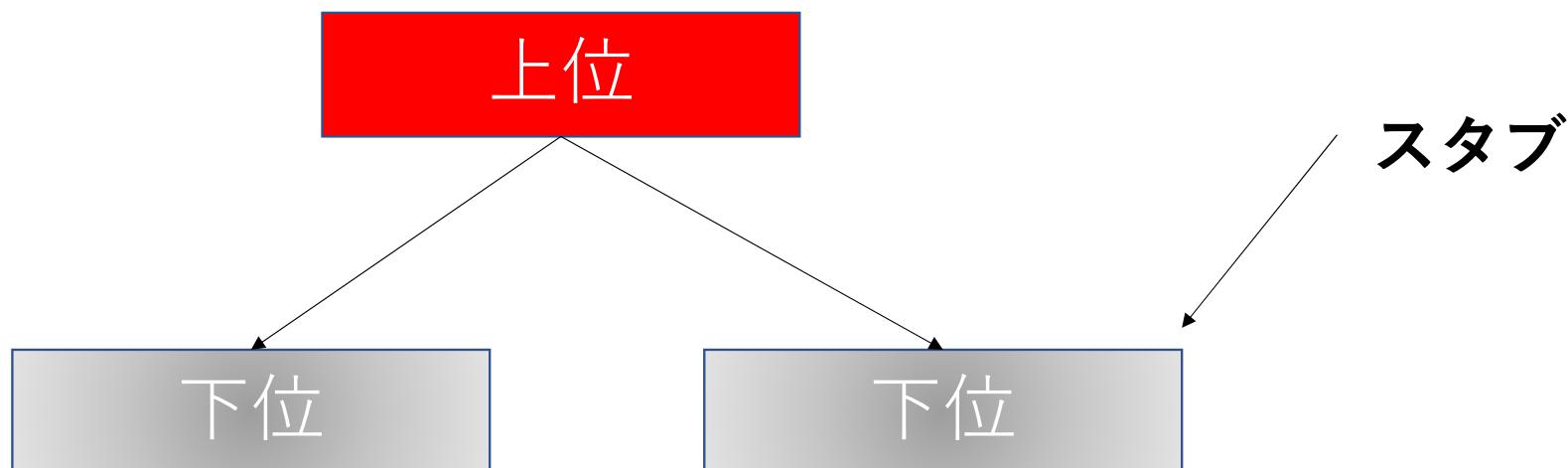
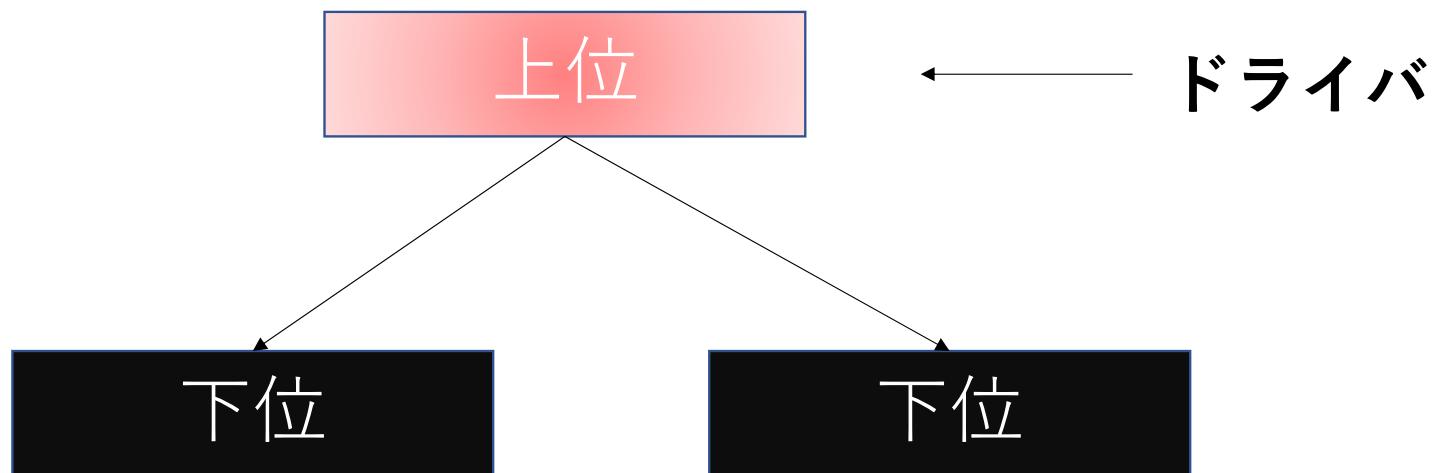


結合テスト

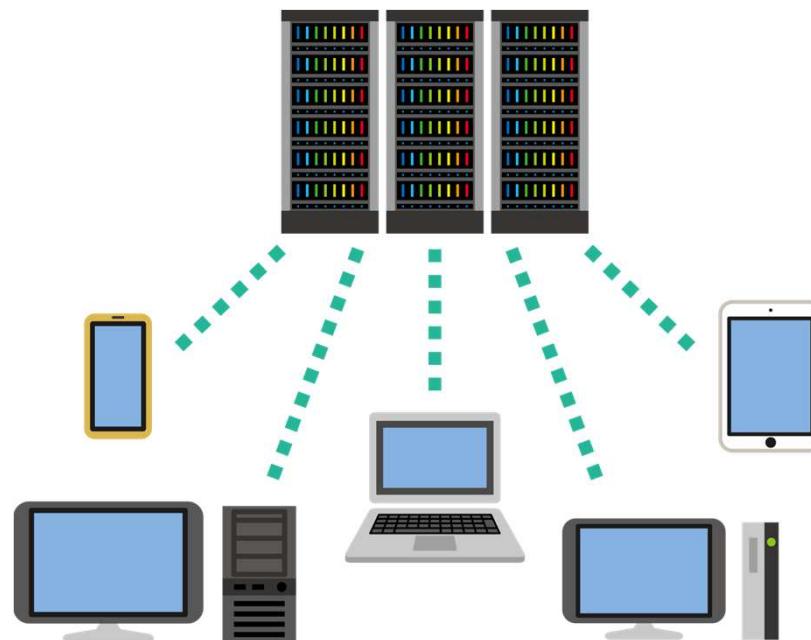


トップダウンテスト

ボトムアップテスト



開発の最終段階で、システム全体を検証する



問45 テスト工程におけるスタブの利用方法に関する記述として、適切なものはどれか。

- ア 指定した命令が実行されるたびに、レジスタや主記憶の一部の内容を出力することによって、正しく処理が行われていることを確認する。
- イ トップダウンでプログラムのテストを行うとき、作成したモジュールをテストするため、仮の下位モジュールを用意して動作を確認する。
- ウ プログラムの実行中、必要に応じて変数やレジスタなどの内容を表示し、必要であればその内容を修正して、テストを継続する。
- エ プログラムを構成するモジュールの単体テストを行うとき、そのモジュールを呼び出す仮の上位モジュールを用意して、動作を確認する。

問57 システム開発委託先（受託者）から委託元（委託者）に納品される成果物に対するユーザ受入テストの適切性を確かめるためのシステム監査の要点はどれか。

- ア 委託者が作成したユーザ受入テスト計画書に従って、受託者が成果物に対してユーザ受入テストを実施していること
- イ 受託者が成果物と一緒にユーザ受入テスト計画書を納品していること
- ウ 受託者から納品された成果物に対して、委託者が要件定義に基づきユーザ受入テストを実施していること
- エ 受託者から納品された成果物に対して、監査人がユーザ受入テスト計画を策定していること