Chapter12

システム開発技術

情報処理技術者試験センターのIPAから提唱されている「基本情報技術者シラバス」では、開発プロセスの各工程は以下の名称で定めており、ユーザサイドと開発側との共通の枠組み、認識を標準化することで、開発時の表現の違いやトラブルを防止することを目的としています。

Chapter12は、シラバスに準拠して解説します。

一方で、従来から用いられているウォータフォールモデルという開発手法をベースとした呼称も存在します。ウォータフォールモデルについては、Chapter13で解説します。

《 IPA提唱の開発プロセス 》 《 従来の開発プロセス 》

（対応）

システム要件定義

システム方式の設計

ソフトウェア要件定義

ソフトウェア方式設計

ソフトウェア詳細設計

ソフトウェア構築

テスト

移行

基本計画

外部設計

内部設計

プログラミング設計

プログラミング

テスト

移行

運用・保守

運用・保守

# 1. システム要件定義

## 1. システム要件の定義

**システム要件定義**では、システム化の目標と範囲、システムの機能、性能や信頼性、拡張性などの品質、要員訓練の方法、処理や操作の手順などを、システム戦略との整合性、実現可能性、妥当性の点から評価し、決定します。なお、システム要件定義において業務上の要件が満たされているか利用部門の長の確認を受ける必要があります。

### １）システム化の目標と対象範囲

何を目標にシステム化を行うのかシステム化の目標を明らかにし、対象となる業務と部署を決定し、システム要件定義書にまとめます。

🏋プラスアルファ

**●機能及び能力の定義**

システム要件定義書では、対象データの種類と形式、処理の内容、操作方法、表示画面の方法、出力帳票の形式などシステムの機能要件と、スループットやレスポンスタイムといった性能要件を明らかにします。

**●業務・組織及び利用者の要件**

システム要件定義書では、業務処理手順、入出力情報要件、操作要件など利用者の要求事項をシステム開発の項目に対応させて定義します。また、開発システムの具体的な利用方法を調査、分析して要件を抽出し、5W2H（Why、When、Where、Who、What、How、How much）の観点から文書化します。

**●その他の要件**

システム要件定義書には、システムの実行環境、信頼性や拡張性などの品質要件なども記載します。

## 2. システム要件の評価

システム要件の評価では、決定したシステム要件がシステム戦略と整合しているか、実現可能か、妥当かなどを、開発者と利用者が共同でレビューします。

|  |
| --- |
| 例題  システム開発の最初の工程で行う作業として，適切なものはどれか。  ア　各プログラムの内部構造を設計する。  イ　現状の業務を分析し，システム要件を整理する。  ウ　サブシステムをプログラム単位まで分割し，各プログラムの詳細を設計する。  エ　ユーザインタフェースを設計する。  システム開発で最初に行う作業がシステム要件定義です。システム要件定義では、現状の業務を分析し、システム要件の定義と評価、レビューを行います。  ア、ウはソフトウェア詳細設計の行われる作業です。エはソフトウェア要件定義で行われる作業です。  基本情報　平成21年度春　問44　[出題頻度：★☆☆]  解答－イ |

# 2. システム方式設計

学習のポイント

✅ システム方式設計の概要をつかんでおこう！

## １．システムの最上位レベルでの方式確立

システム方式設計では、システム要件定義で決定した課題を実現する方式を、ハードウェア、ソフトウェア、手作業の機能別に分類し、必要なシステム構成を決定します。そして、システムが機能を満たしているかを確認するために必要なテストの範囲、手順などを決定します。

### １）ハードウェア・ソフトウェア・手作業の機能分割

全てのシステム要件を、業務効率、作業負荷、作業コストなどの条件に基づいて、ハードウェア、ソフトウェア、手作業に振り分けます。

### ２）ハードウェア方式設計

ハードウェア方式設計では、性能や信頼性、拡張性などを考慮して、ハードウェアの構成を決定します。

### ３）ソフトウェア方式設計

ソフトウェア方式設計では、ソフトウェアを新規に開発するか、既存のソフトウェアパッケージを利用するかなど、ソフトウェアの構成を決定します。

### ４）システム処理方式設計

システム処理方式設計では、集中処理、分散処理、クライアントサーバシステム、Webシステムなど、システムの処理方式を決定します。

### ５）データベース方式設計

データベース方式設計では、使用するデータベースの種類、冗長性などを決定します。

## ２．システム結合テストの設計

システム結合テストの設計では、システム結合テストの範囲、テスト計画、テストの手順などを検討し、システムが利用者の要求する機能を全て満たしているかどうかを確認するシステム結合テスト仕様書を作成します。

## ３．システム方式の評価及びレビュー

システム方式の評価では、決定したシステム方式がシステム要件を満たしているか、実現可能かなどを、開発者と利用者が共同でレビューします。

|  |
| --- |
| 例題　🏋プラスアルファ  図のプロセスでシステム開発を進める場合，システム方式設計に含める作業として，適切なものはどれか。  システム  要件定義  システム  方式設計  ソフトウェア  要件定義  ソフトウェア  方式設計  ア　システムの機能及び処理能力の決定  イ　ソフトウェアの最上位レベルの構造とソフトウェアコンポーネントの決定  ウ　ハードウェアやネットワークの構成の決定  エ　利用者インタフェースの決定  ア　システム要件定義で実施する作業です。  イ　ソフトウェア方式設計で実施する作業です。  エ　ソフトウェア要件定義で実施する作業です。  ITパスポート　平成24年度秋　問32　[出題頻度：★☆☆]  解答－ウ |

# 3. ソフトウェア要件定義

学習のポイント

✅ DFDの特徴、使用する記号の意味を覚えよう！

✅ E-R図は、暗記だけでなく、「多重度（1対1、1対多など）」まで含めて出題される問題に備えよう！

✅ UMLは、各図の特徴と見方を覚えよう！

ソフトウェア要件定義では、ソフトウェア要件の確立、ソフトウェア要件の評価を行います。

## 1. ソフトウェア要件の確立

**ソフトウェア要件定義**では、DFDなどを用いて業務モデル、E-R図などを用いて論理データモデルを作成します。そして、システムを構成するソフトウェアに求められる機能、性能や信頼性、拡張性、インタフェースなどを決定します。

### ①サブシステムの機能仕様とそのインタフェースの設計

システムをいくつかのサブシステムに分割し、サブシステムごとに機能を決定し、サブシステム間の関連を明らかにして、サブシステムインタフェースを設計します。

### ②業務モデルとデータモデルの設計

業務フローやサブシステム間の関係から、業務モデルとデータモデルを作成します。

### ③セキュリティの設計

セキュリティポリシに則したセキュリティ機能の実現方法を設計します。

### ④保守性の考慮

運用開始後の、機能変更や新機能追加に容易に対応できるように、システムの保守性を考慮して設計します。

## 2. ソフトウェア要件の評価

ソフトウェア要件の評価では、決定したソフトウェア要件が、システム要件を満たしているか、システム方式に合致しているか、実現可能かなどを、開発者と利用者が共同でレビューします。

## 3. 業務分析や要件分析に用いられる手法

業務分析や要件分析に用いられる手法には、DFD、E-R図、UMLなどがあります。

### ①DFD（Data Flow Diagram）

DFDは、バブルチャートともいい、物や金の流れを外し、データの流れに着目して表現する技法です。業務プロセスをデータの流れに着目して表現する場合に使用します。

|  |  |  |
| --- | --- | --- |
| 記号 | 名称 | 説明 |
| 名称  名称  名称 | データフロー  （流れ） | データの入力源から、データの出力先までのデータの流れ（変換状態）を表す。 |
|  | プロセス  （処理） | システム中の各データは単純に入力から出力へと流れるだけでなく、演算などの加工が行われる場合がある。プロセスはこのような、データの加工、変換を行う場合に記述される。なお、階層などを識別できるように番号をつけるときもある。 |
|  | ファイル  または  データストア  （蓄積） | データの中には、他の処理へ受け渡しをしたり、共通に使用したりするため、一時的に記憶しておく必要のあるものが存在する。ファイルはこのような一時的に記憶しておく媒体を表す。 |
|  | データ源泉  （入力源）  データ吸収  （出力先） | システムが処理を行うときのおおもととなるデータの発生源や、処理した結果の受け渡し先である出力先を表す。 |

DFDで使用する記号

DFDは、システムの構成要素を分割し、データの流れを中心に図形を用いて記述しているため、あいまいさを排除でき、システムの抜けを防ぐことができるという長所があります。しかし、時間の概念を盛り込むことができないため、その現象がいつ起きるのかはわからないといった短所もあります。

なお、現行のシステムに代わる新しいシステムを構築する際、分析段階において各種のDFDを用い、現行のシステムおよび新しいシステムのモデル化を行うことになります。その際、まず現行のシステムの実際のデータの流れから現物理モデルを作成し、これをもとに現論理モデルを作成、次に新論理モデルを作成し、最後に実際に新しいシステムに実装するために新物理モデルを作成します。

発送

担当者

受注伝票

の作成

会員情報

の検索と

確認

商品情報

の検索と

確認

受注金額

の計算と

確認

会員

（注文客）

会員情報

商品情報

商品情報

商品コード

会員情報

受注伝票

受注金額、合計金額

数量、受注金額、合計金額

商品台帳

会員台帳

単価

DFDで表現した受注業務の例

|  |
| --- |
| 例題  図は構造化分析法で用いられるDFDの例である。図中の“○”が表しているものはどれか。  ア　アクティビティ イ　データストア  ウ　データフロー エ　プロセス  DFD（データフローダイアグラム）で使用する記号には、次の４つがあります。  （データフロー）…データ源泉からデータ吸収までの流れを表します。  （プロセス）…演算などのデータの加工、変換を行う処理を表します。  （データストア（ファイル））…一時的に記憶しておく媒体を表します。  （データ源泉、データ吸収）…データの発生源、データの受け渡し先を表します。  基本情報　平成22年度春　問45　[出題頻度：★★★]  解答－エ |

別冊演習ドリル 》 1-822～826

### ②E-R図

E-R図（Entity Relationship Diagram）は、業務で扱う情報を抽象化し、人や物、場所、金、サービスなどをさす実体（Entity）という概念と、実体間の関連性である関連（Relationship）という概念の２つでモデル化する技法です。データベースを設計する前段階で、よく用いられます。

実体と実体の間の関連について、E-R図ではそれぞれ次のように表します。

実体

実体

1

1

実体

実体

1

＊

１対１

１対多

E-R図の表記法

|  |
| --- |
| 例題  学校における教師，クラス，教室の三つのエンティティの関係を表すE-R図(a)，(b)について考える。この二つのE-R図の読み方のうち，適切なものはどれか。ここで，“１　１”は1対1の対応関係を表し，“１　＊”は1対多の対応関係を表す。  教師  クラス  教室  1　　1  1　　1  教師  クラス  教室  1　 ＊  ＊ 　1  (a)  (b)  ア　(a)では，教師か教室を決めるとクラスが決まるが，(b)では，教師と教室を決めるとクラスが決まる。  イ　(a)では，教師の担当できるクラスは一つだけであるが，(b)では，複数のクラスを担当することがある。  ウ　(a)では，一つのクラスには常に同じ教室が割り当てられるが，(b)では，一つのクラスに複数の教室が割り当てられることがある。  エ　(a)では，一つのクラスは常に1人の教師が担当するが，(b)では，一つのクラスを複数の教師で担当することがある。  (a)は、教師とクラスの関係が1対1のため、教師の担当できるクラスは1つだけです。また、クラスと教室の関係も1対1のため、1つのクラスに1つの教室が割り当てられます。(b)は、教師とクラスの関係が1対多のため、1人の教師が複数のクラスを担当することがあります。また、クラスと教室の関係が多対1のため、複数のクラスが1つの教室を割り当てられることがあります。  ア　(a)の説明は適切ですが、(b)が不適切です。(b)では、たとえば、1人の教師が2つのクラスを担当していて、その2つのクラスは、同じ教室を割り当てられた場合、教師と教室を決めても2つのクラスが該当することになります。  ウ　(a)の説明は適切ですが、(b)が不適切です。(b)では、複数のクラスが1つの教室を割り当てられることがあります。  エ　(a)の説明は適切ですが、(b)が不適切です。(b)では、１人の教師が複数のクラスを担当することがあります。  基本情報　平成16年度秋　問49　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-827～833

### ③UML

UML（Unified Modeling Language）は、開発中のオブジェクト指向システムの成果を定義し、視覚化し、文書化するための言語で、比較的広く用いられていた３種類のオブジェクト指向方法論（OMT法、Booch法、OOSE法）の概念を統一した標準表記法として考案されました。そのため言語とはいっても、一般的なプログラム言語とは異なり、ユースケース図、クラス図、シーケンス図、コミュニケーション図、ステートチャート図などによる表記法のことを指しています。なおUMLは、システム分析設計だけではなく、業務の分析にも適用できます。

|  |
| --- |
| 例題  業務プロセスのモデリング表記法として用いられ，複数のモデル図法を体系化したものはどれか。  ア　DFD イ　E-R図 ウ　UML エ　状態遷移図  ア　DFDは、物や金の流れを外し、データの流れに着目して表現する技法です。  イ　E-R図は、企業活動で取り扱うデータの構造を、人や物、場所、金、サービスなどをさす実体という概念と、実体間の関連性である関連という概念の２つでモデル化する技法です。  エ　状態遷移図は、時間の経過やデータの状態変化など、時間やタイミングなどによって変化していくようなシステムを記述するのに用いられる技法です。  基本情報　平成23年度秋　問63　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-834～836

**ユースケース図**は、利用者がシステムに求める機能を利用者の視点で表現するための図で、利用者とシステムとのやりとりを、次の４種類の記号を使って定義します。

|  |  |
| --- | --- |
| 名称 | 概要 |
| ユースケース | 横長の楕円の中に、システムが提供する機能を記述する。 |
| アクター | 線で描いた人型の下に、人や組織、外部システムの名称を記述する。 |
| システム境界 | 長方形で囲んで、システム化の対象（システムの範囲）を示す。 |
| 関連 | ユースケースとアクターの間に関連がある場合に線で結ぶ。 |

ユースケース図で使われる記号

|  |
| --- |
| 例題  次のユースケース図の説明はどれか。  注文管理システム  在庫管理システム  注文担当者  ア　在庫管理システムは人ではないので，アクターは注文担当者だけである。  イ　システム化の対象には，注文管理システムと在庫管理システムが含まれる。  ウ　注文管理システムはシステム境界の中にあり，注文担当者と在庫管理システムはシステム境界の外にある。  エ　ユースケースは四つあり，各ユースケース名は，注文管理システム，注文処理，注文変更処理，注文取消処理である。  ア　注文担当者と在庫管理システムが、アクターです。  イ　システム化の対象は、注文管理システムです。  エ　ユースケースは、注文処理、注文変更処理、注文取消処理の３つです。  初級システムアドミニストレータ　平成18年度春　問31　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-837～839

**クラス図**は、システムの構成要素とその相互作用を静的に表現した図です。

クラス図には、クラス名、属性（データ）、メソッド（操作）を記入します。なお、クラス名や操作をどこまで公開するかという可視性の代表的なレベルには次のものがあります。

・public（表記方法は＋） ：どのクラスからも参照可能

・private（表記方法は－） ：自クラスのみ参照可能

・protected（表記方法は＃） ：自クラスとその派生クラスで参照可能

本

タイトル

著者

キーワード

タイトルを参照する

著者を参照する

キーワードを参照する

クラス名

属性

メソッド

クラス図（例）

また、クラス図ではクラス間の関係を表すこともできます。クラス間の関連には次のようなものがあります。

|  |  |  |
| --- | --- | --- |
| 記号  クラス  クラス  １ ＊ | 名称 | 概要 |
|  | 関連 | 関連は、あるクラスと別のクラスの関係を表します。また、E-R図と同様の表記で数の関係性（多重度）を表現することもできます。 |
| 基底  クラス  派生  クラス | 汎化 | 汎化は、既存の複数のクラスの共通部分を抽出して新しいクラスを定義することで、新たに作成されたクラスを基底クラス、基底クラスの属性やメソッドを引き継いで利用するクラスを派生クラスといいます。 |
| 全体  部分 | 集約 | 集約は、複数のクラス間で「全体－部分」が成り立つ関係のことです。 |
| 利用 される側  利用  する側  全体  部分 | コンポジション | コンポジションは、集約関係のうち、全体が存在しなければ部分が存在しないような強い集約関係を表します。 |
|  | 依存 | 依存は、処理の実現のために他のクラスを利用したり、一方の変化が他方に影響を与えるなどの関係を表します。 |

クラス間の関連の種類

|  |
| --- |
| 例題  UMLを用いて表した図のデータモデルの多重度の説明のうち，適切なものはどれか。  組織  社員  １ １..\*  ア　社員が１人も所属しない組織は存在しない。  イ　社員は必ずしも組織に所属しなくてもよい。  ウ　社員は複数の組織に所属することができる。  エ　一つの組織に複数の社員は所属できない。  UMLを用いたクラスの多重度の表現方法には次のようなものがあります。  クラス  クラス  ＊  1  ＊  n..m  クラス  クラス  クラス  クラス  ＊  0..＊  クラス  クラス  ＊  1..＊  ①１は１を、＊は複数を表す。  ②0..＊は０以上を表す  ③1..＊は１以上を表す  ④n..mはnからmの範囲を表す  したがって、問題の図は、社員はどこか１つの組織に属しており、組織には必ず１人以上の社員が所属していることを意味しています。  基本情報　平成23年度春　問29　[出題頻度：★★★]  解答－ア |

別冊演習ドリル 》 1-840～849

**シーケンス図**は、オブジェクトの相互作用を時間経過に沿って表現した図です。

利用者情報取得

：係員

：貸出し

：利用者管理

システム

：貸出・返却

システム

貸出申込み

利用者情報

貸出し許可

貸出中か？

貸出し可能か？

YES

利用者ID入力

所蔵図書か？

図書ID入力

NO

シーケンス図（例）

|  |
| --- |
| 例題  UML2.0で定義している図のうち，動的な振る舞いを表現するものはどれか。  ア　オブジェクト図 イ　クラス図 ウ　シーケンス図 エ　パッケージ図  ア　オブジェクト図は、クラスから生成される個々のオブジェクト（インスタンス）同士の関係を表現した図です。  イ　クラス図は、システムの構成要素とその相互作用を静的に表現した図です。  エ　パッケージ図は、クラスを一定の条件でグループ化したパッケージ同士の関係を表現した図です。  基本情報　平成21年度秋　問45　[出題頻度：★★☆]  解答－ウ |

別冊演習ドリル 》 1-850,851

🏋プラスアルファ

**コミュニケーション図**（コラボレーション図）は、オブジェクト間のやりとりを構造的に表現した図です。

|  |
| --- |
| 例題　🏋プラスアルファ  UML2.0のコミュニケーション図が表現するのはどれか。  ア　インターフェースを介したコンポーネント同士の関係やコンポーネントの内容を表現する。  イ　オブジェクト間の相互作用を表現し，オブジェクト間のデータリンクを強調する。  ウ　クラス間の静的な関係を表現する。  エ　時間軸に沿って，状態遷移，時間制約，メッセージの送受信などを表現する。  ア　コンポーネント図に関する記述です。  ウ　クラス図に関する記述です。  エ　シーケンス図に関する記述です。  ソフトウェア開発　平成19年度秋　問42　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-852,853

🏋プラスアルファ

**オブジェクト図**は、クラスから生成される個々のオブジェクト（インスタンス）同士の関係を表現した図です。

|  |
| --- |
| 例題　🏋プラスアルファ  UMLのダイアグラムのうち，インスタンス間の関係を表現するものはどれか。  ア　アクティビティ図 イ　オブジェクト図 ウ　コンポーネント図 エ　ユースケース図  ア　アクティビティ図は、業務手順、ワークフローを表現します。  ウ　コンポーネント図は、ソフトウェアモジュールの構成を表現します。  エ　ユースケース図は、システムがもつ機能を表現します。  基本情報　平成24年度秋　問46　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-854

🏋プラスアルファ

**アクティビティ図**は、業務手順、ワークフローを表現した図です。

|  |
| --- |
| 例題　🏋プラスアルファ  UMLのアクティビティ図の特徴はどれか。  ア　オブジェクト間のメッセージのやり取りが，時系列で表現できる。  イ　オブジェクト群の相互作用を示す協調関係が表現できる。  ウ　現実のビジネスプロセスで生じる並行処理が表現できる。  エ　一つのオブジェクトに着目して，その状態遷移が表現できる。  ア　シーケンス図に関する記述です。  イ　コミュニケーション図（コラボレーション図）に関する記述です。  エ　ステートチャート図（ステートマシン図）に関する記述です。  システムアーキテクト　平成21年度秋Ⅱ　問1　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-855,856

### ④状態遷移図

**状態遷移図**は、時間の経過やデータの状態変化など、時間やタイミングなどによって変化していくようなシステムを記述するのに用いられる技法です。図を作成するに当たっては、偶発的に発生する事象や例外的なケース、誤りや失敗なども考慮しなければなりません。

S0

S1

S2

50／ジュース

100／ジュース＋50

100／ジュース

50／＊

50／＊

100／＊

（基本情報　平成15年度春　問10より抜粋）

|  |
| --- |
| 例題  設計するときに，状態遷移図を用いることが最も適切なシステムはどれか。  ア　月末及び決算時の棚卸資産を集計処理する在庫棚卸システム  イ　システム資源の日次の稼働状況を，レポートとして出力するシステム資源稼動状況報告システム  ウ　水道の検針データを入力として，料金を計算する水道料金計算システム  エ　設置したセンサの情報から，温室内の環境を最適に保つ温室制御システム  状態遷移図は、時間の経過やデータの状態変化など、時間やタイミングにより変化していくようなシステムを記述するのに用いられる技法です。ある状態に対する事象を表現する技法ですが、偶発的に発生する事象や例外的なケースなども考慮する必要があります。  基本情報　平成25年度春　問46　[出題頻度：★★★]  解答－エ |

別冊演習ドリル 》 1-857～859

# 4. ソフトウェア方式設計・ソフトウェア詳細設計

学習のポイント

✅ 5. ソフトウェア設計手法の「オブジェクト指向設計」は、深い知識を必要とせずに解ける問題が多数あり！例題  
 　　 や演習ドリルの問題を解きながら、必要な知識を身につけよう！

✅ 6. モジュール設計は、最も強度が強い「機能的強度」と、最も強度が弱い「暗号的強度」を中心に覚えよう！

## 1. ソフトウェア方式設計

ソフトウェア方式設計では、ソフトウェア要件定義で決定した課題を、開発側の視点からソフトウェアコンポーネント（プログラム）まで分割し、各ソフトウェアコンポーネントの機能とソフトウェアコンポーネント間の処理の手順や関係を決定します。

|  |
| --- |
| 例題  開発プロセスにおいて，ソフトウェア方式設計で行うべき作業はどれか。  ア　顧客に意見を求めて仕様を決定する。  イ　ソフトウェア品目に対する要件を，最上位レベルの構造を表現する方式であって，かつ，ソフトウェアコンポーネントを識別する方式に変換する。  ウ　プログラムを，コード化した１行の処理まで明確になるように詳細化する。  エ　要求内容を図表などの形式でまとめ，段階的に詳細化して分析する。  ア、エ　ソフトウェア要件定義で行うべき作業です。  ウ　ソフトウェア詳細設計で行うべき作業です。  基本情報　平成30年度春　問47　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-860

## 2. ソフトウェア詳細設計

ソフトウェア詳細設計では、各ソフトウェアコンポーネントをプログラミング（コーディング、コンパイル、テスト）の実施単位であるソフトウェアユニット（モジュール）レベルまで詳細化します。

|  |
| --- |
| 例題　🏋プラスアルファ  図は開発するソフトウェアの動作ロジックを検討し，その結果を文書化した流れ図の一部である。このような作業を実施するのはソフトウェア開発のどの段階か。  “エラー”と表示  IN≧1  “エラー”と表示  IN≧4 AND  IN≦9  IN≦12  “上半期”と表示  “下半期”と表示  Yes  Yes  Yes  No  No  No  ア　システム方式設計 イ　システム要件定義  ウ　ソフトウェア詳細設計 エ　ソフトウェア要件定義  ア　システム方式設計では、システムの最上位レベルでの方式確立、システム結合のためのテスト要求事項の定義、システム方式の評価、共同レビューなどが行われます。  イ　システム要件定義では、システム要件の定義、評価、共同レビューなどが行われます。  エ　ソフトウェア要件定義では、ソフトウェア要件の確立、評価が行われます。  ITパスポート　平成22年度春　問50　[出題頻度：★☆☆]  解答－ウ |

## 3. インタフェース設計

ソフトウェア要件定義で決定した課題を実現するために、操作性や視認性などを考慮した画面や帳票のレイアウトの設計（**外部設計**）や、入出力装置を介して取り扱われるデータの物理設計（**内部設計**）を行います。

|  |
| --- |
| 例題  システムの外部設計を完了させるとき，顧客から承認を受けるものはどれか。  ア　画面レイアウト イ　システム開発計画  ウ　物理データベース仕様 エ　プログラムの流れ図  イ　基本計画に関する記述です。  ウ　内部設計に関する記述です。  エ　プログラム設計に関する記述です。  基本情報　平成29年度春　問46　[出題頻度：★★☆]  解答－ア |

別冊演習ドリル 》 1-861,862

## 4. ソフトウェア品質

ソフトウェアの品質をいかに保つかは、システム開発を行う上で重要な課題です。

ソフトウェアの品質を考えるとき、2001年にISOで定められたソフトウェア製品の評価－品質特性とその適用に関するガイドライン（ISO/IEC 9126）が基準となります。この基準をもとに、国内では2003年にJIS規格としてJIS X 0129が制定されました。その後、2011年にISO/IEC 9126を補強する形で**ISO/IEC 25010**が定められ、これに伴い2013年にJIS規格もJIS X 0129から**JIS X 25010**に改訂されました。具体的には、「利用時の品質モデル」と「製品品質モデル」に分割されています。また、JIS X 0129では6つだった製品品質特性は8つに拡大され、品質特性を構成する品質副特性も21個から32個に増えています。

🏋プラスアルファ

**●利用時の品質モデル**

JIS X 25010には、「利用時の品質モデルは、特定の利用者が特定の利用状況において、有効性、効率性、リスク回避性及び満足性に関して特定の目標を達成するためのニーズを満たすために、製品又はシステムを利用できる度合いのことである。利用時の品質の特徴は、有効性、効率性、満足性、リスク回避性及び利用状況網羅性の五つの特性に分類される。」と記述されています。

|  |  |  |
| --- | --- | --- |
| 品質特性 | 品質副特性 | 主な内容 |
| 有効性 |  | 明示された目標を利用者が達成する上での正確さ及び完全さの度合い。 |
| 効率性 |  | 利用者が特定の目標を達成するための正確さ及び完全さに関連して、使用した資源の度合い。 |
| 満足性 | 実用性  信用性  快感性  快適性 | 製品又はシステムが明示された利用状況において使用されるとき、利用者ニーズが満足される度合い。 |
| リスク回避性 | 経済リスク緩和性  健康・安全リスク緩和性  環境リスク緩和性 | 製品又はシステムが、経済状況、人間の生活又は環境に対する潜在的なリスクを緩和する度合い。 |
| 利用状況網羅性 | 利用状況完全性  柔軟性 | 明示された利用状況及び当初明確に識別されていた状況を超越した状況の両方の状況において、有効性、効率性、リスク回避性及び満足性を伴って製品又はシステムが使用できる度合い。 |

表　利用時の品質特性及び品質副特性

🏋プラスアルファ

**●製品品質モデル**

JIS X 25010には、「製品品質モデルは、製品品質特徴を八つの特性（機能適合性、性能効率性、互換性、使用性、信頼性、セキュリティ、保守性及び移植性）に分類する。各特性は、関係する副特性の集合から構成される。」と記述されています。

|  |  |  |
| --- | --- | --- |
| 品質特性 | 品質副特性 | 主な内容 |
| 機能適合性 | 機能完全性  機能正確性  機能適切性 | 明示された状況下で使用するとき、明示的ニーズ及び暗黙のニーズを満足させる機能を、製品又はシステムが提供する度合い。 |
| 性能効率性 | 時間効率性  資源効率性  容量満足性 | 明記された状態（条件）で使用する資源の量に関係する性能の度合い。 |
| 互換性 | 共存性  相互運用性 | 同じハードウェア環境又はソフトウェア環境を共有する間、製品、システム又は構成要素が他の製品、システム又は構成要素の情報を交換することができる度合い、及び／又はその要求された機能を実行することができる度合い。 |
| 使用性 | 適切度認識性  習得性  運用操作性  ユーザエラー防止性  ユーザインタフェース快美性  アクセシビリティ | 明示された利用状況において、有効性、効率性及び満足性をもって明示された目標を達成するために、明示された利用者が製品又はシステムを利用することができる度合い。 |
| 信頼性 | 成熟性  可用性  障害許容性（耐故障性）  回復性 | 明示された時間帯で、明示された条件下に、システム、製品又は構成要素が明示された機能を実行する度合い。 |
| セキュリティ | 機能性  インテグリティ  否認防止性  責任追跡性  真正性 | 人間又は他の製品若しくはシステムが、認められた権限の種類及び水準に応じたデータアクセスの度合いをもてるように、製品又はシステムが情報及びデータを保護する度合い。 |
| 保守性 | モジュール性  再利用性  解析性  修正性  試験性 | 意図した保守者によって、製品又はシステムが修正することができる有効性及び効率性の度合い。 |
| 移植性 | 適応性  設置性  置換性 | 1つのハードウェア、ソフトウェア又は他の運用環境若しくは利用環境からその他の環境に、システム、製品又は構成要素を移すことができる有効性及び効率性の度合い。 |

表　製品品質特性及び品質副特性

|  |
| --- |
| 例題　🏋プラスアルファ  JIS X 25010：2013で規定されたシステム及びソフトウェア製品の品質副特性の説明のうち，信頼性に分類されるものはどれか。  ア　製品又はシステムが，それらを運用操作しやすく，制御しやすくする属性をもっている度合い  イ　製品若しくはシステムの一つ以上の部分への意図した変更が製品若しくはシステムに与える影響を総合評価すること，欠陥若しくは故障の原因を診断すること，又は修正しなければならない部分を識別することが可能であることについての有効性及び効率性の度合い  ウ　中断時又は故障時に，製品又はシステムが直接的に影響を受けたデータを回復し，システムを希望する状態に復元することができる度合い  エ　二つ以上のシステム，製品又は構成要素が情報を交換し，既に交換された情報を使用することができる度合い  ア　運用操作性に分類されます。  イ　解析性に分類されます。  エ　相互運用性に分類されます。  応用情報　平成27年度春　問48　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-863

## 5. ソフトウェア設計手法

ソフトウェア設計手法は、業務手続（プロセス）をそのまま記述するプロセス中心設計から、業務データの構造を決めてからプロセスを設計するデータ中心設計、そしてデータとプロセスを一体化したオブジェクトとして扱いオブジェクトごとに設計するオブジェクト指向設計へと時代とともに新しい手法が考え出されてきました。どの技法を用いるかは、開発環境などによって決定されますが、具体的な設計作業では、段階的詳細化という構造化設計技法の考え方に基づいて、モジュール分割が行われ、モジュールごとに設計、プログラミングされます。

### １）プロセス中心設計

プロセス中心設計は、伝統的なシステム開発の手法で、従来、手作業で行っていた業務をそのままコンピュータに置き換えることから、業務手続（プロセス）に沿って設計を行います。そのため、経営環境の変化によって組織や業務手続が変化した場合に、システムを大きく変更しなければなりません。

### ２）データ中心設計

データ中心設計は、データ中心アプローチ（DOA：Data Oriented Approach）とも呼ばれ、組織や業務手続が変化しても業務データはほとんど変化しない（プロセスはデータに付随する）という考え方から、まずデータの構造を決め、その後、そのデータを使用するプロセスの設計を行います。そのため、データの標準化が必要となり、対象業務のモデル化が重要となります。このとき、データの安定性という観点から、最も安定した情報資源に着目することが望ましいとされています。

データ設計を行う場合には、プロセスの方法や順序などは考慮しません。また、データ構造の安定性に着目しているため、開発後の仕様変更に関しても、データ構造を局所的に変更、追加することは、やや困難になります。

|  |
| --- |
| 例題  ソフトウェアの分析・設計技法の特徴のうち，データ中心分析・設計技法の特徴として，最も適切なものはどれか。  ア　機能を詳細化の過程で，モジュールの独立性が高くなるようにプログラムを分割していく。  イ　システムの開発後の仕様変更は，データ構造や手続の局所的な変更で対応可能なので，比較的容易に実現できる。  ウ　対象業務領域のモデル化に当たって，情報資源であるデータ構造に着目する。  エ　プログラムが最も効率よくアクセスできるようにデータ構造を設計する。  データ中心分析・設計技法は、データ中心アプローチとも呼ばれ、業務のやりかた（プロセス）は頻繁に変化するが、最も安定した情報資源（データ）は変化しないといった「データ不変の法則」という経験則に基づいたものです。  ア　機能単位の分割はプロセス中心設計技法に該当します。  イ　オブジェクト指向モデルに関する記述です。  エ　データ中心分析・設計技法に限ったことではありません。  応用情報　平成31年度春　問46　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-864,865

🏋プラスアルファ

**３）構造化設計**

構造化設計は、大規模化したプログラムを効率よくミスなく設計するための方法論です。「全てのプログラムは、順次、選択、繰返しの３つの基本制御構造の組合せで作ることができる」という構造化定理をもとに、1967年にダイクストラによって唱えられました。

構造化設計の手順には、主たる機能から始まって段階的に詳細化していくトップダウン方式（**段階的詳細化**）と、個々の小さな機能を段階的にまとめていくボトムアップ方式（**段階的統合化**）があります。

|  |
| --- |
| 例題　🏋プラスアルファ  構造化プログラミングにおいて，プログラムを作成するときに用いる三つの制御構造はどれか。  ア　繰返し，再帰，順次 イ　繰返し，再帰，選択  ウ　繰返し，順次，選択 エ　再帰，順次，選択  ソフトウェアの設計、作成、保守が近年では別々の人間により行われることから、アルゴリズムの「容易さ」は重要な条件です。構造化プログラミングとは、この条件を達成するために考案されたプログラム構築技法です。「１つの入り口と１つの出口を持つ構造であれば、３つの基本制御構造で表現できる」という構造化定理に基づいた構築方法であり、３つの基本制御構造とは、「順次」「選択」「繰返し」です。  基本情報　平成15年度春　問53　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-866

### ４）オブジェクト指向設計

コンピュータの世界では、画面上のアイコンやボタン、さらに文書中の画像や音声など、ユーザによって処理や操作の対象となるモノをオブジェクトといいます。オブジェクトは、単なる画像やデータではなく、いくつもの属性や手続きを伴っています。例えば１つのアイコンは、その名前、絵柄、ファイルサイズなどの「属性」をもち、操作に応じて選択する、移動する、ファイルを開く、という「処理（手続き）」を含んでいます。このように、オブジェクトとは、「データ」とそれに伴う「手続き（メソッド）」を１つにまとめたモノというわけです。

**オブジェクト指向**は、データとメソッドを1つにまとめて扱うことで、利用性の高い設計や開発を行う考え方です。例えば、「文書を表示する」「音楽を演奏する」という場合、「文書（データ）」と「表示する（メソッド）」、「音楽」と「演奏する」はともに対応関係にあり、「文書を演奏する」「音楽を表示する」とはなりません。このことから、「表示する」「演奏する」という手続きを、「文書」「音楽」というデータに一体化させ、メソッドの具体的な内容はオブジェクト自身に決めさせる、ということがオブジェクト指向の考え方です。

オブジェクト指向設計は、プログラムをオブジェクト単位で開発し、オブジェクトを部品のように組み合わせて利用できることや、他のプログラムに再利用できるといった特徴をもった技法です。

#### ①オブジェクト指向を形づくる基本的な言葉とその意味

**ⅰ）データ**

データは、情報を人やコンピュータが扱えるように、文字や数字、記号で表現したものです。

**ⅱ）メソッド**

メソッドは、手続き（振舞い）のことです。

**ⅲ）カプセル化**

カプセル化は、データとメソッドを一体化することです。

**ⅳ）オブジェクト**

オブジェクトは、カプセル化を行った結果として作成されたデータとメソッドを一体化したものです。

商品コード

在庫数量

データ

入庫

出庫

メソッド

商品コード

在庫数量

入庫

出庫

オブジェクト

カプセル化

データ、メソッド、カプセル化、オブジェクト（例）

**ⅴ）情報隠蔽**

情報隠蔽は、カプセル化したデータとメソッドをブラックボックス化することです。これにより、データに対するダイレクトなアクセスを禁止することにつながるので、オブジェクトの独立性が高まります。

商品コード

在庫数量

入庫

出庫

情報隠蔽（ブラックボックス）

商品コードを変更したい

商品を出庫したい

○

×

情報隠蔽（例）

**ⅵ）クラス**

クラスは、類似する複数のオブジェクトから共通する部分を抜き出して定義したもので、一種のテンプレート（ひな型）ということができます。また、クラスを階層化したときの上位のクラスを**スーパクラス**（**基底クラス**）と呼び、下位のクラスを**サブクラス**（**派生クラス**）と呼びます。

【クラス：企業】

企業名

住所 データ

電話番号

取引 メソッド

Ａ商事

オブジェクト

Ｂ商事

Ｃ商事

Ｄ商事

クラス（例）

**ⅶ）インスタンス**

インスタンスは、実体のことで、実際のデータ値や振舞いを意味します。クラスは、クラスのみでは実行することができません。クラスの定義に基づいて実際に値を与えたインスタンスを生成して実行します。したがって、１つのインスタンスには、１つのクラスが対応します。

○○商事

東京都豊島区

03-9876-5432

取引をする

××コーポレーション

東京都千代田区

03-2345-6789

取引をする

△△システム

東京都新宿区

03-3456-7890

取引をする

インスタンス（例）

|  |
| --- |
| 例題  オブジェクト指向における基底クラスと派生クラスの関係にあるものはどれか。  ア　“会社”と“社員” イ　“自動車”と“エンジン”  ウ　“図形”と“三角形” エ　“人間”と“頭”  オブジェクト指向において、同じ特性（性質）をもつオブジェクトの集合をクラスと呼びます。そして、複数のクラスに共通な特性のみを抽出したクラスを、元のクラスに対するスーパクラス、元のクラスをスーパクラスに対するサブクラスといいます。  一方、JavaやC++などのオブジェクト指向言語においては、一般的にスーパクラス、サブクラスはそれぞれ「基底（ベース）クラス」、「派生クラス」と呼ばれます。  呼称の違いはあるにせよ、基底（スーパ）クラスと派生（サブ）クラスの関係は製品と部品のような物質的なものではなく、特性に関するものであることは同じです。  選択肢の中で特性に関するものは、選択肢ウの“図形”と“三角形”のみであり、これが解答となります。  基本情報　平成20年度秋　問41　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-867～874

#### ②オブジェクト指向の特徴

**ⅰ）メッセージ**

メッセージは、オブジェクトに対して指示できる唯一の手段です。これ以外の方法でオブジェクトにアクセスさせないことで、オブジェクトの独立性を高めることができます。また、オブジェクトをブラックボックスと考え、オブジェクトの中身を知らなくても必要な操作ができるようにしています。

オブジェクト

（ブラックボックス化）

オブジェクトがどのような  
構造なのか知らなくてよい

〔メッセージ〕

Ａ商事の売上高

〔応答〕

２億円

〔メッセージ〕

Ｂ商事を削除

〔応答〕

OK

〔メッセージ〕

Ｅ商事を追加

〔応答〕

OK

メッセージ（例）

**ⅱ）インヘリタンス**（**継承**・相続）

インヘリタンスは、上位または下位のクラスへデータやメソッドを引き継ぐことです。次の例では、氏名、住所、電話番号、所属部署は各クラスに共通する項目です。インヘリタンスは、それぞれのクラスについて個別に定義しなければならない部分のみ定義（**差分プログラミング**）し、共通部分は上位クラス（スーパクラス）から継承することにより、生産性を向上させようとしています。

特化

汎化

※差分プログラミング

営 業 部　→　「担当顧客」のみ作成

システム部　→　「担当分野」のみ作成

【スーパクラス】

・氏名 ・住所

・電話番号 ・所属部署

社員

【サブクラス】

・担当顧客

営業部

【サブクラス】

・担当分野

システム部

インヘリタンス（例）

サブクラスの共通部分を抽出してスーパクラスを定義することを**汎化**といい、逆にスーパクラスを細分化して定義することを**特化**といいます。これらの関係を汎化・特化関係または、**is\_a関係**といいます。

なお、インスタンスを生成できず、インヘリタンスを前提としたクラスを**抽象クラス**と呼びます。

**ⅲ）ポリモフィズム**（多様性・**多相性**）

ポリモフィズムとは、同一メッセージを各オブジェクトに送っても、メッセージを受け取ったオブジェクトによって異なった動作をすることをいいます。なお、ポリモフィズムは、スーパクラスで定義されたメソッドをサブクラスで再定義する**オーバーライド**を行うことで実現します。

仕事をしなさい

社員

経理部

営業部

システム部

得意先を訪問します

伝票を集計します

プログラムを作ります

メッセージ

ポリモフィズム（例）

**ⅳ）複合オブジェクト**

複合オブジェクトとは、複数のオブジェクトを組み合わせて作られたオブジェクトのことです。

フロント

ホテル

客室

ベッド

浴室

トイレ

バスタブ

シャワー

part\_of

part\_of

part\_of

part\_of

part\_of

part\_of

part\_of

複合オブジェクト（例）

上の例では、さまざまな設備が組み合わさって1つの「ホテル」を構成します。この設備を1つのオブジェクトとしてとらえたとき、「ホテル」は複合オブジェクトということができます。このような構造を、集約・分解関係または**part\_of関係**といいます。

**ⅴ）委譲**

委譲（デリゲーション）とは、あるオブジェクトに送られたメッセージの内容がそのオブジェクトでは処理できないときに、他のオブジェクトにそのメッセージを送って処理をしてもらうことです。

|  |
| --- |
| 例題  次のクラス図におけるクラス間の関係の説明のうち，適切なものはどれか。  自動車  バス  トラック  セダン  バン  ・・・  ア　“バス”，“トラック”などのクラスが“自動車”クラスの定義を引き継ぐことを，インスタンスという。  イ　“バス”，“トラック”などのクラスの共通部分を抽出し“自動車”クラスとして定義することを，汎化という。  ウ　“バス”，“トラック”などのクラスは，“自動車”クラスに対するオブジェクトという。  エ　“バス”，“トラック”などのそれぞれのクラスの違いを“自動車”クラスとして定義することを，特化という。  クラスの階層構造を表わすために用いられるのが「汎化－特化構造」です。汎化は下位クラス（“バス”、“トラック”など）に共通する性質（“自動車”）をまとめて抽象化することであり、特化は上位クラスの性質をいくつかに分解して具現化することです。  ネットワーク　平成23年度秋Ⅱ　問24　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-875～883

## 6. モジュールの設計

システムが大規模になると、プログラムが複雑なものになり、それに伴ってエラーが発生する可能性も高くなってしまいます。複雑なプログラムは、理解しにくいだけでなく、システムの品質を低下させる原因になります。そのため、プログラムの構造を調べ、複数の機能が含まれているならば、プログラムを構成する最小機能単位である**モジュール**に分割します。

### １）分割基準

プログラムの信頼性や保守性などを高めるためには、構造的に単純なプログラムを設計しなければなりません。そのためには、モジュールの独立性が高くなるように分割します。なお、モジュールの独立性を高めるためには、モジュール内部の関連性（強度）を強く、モジュール相互の関連性（結合度）を弱くする方法がとられます。

#### ①モジュール強度

モジュール強度は、モジュール内部の関連性を表す尺度であり、モジュール内部の各要素の関連性が強いほど、モジュールの独立性が高くなります。なお、モジュール強度は、次の７段階に分けて評価します。

|  |  |  |  |
| --- | --- | --- | --- |
| 独立性 | 強度 | 名称 | モジュールの内容 |
| 低 | 低 | 暗号的強度 | 関連性の全くない複数の機能を実行するモジュール |
|  |  | 論理的強度 | 関連したいくつかの機能を１つにまとめ、そのうち１つを選択して実行するモジュール |
|  |  | 時間的強度 | ある特定の時期に複数の逐次的な機能を実行するモジュール |
|  |  | 手順的強度 | 複数の逐次的な機能を実行するモジュール |
|  |  | 連絡的強度 | 複数の逐次的な機能を実行し、機能間でデータの受け渡しを行ったり、あるいは同一データを参照するようなモジュール |
|  |  | 情報的強度 | 特定のデータ構造を扱う複数の機能を１つにまとめたモジュール |
| 高 | 高 | 機能的強度 | １つの固有の機能を実行するモジュール |

モジュール強度

|  |
| --- |
| 例題  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  ア　手順的強度に関する記述です。  イ　暗号的強度に関する記述です。  ウ　機能的強度に関する記述です。  エ　時間的強度に関する記述です。  モジュール強度の高い順に並べるとウ→ア→エ→イとなり、ウが解答となります。  応用情報　平成22年度春　問45　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-884,885

#### ②モジュール結合度

モジュール結合度は、モジュール間の関連性を表す尺度であり、他のモジュールとの関連性が弱いほど、モジュールの独立性が高くなります。なお、モジュール結合度は、次の６段階に分けて評価します。

|  |  |  |  |
| --- | --- | --- | --- |
| 独立性 | 結合度 | 名称 | モジュールの内容 |
| 低 | 高 | 内容結合 | あるモジュールが他のモジュールの内容の一部を共有したり、他のモジュール内のデータを直接参照するような場合 |
|  |  | 共通結合 | 共通域に定義されたデータ構造（複数のデータを集めたもの）を複数のモジュールで共有する場合 |
|  |  | 外部結合 | 外部宣言したデータを複数のモジュールで共有する場合 |
|  |  | 制御結合 | モジュールの実行に影響を与えるような制御情報の受け渡しを行う場合 |
|  |  | スタンプ結合 | 複数のモジュール間で、共通域にないデータ構造の受け渡しを行う場合 |
| 高 | 低 | データ結合 | 複数のモジュール間で、必要なデータだけを受け渡しする場合 |

モジュールの結合度

|  |
| --- |
| 例題  モジュール結合度が最も弱くなるものはどれか。  ア　一つのモジュールで，できるだけ多くの機能を実現する。  イ　二つのモジュール間で必要なデータ項目だけを引数として渡す。  ウ　他のモジュールとデータ項目を共有するためにグローバルな領域を使用する。  エ　他のモジュールを呼び出すときに，呼び出したモジュールの論理を制御するための引数を渡す。  複数のモジュール間で、必要なデータだけを受け渡すデータ結合が最も結合度が弱いです。  ア　モジュール強度に関する記述です。  ウ　共通結合に関する記述です。  エ　制御結合に関する記述です。  基本情報　令和4年度サンプル　問39　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-886～887

## 7. レビュー

レビューは、システム開発の各工程において、それぞれ複数の関係者が集まり、文書（ドキュメント）の問題点を洗い出すために行われる討議で、ソフトウェアの品質を確保するためには欠かすことはできません。なお、レビューの目的はエラーを検出することであり、解決策を検討する必要はありません。

レビュー実施に当たっては、参加者は事前に資料を受け取り、十分検討しておく必要があります。実施時間は１、２時間程度、参加者は４、５名程度とし、なるべく管理者は参加しないほうが望ましいといえます。また、参加者全体の連帯責任とし、発見されたエラーは文書化します。

### １）レビューの対象と種類

外部仕様を対象としたレビューでは、ユーザの希望が網羅されているか、それが実現できるのかを検証します。内部仕様を対象としたレビュー、つまりソフトウェアを制作する段階でのレビューでは、外部仕様との一貫性の検証と内部仕様に矛盾がないか、外部仕様での要求定義が実現されているかを確認します。なお、設計工程で行われるレビューを**デザインレビュー**、開発工程で行われるレビューを**コードレビュー**と呼びます。

|  |
| --- |
| 例題  デザインレビューを実施するねらいとして，適切なものはどれか。  ア　開発スケジュールを見直し，実現可能なスケジュールに変更する。  イ　仕様の不備や設計の誤りなどを早期に発見し，手戻り工数の削減を図る。  ウ　設計工程での誤りの混入を防止し，テストを簡略化して，開発効率の向上を図る。  エ　設計の品質を向上させることで，開発規模見積りの精度の向上を図る。  ア　スケジュールの見直しが目的ではありません。  ウ　テストを簡略化することはできません。  エ　開発規模見積りの精度を向上させることはできません。  基本情報　平成20年度春　問46　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-889～891

### ２）レビューの代表的な方法

レビューの代表的な方法には、インスペクション、ウォークスルー、ラウンドロビンの３つがあります。

#### ①インスペクション

インスペクションは、**モデレータ**と呼ばれるこの作業の責任者（インスペクションリーダ）が中心となって行う組織的なレビューです。インスペクションにおいては、モデレータが主催し、出席者の選定や発見されたエラーの修正内容の承認を行います。

#### ②ウォークスルー

ウォークスルーは、もともとは、開発されたプログラムに対して適当な値を入力して、机上で実行過程を模擬的にたどる、という検査方法を指す言葉でしたが、現在では、レビュー対象物の作成者がその内容を説明し、複数の担当者が随時質問するという形式で進められるレビューを意味します。ウォークスルーでは、開発者自身が主催するという点がインスペクションと大きく異なります。

#### ③ラウンドロビン

ラウンドロビンは、レビューの参加者が持ち回りでレビュー責任者を務めながら、全体としてレビューを遂行していくものです。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  ａ～ｃの説明とレビューとの対応について，適切な組合せはどれか。  ａ　参加者全員が持ち回りでレビュー責任者を務めながらレビューを行うので，参加者全員の参画意欲が高まる。  ｂ　レビュー対象物の作成者が説明者になり，入力データの値を仮定して，手続をステップごとに机上でシミュレーションしながらレビューを行う。  ｃ　あらかじめ参加者の役割を決めておくとともに，進行役の議長を固定し，レビューの焦点を絞って迅速にレビュー対象を評価する。   |  |  |  |  | | --- | --- | --- | --- | |  | ａ | ｂ | ｃ | | ア | インスペクション | ウォークスルー | ラウンドロビン | | イ | ウォークスルー | インスペクション | ラウンドロビン | | ウ | ラウンドロビン | インスペクション | ウォークスルー | | エ | ラウンドロビン | ウォークスルー | インスペクション |   各選択肢の内容を検討すると次のようになります。  ａ　参加者全員がレビュー責任者を務めながら遂行していくレビューであるので、ラウンドロビンに関する記述です。  ｂ　入力データの値を仮定してコードを追跡し、机上でシミュレーションを行うことから、ウォークスルーに関する記述です。  ｃ　あらかじめ参加者の役割を決定しレビューに参加することから、インスペクションに関する記述です。  ソフトウェア開発　平成19年度秋　問44　[出題頻度：★★☆]  解答－エ |

別冊演習ドリル 》 1-892～894

# 5. ソフトウェア構築

## 1. ソフトウェアコード作成

学習のポイント

✅ 出題頻度を参考に効率よく学習しよう！

ソフトウェア詳細設計書に基づいて、プログラム言語を用いてソフトウェアコード（プログラム）を記述（コーディング）し、作成した個々のプログラムに誤り（バグ）がないかを検証する作業を**プログラミング**と呼びます。

## 2. コーディング基準

学習のポイント

✅ 出題頻度を参考に効率よく学習しよう！

プログラミングでは、将来のメンテナンスや修正のためにも、わかりやすいプログラムを記述することが重要になります。そのため、コーディング基準（記述方法に関する基準）を設けて、プログラムの標準化を図る必要があります。具体的には、変数の命名に関する規約や、大文字・小文字の使用用途の定義、コメントの使用方法などが規約としてあげられます。

コーディング基準に沿って作成されたプログラムは、開発容易性や保守容易性という観点から品質の高いプログラムということになります。

|  |
| --- |
| 例題  プログラミングの標準化に関する記述のうち，適切なものはどれか。  ア　プログラマの個性を否定することが目的ではなく，コンパイラによる最適化の効果を得やすくすることが本来の目的である。  イ　プログラミングに関する規約を設けることによって，プログラマの犯しやすい誤りを未然に防止する効果がある。  ウ　プログラム言語に依存しない共通事項についての規程を定めることが目的である。  エ　プログラムの標準的な実行時間を明確にし，効率の良いプログラムの作成を促進する効果がある。  プログラミングに関する基準を設けることで、担当者スキルのバラつきをおさえ、犯しやすい誤りを未然に防ぐことやプログラムの一貫性を保つことにつながります。結果として、理解しやすいプログラムを開発することができ、開発容易性や保守容易性という観点から品質の高いシステム作りが可能となります。  基本情報　平成20年度春　問47　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-895

## 3. デバッグ

学習のポイント

✅ 登場する用語の意味を覚えよう！

作成したプログラムに潜む誤りを発見し、修正する作業をデバッグ（debug）と呼びます。

デバッグの方法には、プログラムリストや流れ図をもとにプログラムの論理を読んで検査し、誤りを検出して修正する**机上デバッグ**や、デバッグツールを使用する方法があります。

デバッグツールには、静的テストツール（静的解析ツール）と動的テストツールがあります。

**静的テストツール**は、プログラム言語で記述されたプログラムリスト（ソースコード）を解析して、プログラムの誤りを検出します。

**動的テストツール**は、プログラムを実行して、実行結果から誤りの検出や品質の評価を行います。代表的な動的テストツールには、プログラムの実行過程に沿ってメモリやレジスタの内容を書き出す**トレーサ**、特定の時点で成立する条件をプログラム中に埋め込み予定通りプログラムが動作するか確認する**アサーションチェッカ**、テストの実行結果を蓄積して、網羅率を表示する**カバレージモニタ**などがあります。

|  |
| --- |
| 例題  静的テストツールの機能に分類されるものはどれか。  ア　ソースコードを解析して，プログラムの誤りを検出する。  イ　テスト対象モジュールに必要なドライバ又はスタブを生成する。  ウ　テストによって実行した経路から網羅度を算出する。  エ　プログラムの特定の経路をテストするためのデータを生成する。  プログラムを実行せずにプログラムの不具合を発見するツールが静的テストツールです。これに対して、プログラムを実行してテストするツールを動的テストツールと呼びます。  正解として選ばれなかった選択肢は、動的テストツールの機能です。  基本情報　平成28年度春　問21　[出題頻度：★★★]  解答－ア |

別冊演習ドリル 》 1-896～899

## 4. ソフトウェアユニットのテスト

学習のポイント

✅ 「バグ曲線」は問題を解きながら要点を整理しよう！

✅ ブラックボックステストとホワイトボックステストの違いを覚えよう！

✅ ブラックボックステスト、ホワイトボックステストのテストデータの作成方法の種類を覚えよう！

### １）テストの目的

ソフトウェアユニットのテスト（**単体テスト**）は、ソフトウェア詳細設計で定義したテスト仕様に従って行い、ソフトウェア詳細設計書に記載された要求事項を満たしているか確認します。

|  |
| --- |
| 例題  プログラムのテストの目的として，最も重要なものはどれか。  ア　バグがないことを示すこと イ　バグの原因を究明すること  ウ　バグを修正すること エ　バグを見つけること  プログラムを作成したら、要求事項を満たしているか確認します。プログラムにバグがあると、要求どおりに動作しないので、バグを見つけることが重要です。なお、選択肢ウの「バグを修正すること」をデバッグと呼びます。  基本情報　平成13年度秋　問51　[出題頻度：★☆☆]  解答－エ |

別冊演習ドリル 》 1-900

### ２）テストの実施と評価

品質の良いプログラムを作成するためには、さまざまなテスト手法に基づいてプログラムのテストを行うとともに、テストの状況を管理することも必要です。

#### ①エラー埋込法（バグ埋込法）

エラー埋込法は、開発者が意図的にプログラム中にエラー（バグ）を埋め込んでおき、デバッグさせて、発見されたエラーのうちの意図的に埋め込んだエラーの数の比率から全体のエラーの数を推定する方法です。

エラー埋込法では、次の式により全体の潜在エラー数（真のエラー数）を推定できます。

＝

🏋プラスアルファ

その後、エラー埋込み法は、２つの独立したテストグループが同一のテストを行い、発見されたエラーの数から全体の潜在エラー数を推定する方法に改善されました。この方法を二段階エディット法といいます。

**二段階エディット法**では、２つの独立したテストグループをA、Bとし、それぞれNA個、NB個のエラーを発見し、このうちNAB個が共通するエラーであったときの全体の潜在エラー数を次の式で推定します。

N＝NA×NB／NAB

|  |
| --- |
| 例題  バグ埋込み法によってソフトウェア内に残存するバグを推定する。テストによって現在までに発見されたバグ数は48個であり，総埋込みバグ22個のうち，テストによって発見されたものは16個であった。あと幾つのバグが潜在していると推定されるか。ここで，埋込みバグの発見数とソフトウェアのバグの発見数は比例するものとする。  ア　６ イ　10 ウ　18 エ　22  バグ埋込み法では、次の式により潜在バグ数を推定することができます。  ＝  真のバグ数をXとおくと、  ＝  X＝66  となります。したがって、残りの真のバグは66－48＝18個と推定されます。  基本情報　平成20年度秋　問43　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-901,902

②**バグ曲線**

実施しているテストの品質を管理する目的で使われるグラフに、**バグ曲線**（**バグ管理図**、**信頼度成長曲線**）があります。バグ曲線は、横軸にテスト項目消化件数（テスト時間）、縦軸に累積バグ件数（エラー累積数）をとって描いた曲線です。

テストが適正な場合、開始当初は多数のバグが発見されますが、テストが進むにつれてシステムの品質も安定し、発見されるバグの数も減ることから、Ｓ字型を描くという特徴があります。そのため、いつまでも開始当初と同じようにバグが増え続けたり、予想よりも早く発見されるバグの数が収束したりして、Ｓ字曲線を描かない場合には何らかの問題があると考えられます。

テスト項目消化件数

累積バグ件数

|  |
| --- |
| 例題  プログラムテストの管理図において，図のようにすべての線が横ばい状態になった。この状況から推測できることとして，適切なものはどれか。  時間  テスト項目数  検出バグ数  未消化テスト項目数  未解決バグ数  バグ数  ア　解決困難なバグに直面しており，その後のテストが進んでいない。  イ　テスト項目の消化実績が上がっており，バグの発生がなくなった。  ウ　バグが多発し，テスト項目の消化実績が上がらなくなった。  エ　バグ発生とテスト項目消化の比率が一致し，未解決バグがなくなった。  プログラムテストの管理図において、すべての線が横ばいになっているということは、テスト項目の消化、バグの検出、バグの解決のすべてが進まない状態にあるということです。つまり、テスト全体にかかわる致命的なバグの発見により、テスト作業が中断しており、また、そのバグも解決されていないという状況であると考えられます。  イ　テスト項目の消化実績が上がれば、未消化テスト項目は減少を続けます。  ウ　検出バグ数のグラフは、バグが多発しているとはいえません。正常な曲線です。  エ　未消化バグ数はゼロになっていません。  基本情報　平成19年度秋　問47　[出題頻度：★★☆]  解答－ア |

別冊演習ドリル 》 1-903～905

### ３）テストの手法

代表的なテストの手法に、ブラックボックステストとホワイトボックステストの２つがあります。

#### ①ブラックボックステスト

ブラックボックステストは、プログラムの内部構造は全く考えず、入力データと出力結果だけを見て、機能と性能が要求どおりになっているかどうかを検証する方法です。

ユーザ部門の立場から見た機能のテストに向いていますが、外部仕様にないプログラム内部の特殊な処理はテストできないという欠点をもっています。

|  |
| --- |
| 例題  モジュールの内部構造を考慮することなく，仕様書どおりに機能するかどうかをテストする手法はどれか。  ア　トップダウンテスト イ　ブラックボックステスト  ウ　ボトムアップテスト エ　ホワイトボックステスト  ア　トップダウンテストとは、結合テストの１つで、最上位モジュールから順次、下位モジュールを結合しながらテストする方法です。下位モジュールをシミュレートするプログラム（スタブ）が必要となります。  ウ　ボトムアップテストとは、結合テストの１つで、最下位モジュールから順次、上位モジュールを結合しながらテストする方法です。上位モジュールをシミュレートするプログラム（ドライバ）が必要となります。  エ　ホワイトボックステストとは、モジュールの内部構造および論理、制御構造の流れに着目したテスト方法です。  基本情報　令和4年度サンプル　問40　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-906～908

ブラックボックステストにおけるテストデータの作成には、次のような方法があります。

**ⅰ）同値分割**

同値分割は、テスト対象となるデータの取り得る値を調べ、同じ意味をもつ範囲を１つのクラスとしていくつかのクラスに分割し、各クラスからテストデータとなる値を１つ以上選ぶ方法です。各データがとることのできる正しい範囲を**有効同値クラス**、無効な誤った範囲を**無効同値クラス**といいます。

例　100点満点で実施したテストの得点を調べるためのテストデータ

無効同値クラス

…,－2,－1

101,102,…

0,1,2,3, …………… ,99,100

有効同値クラス

無効同値クラス

０未満の値，０～100の中の値，101以上の値の３つをテストデータとして用意する。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  表は，あるプログラムの入力データを，有効同値クラスと無効同値クラスに分けたものである。同値分割法によってテストケースを設計する場合，最小限のテストデータの組合せとして，適切なものはどれか。   |  |  | | --- | --- | | 同値クラス | データ | | 無効同値クラス | －２，－１，０ | | 有効同値クラス | １，２，３，４，５ | | 無効同値クラス | ６，７，８ |   ア　－２，０，１，５，６，８ イ　０，１，５，６  ウ　－１，３，６ エ　１，５  同値分割法とは、ブラックボックステストにおけるテストケース設計法の1つであり、入力データを有効な範囲（有効同値クラス）と無効な範囲（無効同値クラス）に分割し、それぞれの範囲から代表値を抽出する方法です。  基本情報　平成18年度秋　問43　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-909,910

**ⅱ）限界値分析**

限界値分析は、テスト対象となるデータを同値クラスに分割し、各クラスの境目の両どなりの値（境界値）をテストデータとして選ぶ方法です。

例　100点満点で実施したテストの得点を調べるためのテストデータ

無効同値クラス

…,－2,－1

101,102,…

0,1,2,3, …………… ,99,100

有効同値クラス

無効同値クラス

テストデータとして選択する値

－１，０，100，101の４つをテストデータとして用意する。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 例題  入力項目“年齢”（整数値）の正常データ範囲が15≦年齢≦60であるとき，限界値分析に用いるテストデータの組合せとして，適切なものはどれか。  ア　14，15，60，61  イ　14，61  ウ　15，16，59，60  エ　15，60  ブラックボックステストの1つである限界値分析では、境界値とその1つ外側のデータでテストを行います。  この問題では、データ範囲として、「15以上60以下」とあるので、15とその1つ外側の14、60とその1つ外側の61をテストデータとして用意する必要があります。   |  |  |  | | --- | --- | --- | | ←無効同値クラス→ | ←　　　　有効同値クラス　　　　→ | ←無効同値クラス→ | | ････ , 12 , 13 , 14 | 15 , 16 , 17 ･･･ , 58 , 59 , 60 | 61 , 62 , 63 ････ |   初級システムアドミニストレータ　平成15年度秋　問37　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-911～913

🏋プラスアルファ

**●原因－結果グラフ**

原因－結果グラフは、テスト対象のデータが、明確にクラス分けできないようなときに有効な方法です。入力（原因）と出力（結果）の関係をグラフで表現し、デシジョンテーブルを作成してテスト項目を設計します。

#### ②ホワイトボックステスト

ホワイトボックステストは、プログラムの内部構造や論理を詳細に調べて検証する方法です。

プログラマの立場から見た詳細な機能をテストすることはできますが、仕様にありながらプログラムで実現されていない機能は発見できない場合がある、という欠点をもっています。

|  |
| --- |
| 例題  ホワイトボックステストの説明として，適切なものはどれか。  ア　外部仕様に基づいてテストデータを作成する。  イ　同値分割の技法を使用してテストデータを作成する。  ウ　内部構造に基づいてテストデータを作成する。  エ　入力と出力の関係からテストデータを作成する。  テスト法には、内部仕様に基づいてテストするホワイトボックステストと、外部仕様に基づいてテストするブラックボックステストがあります。ホワイトボックステストのテストデータ作成ではテストの網羅性が基準となり、命令網羅・判定条件網羅・条件網羅・複数条件網羅などの方法があります。ブラックボックステストのテストデータ作成では、同値分割・限界値分析・原因－結果グラフなどの方法があります。  基本情報　平成23年度春　問49　[出題頻度：★★☆]  解答－ウ |

別冊演習ドリル 》 1-914,915

ホワイトボックステストのテストデータの作成方法には、次に示すような方法があります。

**ⅰ）命令網羅**

命令網羅は、プログラム中の全ての命令を、最低１回は実行させるようにテストケースを設定する方法です。命令網羅では、判定条件は考慮しません。

判定条件

命令

偽

真

**ⅱ）判定条件網羅（分岐網羅）**

判定条件網羅は、真と偽を少なくとも１回以上実行するようにテストケースを設定する方法です。

判定条件

Ａ

命令

真

偽

①

②

|  |  |  |
| --- | --- | --- |
|  | ① | ② |
| Ａ | 真 | 偽 |

**ⅲ）条件網羅**

条件網羅は、判定条件の真と偽についてそれぞれの組合せを満たし、かつ少なくとも1回は実行するようにテストケースを設定する方法です。

判定条件

Ａ又はＢ

命令

真

偽

①,②

|  |  |  |
| --- | --- | --- |
|  | ① | ② |
| Ａ | 真 | 偽 |
| Ｂ | 偽 | 真 |

**ⅳ）判定条件／条件網羅**

条件網羅では、複合条件の個々のＡ、Ｂについて、それぞれ真と偽の検証ができますが、全体の判定条件としてはともに真となり、偽の経路の検証ができません。そこで、条件網羅に判定条件網羅を組合せたものが、判定条件／条件網羅です。

判定条件

Ａ又はＢ

命令

真

偽

①,②

③

|  |  |  |  |
| --- | --- | --- | --- |
|  | ① | ② | ③ |
| Ａ | 真 | 偽 | 偽 |
| Ｂ | 偽 | 真 | 偽 |

判定条件

Ａ又はＢ

**ⅴ）複数条件網羅（分岐条件網羅）**

複数条件網羅は、全ての起こり得る真と偽の組合せと、それに伴う分岐を網羅するようにテストケースを設定する方法です。

命令

真

偽

①,②,③

④

判定条件

Ａ又はＢ

判定条件

Ａ又はＢ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ① | ② | ③ | ④ |
| Ａ | 真 | 真 | 偽 | 偽 |
| Ｂ | 真 | 偽 | 真 | 偽 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  プログラムの流れ図で示される部分に関するテストデータを，判定条件網羅（decision coverage）によって設定した。このテストデータを複数条件網羅（multiple condition coverage）による設定に変更するとき，加えるべきテストデータのうち，適切なものはどれか。ここで，（ ）で囲んだ部分は，一組みのテストデータを表すものとする。  Yes  処理１  処理２  Ａ>６ or Ｂ=０  No  ・判定条件網羅によるテストデータ  （Ａ＝４，Ｂ＝１），（Ａ＝５，Ｂ＝０）  ア　（Ａ＝３，Ｂ＝０），（Ａ＝７，Ｂ＝２）  イ　（Ａ＝３，Ｂ＝２），（Ａ＝８，Ｂ＝０）  ウ　（Ａ＝４，Ｂ＝０），（Ａ＝８，Ｂ＝０）  エ　（Ａ＝７，Ｂ＝０），（Ａ＝８，Ｂ＝２）  複数条件網羅は、条件のすべての起こり得る真と偽の組合せと、それに伴う分岐を網羅するようにテストケースを設定する方法です。  すべての起こり得る真と偽の組合せは、次の①～④になります。  ここで③、④は判定条件網羅によるテストデータとして挙げられているため、①、②に該当する組合せを選択肢より選びます。   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | ① | ② | ③ | ④ | | Ａ＞６ | 真（Ａ＝７） | 真（Ａ＝８） | 偽（Ａ＝５） | 偽（Ａ＝４） | | Ｂ＝０ | 真（Ｂ＝０） | 偽（Ｂ＝２） | 真（Ｂ＝０） | 偽（Ｂ＝１） | | 分岐先 | Yes | Yes | Yes | No |   基本情報　平成30年度春　問49　[出題頻度：★★★]  解答－エ |

別冊演習ドリル 》 1-916～921

# 6. ソフトウェア結合・ソフトウェア適格性確認テスト

## 1. ソフトウェア結合テスト

学習のポイント

✅ トップダウンテストの特徴と、スタブを使用することを覚えよう！

✅ ボトムアップテストの特徴と、ドライバを使用することを覚えよう！

ソフトウェア結合テストは、ソフトウェア方式設計で定義したテスト仕様に従い、ソフトウェアユニットのテスト（単体テスト）で確認された個々のモジュールを、２つ以上連結して正しく動作するか確認します。

|  |
| --- |
| 例題  モジュール間やサブシステム間のインタフェースを検証するために行うテストはどれか。  ア　運用テスト イ　結合テスト ウ　システムテスト エ　単体テスト  ア　運用テストは、完成したシステムを導入し、実際の動作環境で要求仕様のとおりに動作するかをテストします。  ウ　システムテスト（総合テスト）は、結合テストの完了したプログラムが実際の動作環境で正しく動作するかをテストします。  エ　単体テストは、モジュールの機能が満たされているか、誤りがないかをテストします。  基本情報　平成18年度秋　問42　[出題頻度：★☆☆]  解答－イ |

ソフトウェア結合テストには次のような方法があります。

### １）トップダウンテスト

トップダウンテストは、最上位モジュールから順に結合しながらテストを重ねていく方法です。ただし、上位モジュールから順に結合しながらテストしていく場合、下位モジュールが完成していないとテストが行えません。そのため**スタブ**と呼ばれる仮の下位モジュール（上位モジュールからの呼び出しに対し無条件に値を返す）を用意する必要があります。下位モジュールが完成したら、スタブと交換し再度テストを行います。

なお、トップダウンテストの特徴は、次のとおりです。

・上位のインタフェースが早い時期にテストできる。

・重要度の高い上位モジュールが繰返し実行されるため、信頼度が高い。

・モジュール単体の機能や個々の論理を十分にテストできないことがある。

・新規開発システムに適用すると効果的である。

Ａ

Ｂ

Ｃ

スタブ

スタブ

テストモジュール

テスト完了

Ａ

Ｂ

Ｃ

Ｄ

Ｅ

テスト完了

テスト完了

テストモジュール

スタブ

スタブ

テストモジュール

Ａ

|  |
| --- |
| 例題  階層構造のモジュール群から成るソフトウェアの結合テストを，上位のモジュールから行う。この場合に使用する，下位のモジュールの代替となるテスト用のモジュールはどれか。  ア　エミュレータ イ　シミュレータ  ウ　スタブ エ　ドライバ  ア　エミュレータは、あるシステム上でほかのOSやCPUの機能を再現し、そのOSやCPU向けのアプリケーションソフトを動作させるものです。通常、ハードウェアで作成されています。  イ　シミュレータは、エミュレータと同様の働きをするものです。エミュレータと異なり、ソフトウェアで作成されています。  エ　ドライバは、ボトムアップテストに使用する上位モジュールの代替となるテスト用のモジュールです。  基本情報　平成28年度秋　問49　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-922,923

### ２）ボトムアップテスト

ボトムアップテストは、最下位モジュールから順に結合しながらテストを重ねていきます。上位のモジュールが未完成の場合は、機能をシミュレートするためのテスト用のモジュールである**ドライバ**が必要です。

なお、ボトムアップテストの特徴は、次のとおりです。

・開発の初期段階から並列作業が可能である。

・モジュール単体の機能や論理のテストを十分に行うことができる。

・最終段階でインタフェース上の問題点が発生しやすい。

・稼働中のシステムを修正する場合に有効である。

ドライバ

Ｂ

Ｃ

Ｄ

Ｅ

テスト完了

テストモジュール

Ｄ

Ｅ

テストモジュール

ドライバ

Ａ

Ｃ

Ｄ

Ｅ

テストモジュール

テスト完了

テスト完了

Ｂ

|  |
| --- |
| 例題  ボトムアップテストにおいて，被テストモジュールの上位モジュールの機能を代行するものはどれか。  ア　シミュレータ イ　スタブ  ウ　デバッガ エ　ドライバ  ア　シミュレータは、他のコンピュータ用のプログラムの命令を解読しながら実行する言語プロセッサで、ＣやBASICなどのプログラム言語を使って作成されます。  イ　スタブは、下位モジュールの機能をシミュレートするテスト用のモジュールです。  ウ　デバッガは、プログラムエラーの検出や修正を効率的に行うために利用するソフトウェアのことです。  初級システムアドミニストレータ　平成19年度春　問32　[出題頻度：★☆☆]  解答－エ |

別冊演習ドリル 》 1-924～926

## 2. ソフトウェア適格性確認テスト

学習のポイント

✅ 各テストの意味を覚えよう！

ソフトウェア適格性確認テストは、ソフトウェア要件定義で定義したとおりにソフトウェア要件が実現されているかを確認します。なお、この段階を狭い意味での**システムテスト**（**総合テスト**）と呼ぶこともあります。

ソフトウェア適格性確認テストには、次のような種類があります。

#### ①機能テスト

機能テストは、ユーザが要求した機能が実現されているかを検証します。

#### ②性能テスト

性能テストは、応答時間やスループットなどの性能を検証します。

#### ③負荷テスト

負荷テストは、量的な負荷を与えて、業務に耐えられるかを検証します。

#### ④リグレッションテスト（退行テスト）

リグレッションテストは、修正した内容が、それまで正常に動作していた他の機能に影響を与えていないかどうかを検証します。

|  |
| --- |
| 例題  ソフトウェアのテストの種類のうち，ソフトウェア保守のために行った変更によって，影響を受けないはずの箇所に影響を及ぼしていないかどうかを確認する目的で行うものはどれか。  ア　運用テスト イ　結合テスト  ウ　システムテスト エ　リグレッションテスト  ア　運用テストは、実際の動作環境で要求どおりの結果が得られるかどうかを検証する最終段階のテストです。このテストを経て初めて運用開始となります。  イ　結合テストは、単体テストを終了したモジュールを組み合わせて行うテストであり、モジュール間のインタフェースに関するエラーを検出することを目的とします。  ウ　システムテストは、結合テストの終了後、ソフトウェアとしての要件が満たされているかどうかを検証するテストであり、総合テストとも呼ばれています。  基本情報　平成26年度秋　問49　[出題頻度：★★★]  解答－エ |

別冊演習ドリル 》 1-927,928

# 7. システム結合・システム適格性確認テスト

## 1. システム結合テスト

学習のポイント

✅ 出題頻度を参考に効率よく学習しよう！

システム結合テストは、システム方式設計で定義したテスト仕様に従って、ソフトウェア、ハードウェア、手作業及び必要に応じてほかのシステムのすべてを結合したシステムが要件を満たしているか確認します。具体的には、「システム要件のテスト網羅性」、「使用されたテスト手法及び標準の適切性」、「期待した結果への適合性」、「システム適格性確認テストの実現可能性」、「運用及び保守の実現可能性」を考慮して評価します。

## 2. システム適格性確認テスト

システム適格性確認テストは、システム要件定義で定義したとおりにシステム要件が実現されているかを確認します。なお、この段階を広い意味での**システムテスト**（**総合テスト**）と呼ぶこともあります。

テストケースは、外部設計の担当者が作成します。その際にテストデータとして、実際に業務で使うデータや、業務上例外として処理されるデータなどを用意します。そこで、この段階からユーザもテストに参加します。

|  |
| --- |
| 例題  システム適格性確認テストを実施するとき，用意しておくべきテストデータはどれか。  ア　実際に業務で使うデータや，業務上例外として処理されるデータ  イ　ソフトウェアユニット間のインタフェースに関するエラーを検出するデータ  ウ　ソフトウェアユニット内の全分岐を１回以上通るデータ  エ　ソフトウェアユニット内の全命令が１回以上実行されるデータ  イ　ソフトウェア結合テストで使用するデータです。  ウ　ソフトウェアユニットテストにおける分岐網羅テストで使用するデータです。  エ　ソフトウェアユニットテストにおける命令網羅テストで使用するデータです。  基本情報　平成21年度秋　問48　[出題頻度：★☆☆]  解答－ア |

# 8. 導入

学習のポイント

✅ 例題や演習ドリルの問題を解きながら、要点を整理して解答力を身につけよう！

システム適格性確認テストが終了すると、発注元へシステムを納入するためのソフトウェア導入計画を立案することになります。

## 1. ソフトウェアの導入計画の作成

ソフトウェアの導入計画を作成するに当たっては、**移行テスト**を行って導入の際に必要となる作業項目や管理指標、作業実行時に発生すると予想される問題点などの各種データを事前に収集し、それをもとに導入先のシステム環境、作業手順、役割分担などを決定します。なお、現システムから新システムへの移行は、サービス停止時間が極力短くなるようにしなければなりません。また同時に、移行作業で問題が発生した場合に旧システムに戻す判断基準をあらかじめ決めておく必要があります。

|  |
| --- |
| 例題  システムの移行テストを実施する主要な目的はどれか。  ア　確実性や効率性の観点で，既存システムから新システムへの切替え手順や切替えに伴う問題点を確認する。  イ　既存システムの実データのコピーを利用して，新システムでも十分な性能が得られることを確認する。  ウ　既存の他システムのプログラムと新たに開発したプログラムとのインタフェースの整合性を確認する。  エ　新システムが，要求された全ての機能を満たしていることを確認する。  イ　システムテスト（ソフトウェア適格性確認テスト）の性能テストに関する記述です。  ウ　結合テスト（ソフトウェア結合テスト）に関する記述です。  エ　システムテスト（ソフトウェア適格性確認テスト）の機能テストに関する記述です。  基本情報　平成31年度春　問56　[出題頻度：★☆☆]  解答－ア |

## 2. ソフトウェア導入の実施

移行テストが完了すると、次に移行対象システムを含めた本稼働環境あるいはそれに準じるテスト環境において、発注者立会いのもと日々の業務の正常な運用に問題がないかの検証（**運用テスト**）が行われます。具体的には、業務処理量の変動にシステムの処理能力が対応できるか、業務手順どおりに運用できるかなどがテストされます。そして運用テストが正常に終了すると、システムが発注先から発注者へ引き渡されます。

|  |
| --- |
| 例題  運用テストにおける検査内容として，適切なものはどれか。  ア　個々のソフトウェアユニットについて，仕様を満足していることを確認する。  イ　ソフトウェア品目の中で使用しているアルゴリズムの妥当性を確認する。  ウ　ソフトウェアユニット間のインタフェースが整合していることを確認する。  エ　利用者に提供するという視点で，システムが要求を満足していることを確認する。  ア、イ　ソフトウェアユニットテストに関する記述です。  ウ　ソフトウェア結合テストに関する記述です。  基本情報　平成24年度秋　問49　[出題頻度：★★☆]  解答－エ |

別冊演習ドリル 》 1-929～931

移行方式には、全社一斉に旧システムから新システムへ切り替える**一斉移行方式**と、支店ごと、もしくは拠点ごとに旧システムから新システムへ切り替える**段階的移行**（順次移行）**方式**の２つがあります。

一斉移行方式を採用した場合には、新旧システムを並行して運用することによる作業の二重負担を避けることができ、大きな経済的効果が期待できますが、一方でシステムの規模が大きいと調整が難しいというリスクも伴います。

なお、段階的移行方式には、システムを分割しサブシステムごとに移行するサブシステム順次移行方式、一部の部門や地域に先行して新システムを導入して影響を確認した後に全体に導入する**パイロット移行方式**、業務別に新システムに移行する業務別順次移行方式などがあります。

|  |
| --- |
| 例題  システムの移行方式の一つである一斉移行方式の特徴として、最も適切なものはどれか。  ア　新旧システム間を接続するアプリケーションが必要となる。  イ　新旧システムを並行させて運用し，ある時点で新システムに移行する。  ウ　新システムへの移行時のトラブルの影響が大きい。  エ　並行して稼働させるための運用コストが発生する。  一斉移行方式は、旧システムをすべて同時に新システムに切り替える方式です。これに対し、旧システムから部分ごとに少しずつ新システムに移行する方式を、段階的移行方式と呼びます。  一斉移行方式を採用した場合には、新旧システムを並行して運用することによる作業の二重負担を避けることができ、大きな経済的効果が期待できますが、一方でシステムの規模が大きいと調整が難しいというリスクも伴います。  なお、その他の選択肢は、段階的移行方式の特徴です。  基本情報　平成26年度春　問55　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-932

# 9. 受入れ支援

学習のポイント

✅ 概要をつかんでおこう！

ソフトウェア適格性確認テスト、システム適格性確認テストの結果を考慮して、受入れの準備、受入れレビュー、テストを行います。なお、受入れの準備、受入れレビュー、テストは取得者が行い、開発者はこれを支援します。

**受入れテスト**では、要求仕様が満たされているか、要求性能が満たされているか、利用者マニュアルは適切か、操作性に問題はないか、運用手順は妥当かなどを検証します。

受入れ用テストが正常に終了すると、システムが発注先から発注元へ納品されます。納入に当たっては、事前に発注元の利用者への教育訓練が行われます。

# 10. 保守・廃棄

学習のポイント

✅ 出題頻度を参考に効率よく学習しよう！

## 1. 保守のタスク

システムの稼働後も、システム又はソフトウェアの不具合によって、業務に支障をきたすことがあります。システム又はソフトウェアの保守は、処理の正確性、効率性、機能の拡張性を向上するために行われるものです。

## 2. 廃棄のタスク

使用を終えたシステム又はソフトウェアは、現行のシステムに影響がないことを確認して、起動不能にしたり、解体したりして廃棄します。

## 3. 保守のタイプ及び形態

保守は対象により、ハードウェア保守とソフトウェア保守に分けられます。

### １）ハードウェア保守

ハードウェアの保守には、実施時期で分類すると、事後保守、予防保守、定期保守の３種類があります。**事後保守**は、障害が発生した後に障害を取り除くために行われる保守で、**予防保守**は、システムの故障や障害が発生するのを未然に防ぐために行われる保守です。また、**定期保守**は、あらかじめ計画され定期的に行われる保守をいいます。予防保守や定期保守を行うことで、故障発生の頻度を低くすることができます。

また実施場所からみると、現地保守と遠隔保守に分けられます。**遠隔保守**は通信回線などを通じて保守を行う方法で、分散システムにおいては修理時間を短縮することができます。

なお、修理時間を短縮する方法として、部品をモジュール化し、特別な工具を使用せずに交換できるようにした**ツールレス保守**が近年増えています。

|  |
| --- |
| 例題  予防保守に関する説明として，適切なものはどれか。  ア　サーバが何らかの原因でハングアップしたとき，自動的にシステムをリセットし，再起動する。  イ　サーバにバッテリーを内蔵しておき，瞬断のような短時間の電圧低下が発生してもサーバがダウンしないようにしておく。  ウ　ハードディスク装置などの自動訂正済みエラーを分析することで，故障の前兆をとらえて部品をあらかじめ交換する。  エ　メモリモジュールをあらかじめ複数挿入しておき，どれかが故障した場合には，そのモジュールを論理的に切り離して起動する。  予防保守とは、障害が発生する前に障害発生の恐れがある部分に対処することです。  正解以外の選択肢はすべて障害に対する復旧対策であり、予防対策ではありません。  初級システムアドミニストレータ　平成13年度春　問47　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-933,934

### ２）ソフトウェア保守の形態

ソフトウェア保守は、ソフトウェアの障害を修正するために行われる**修正保守**（訂正保守）とソフトウェア修正でない改良・改善のための**改良保守**に大別されます。

修正保守には、引渡し後に発見されたソフトウェア製品の問題に対処し、是正するために行われる**是正保守**、本番稼働後の運用時に予想されるトラブルの発生を未然に防ぐために行われる**予防保守**があります。

改良保守には、引渡し後にソフトウェア製品の性能または保守性を改善するために行われる**完全化保守**、業務要件の変更、ハードウェアやOSのアップグレードのような処理環境の変更に対応させるために行う**適応保守**があります。

|  |
| --- |
| 例題  ソフトウェアの“修正保守”に関する説明として，適切なものはどれか。  ア　誤りの修正ではなく，より良いアルゴリズムの採用や出力メッセージの充実など，ソフトウェアの完成度を高めるために実施する。  イ　業務要件の変更や，ハードウェアやOSのアップグレードのような処理環境の変更に対応する。  ウ　本番稼働後の運用時に予想される問題に対して，トラブルを予防する目的で実施する。  エ　要求された機能が達成されていない場合，業務に支障が出ないように機能仕様書との不一致を修正する。  ア　ソフトウェアの完成度を高める目的で行う“完全化保守”に関する記述です。  イ　処理環境の変化に対応した変更を行うための“適応保守”に関する記述です。  ウ　運用時に発生する可能性があるトラブルを、事前に防ぐ目的で行う“予防保守”に関する記述です。  基本情報　平成14年度春　問60　[出題頻度：★☆☆]  解答－エ |

別冊演習ドリル 》 1-935～937

🏋プラスアルファ

**●廃棄**

新たなシステム又はソフトウェアの導入や更新に伴い、不要となったシステム又はソフトウェアは廃棄することになります。廃棄に当たっては、適切な廃棄計画を立案し、利用者へ通知するとともに、新たなシステム又はソフトウェアに関する教育訓練を行う必要があります。