

## 040101開発企画設計解説

### 問01 イ

ウォーターフォールモデルの実施順序に関する問題である。

各工程の内容を整理すると次のようになる。

- ① 基本計画は現状の問題を洗い出し解決策を検討しシステム基本計画書をまとめる。
- ② 外部設計は要求仕様をもとに機能を確定しシステム構成を明確にする。
- ③ 内部設計はシステム構築上必要な機能をプログラムに分割し、プログラム間の処理を明確にする。
- ④ プログラム設計は内部設計書に基づいて各プログラムの構造設計を行う。プログラムの属性を決定し、モジュールに分解し、モジュール間のインタフェースを決める。
- ⑤ プログラミングはモジュール内の詳細処理手順を設計・コーディング、テストする。
- ⑥ テストは結合テストやシステムテストを実施する。

aは要件定義、bは内部設計、cはプログラム作成、dはテスト、eはプログラム設計、fは外部設計である。工程の順序は、要件定義(a)、外部設計(f)、内部設計(b)、プログラム設計(e)、プログラム作成(c)、テスト(d)となる。a、f、b、e、c、dとなり、求める答えはイとなる。

### 問02 エ

ウォーターフォールモデルの特徴に関する問題である。

ウォーターフォールモデルは、システム開発の上流から下流に向かって、分割された各工程の成果物を確認しながら、段階的に開発を進める方式であり、この方式によってシステム開発の一貫性を保証する。各フェーズの目的、作業内容、成果物が明確であり、各工程の完了時に成果物確認やレビューを行いながら作業を進める。求める答えはエとなる。

### 問03 イ

プロトタイプモデルの特徴に関する問題である。

プロトタイプモデルは、ウォーターフォールモデルの難点を解決するために考えられたモデルである。ユーザの要求仕様を開発の早い段階に目に見える試作品として現実化する手法である。ユーザの認識結果を開発者にフィードバックし、認識のずれや曖昧さを取り除くことができ最終段階での食い違いをなくすることができる。

プロタイプモデルの長所・短所

- ① 短期間で開発を完了する。
- ② ユーザの意向を反映しながら、システム開発が進められる。
- ③ 小規模なシステムの開発に適している。
- ④ オンラインシステムやネットワークシステムの開発に適している。
- ⑤ 大規模なシステム開発にはまだ不十分な点がある。
- ⑥ 改良や発展を続けていくと、つぎはぎ的となり、整合性が低く効率の悪いシステムになりやすい。

⑦ 開発者が多方面の知識や技術を要求される。

アはウォーターフォールモデル、イはプロトタイプモデル、ウ、エはスパイラルモデルである。  
求める答えはイとなる。

#### 問04 エ

システム開発モデルに関する問題である。

ウォーターフォールモデルは、要求分析、システム設計、製造、テスト、運用の一連の段階的工程からなる。プロトタイプモデルは、初期段階の試作、ユーザインタフェースの確定、手戻りによるリスクの防止などのキーワードが重要である。スパイラルモデルは、成長型モデルで、開発コストや品質からリスクを評価し、リスクを最小にするモデルである。

aはプロトタイプモデル、bはスパイラルモデル、cはウォーターフォールモデルである。求める答えはエとなる。

#### 問05 イ

データ中心分析・設計技法の特徴に関する問題である。

アは運用保守を効率的に推進するソフトウェアの保守に対する考え方である。

イの対象業務のモデル化とその情報資源の活用がデータ中心分析・設計手法の基本的な考え方で、求める答えはイとなる。

ウはデータへのアクセスを効率的に行うデータベース設計の考え方である。

エはモジュールの独立性を高めソフトウェアの再利用を高める考え方である。

#### 問06 イ

データ中心アプローチの開発手順に関する問題である。

最初にデータベースの構築であり、データベース構築ではデータの正規化が重要テーマである。企業モデルを分析し、データ名称の標準化、データ項目の定義、関連性の分析、データの正規化が行われる。更に、業務処理との対応を検討し、ERモデリングを作成する。

データモデリング→ドメイン／原始オブジェクト分析→カプセル化→応用プログラム設計の手順になる。求める答えはイとなる。

#### 問07 エ

データ中心アプローチに関する問題である。

データ中心アプローチの展開手順

- ① 企業目標の設定では、企業全体を対象にデータを分析する。
- ② 企業活動の分析では、企業目標を達成するために必要な活動を明確にし、ビジネスプロセスとデータクラスの関連を明確にする。
- ③ 情報システムの体系化では、ビジネスプロセスとデータクラスの関連を分析して、情報システム体系を明確にする。
- ④ データ体系の確立では、データクラスをブレイクダウンして、管理対象であるエンティティを識別し、企業データモデルを作成する。
- ⑤ データモデルの作成で、企業データモデルに基づいて個別業務のデータモデルを作成す

る。

⑥ 個別業務システムの開発では、データモデルを前提にして個別業務システムを開発する。

アは、特定業務に関連するデータの体系化が問題である。

イは、機能設計から詳細設計へ進むアプローチが問題である。

ウは、独立性の高い対象業務の分割から実行しているのが問題である。

エのデータとデータ操作を一体化した標準部品を利用してシステムを構成する考え方がデータ中心アプローチである。求める答えはエとなる。

#### 問08 エ

IT投資の事前評価に関する問題である。

事前評価は、企画プロセスで作成されるシステム化構想やシステム化計画に対して実施される評価であり、経営戦略に対する合目的性、投資効果性、実現可能性、開発の合理性、運用の容易性、技術的整合性などの項目の内容の妥当性が評価される。

投資目的に基づいた効果目標を設定し、実施可否判断に必要な情報を上位マネージメントに提供する。

ア、イは中間評価、ウは事後評価、エは事前評価の説明である。求める答えはエとなる。

#### 問09 エ

全体計画立案時の策定される業務モデルに関する問題である。

業務モデルは、企業活動とその活動に必要な情報、その情報の流れ、データの持ち方を構造化したもので、企業経営のあり方をモデル化したものである。

業務モデル作成の手順は、①ビジネスプロセスの定義、②データクラスの定義、③ビジネスプロセスとデータプロセスの関連づけ、などの一連の順序で行われる。この作業が全体計画時に行われる業務モデルの策定作業である。

エのビジネスプロセスとデータクラスを関連づけるが全体計画時に行われる作業である。求める答えはエとなる。

アの基幹系の機能とデータ項目の定義、イの既存システムとデータベースの関係の定義、ウの組織の機能と帳票の関連づけはその後の業務モデルの分析時に行う。

#### 問10 ウ

企画プロセスに関する問題である。

企画プロセスは、経営要求・課題の確認、事業環境・業務環境の調査分析、現行業務・システムの調査分析、情報技術動向の調査分析、対象となる業務の明確化、業務の新全体像や新システムの全体イメージの作成、対象の選定と投資目標の策定、システム化構想の文書化と承認、システム化推進体制の確立などのタスクで構成される。

アは開発プロセスのシステム要件の定義、イは運用プロセスの業務システムの移行、ウは企画プロセスのシステム化構想の立案、エは開発プロセスのソフトウェアテストである。求める答えはウとなる。

### 問11 ア

全体最適化に関する問題である。

システムや組織において、各部分機能の最適を図ることを部分最適、システム・組織の全体の最適を図ることを全体最適という。企業及び企業グループで調達、生産、物流、販売など個々の業務機能のみの生産性を上げる（部分最適）のではなく、業務機能全体の効率や生産性を最適化する（全体最適）ことを考えてその企業や企業グループの収益を最大化することを目指す、などというように使われる。

全体最適化計画の策定はシステム化構想の立案時に検討される。

アの全体業務と情報システムのあるべき姿の明確化はシステム化構想の立案段階に行われる内容であり、業務モデルを定義することによって可能となる。求める答えはアとなる。

イのシステム化に要する期間、開発工数、開発費用の見積もりは、システム化計画の立案で行われる。

ウの情報システムの構成要素の洗い出しは、システム化計画の立案で行われる。

エのユーザマニュアルや運用マニュアルの作成準備は運用プロセスで行われる。

### 問12 イ

ソフトウェアライフサイクルにおける企画プロセスの目的に関する問題である。

企画プロセスの目的は、経営事業の目的、目標を達成するために必要なシステムに関する要求事項の集合とシステム化の方針、および、システム化を実現するための実施計画を得ることである。システムが関与する経営戦略を受けて、システム化構想の立案、システム化計画の立案に関わる活動が行われる。

アの業務のあり方、入出力情報などは要件定義プロセスで行う。

イの経営事業の目的、目標を達成するために必要なシステムに関する経営上のニーズ、システム化の方針、システム化実施計画などは、企画プロセスで行われる。求める答えはイとなる。

ウの必要なシステム機能、システム開発方式などは開発プロセスで行われる。

エの必要なシステムの機能、能力、ライフサイクルなどは開発プロセスで行われる。

### 問13 ア

要件定義で行う作業に関する問題である。

要件定義は、システム化要件を定義することである。次の内容について要件を定義する。

- |                          |                   |
|--------------------------|-------------------|
| ① システム化の目標と範囲            | ② システム化の機能要件・性能要件 |
| ③ 業務処理手順                 | ④ 入出力情報要件         |
| ⑤ 操作要件                   | ⑥ 実行環境要件          |
| ⑦ 周辺インタフェース要件            | ⑧ 開発環境要件          |
| ⑨ 主要データベース・主要データ項目に関する要件 |                   |

アはシステム化要件定義であり、要件定義プロセスで行う。

イはソフトウェアの開発であり、開発プロセスで行う。

ウはシステム化方針の立案であり、企画プロセスで行う。

エはシステムの運用、運用支援であり、運用プロセスで行う。

#### 問14 イ

要件定義の非機能要件に関する問題である。

業務要件を実現するために必要なシステムの機能要件以外の要件を非機能要件という。

非機能要件には、品質要件、技術要件、運用・操作要件、移行要件、付帯要件などが含まれる。

技術要件には、システムの実現方法、システム構成、システム開発方式、開発基準、標準、開発環境などが含まれる。

機能要件は業務要件を実現するために必要なシステム要件であり、業務内容(手順、入出力情報、組織、責任、権限など)、業務特性(ルール、制約など)、業務用語、外部環境と業務の関係・授受する情報(インタフェース)などが含まれる。

ア、ウ、エの内容は機能要件であり、イの内容の開発基準、標準は非機能要件である。求める答えはイとなる。

#### 問15 ア

E-R図に関する問題である。

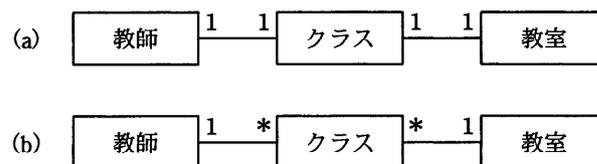
E-R図は、E-Rモデルでユーザ要求定義の分析に使用されるダイアグラムで、データ構造をエンティティ、アトリビュート、エンティティ間の関連という概念で捕らえて分析した上でモデル化する際に、これらの内容を図式化して表現するために用いる。

アのエンティティ間の関連が正しい答えである。求める答えはアとなる。

#### 問16 イ

E-R図に関する問題である。

(a)は、教師は1つのクラスをもち、クラスの使用する教室は1つである。(b)は、教師は複数のクラスをもち、複数のクラスが1つの教室を使用する。



アの(a)の記述は適切だが、(b)では教師と教室を決めてもクラスは決まらない。教師は複数のクラスを担当し、教室は複数のクラスが使用するため。

イの記述内容は適切である。求める答えはイとなる。

ウの(a)の記述は適切であるが、(b)は1つの教室を複数のクラスが使用する。

エの(a)の記述は適切であるが、(b)は一人の教師は複数のクラスを担当する。

#### 問17 ア

DFDに関する問題である。

アのDFDは、システムをデータの流を中心に表現する手法で、データフロー、処理、データストア、外部の四つの記号を使用する。求める答えはアである。

イの状態遷移図は、内部状態の推移関係を図示したものである。各種状態をとる中で一時点

では1個の状態をとり、ある事象が発生すると状態は遷移する。

ウの流れ図は、処理手順を図形で表示するもので、処理手順に従って記述されているため処理手順の誤りなどを容易に見出せる、プログラムの経験がなくても使用することが可能なことやファイルやデータの受け渡しが明確になるなどの特徴がある。

エのペトリネットは、事象応答分析に基づいた要求モデルを表現するもので、並列的に動作する機能間同士の同期を表現することが可能なために、制御中心のシステムの要求分析に最適である。ノードと有向矢印、場所・ノードに記入された印によって、作用の推移状況や事象同士の同期のタイミング等を表すことができる。

#### 問18 エ

D F Dに関する問題である。

D F Dは業務処理におけるデータの流れを図的に表現したものである。データの源泉・吸収、データフロー、データ蓄積、処理の記号からなる。

D F Dの特徴は次の通りである。

- ① 理解しやすい。
- ② 階層化が可能である。
- ③ データ中心である。
- ④ 分析から設計まで使用可能である。

アはプロセスフロー図または業務処理フロー図、イは状態遷移図、E-Rダイアグラム、エはデータフローダイアグラム(D F D)である。求める答はエとなる。

#### 問19 イ

D F Dの階層的表現に関する問題である。

D F Dを階層的に表現する場合の留意点

- ① 機能を下位に展開するとき、入出力フローに矛盾が発生しないようにする。
- ② 上位機能の入出力データフローやデータストアは、下位に展開したときも、必ず、表れなければならない。
- ③ 下位レベルの中だけに表れるデータフローやデータストアは、上位レベルのD F Dには、記入する必要はない。
- ④ 各機能の下位への展開は機能ごとに検討する。

本図の1のバブルには2本の入力データフローと2本の出力データフローがある。

アの場合は、入力データフローが1本しかなく不足しており、適切でない。

イは入力データフローが2本、出力データフローが2本、バブル1の機能のみが詳細化されているため適切な内容である。求める答えはイとなる。

ウのD F Dは、直下のレベルのD F Dに、上位のバブル1とバブル2の2つのプロセスが含まれており、下位への展開は機能ごとに展開する項目に反する。

エのD F Dのバブル1-2の出力データフローはあるが、入力データフローがない。入力データフローがないと処理ができない。

#### 問20 エ

状態遷移図に関する問題である。

状態遷移図は、時間の経過や状況の変化に基づいて、その時々動作や状態を記述するもので

あるから、室内の温度を計測するセンサからの情報に基づいて、室内の環境を最適に保つ温室制御を表すために用いられる。求める答えはエとなる。

#### 問21 エ

オブジェクト指向の基本概念に関する問題である。

抽象化は、いくつかのクラスの共通的な性質に着目し、その共通性のみを残し、その他の細かい相違点を無視することによってより一般化したクラスを定義する汎化のような考え方である。抽象化の反対は具現化である。カプセル化は、データと手続きを一体化し、オブジェクトが持っている情報を外部から隠し、決められた方法でしかアクセスできないようにすることである。これによって、個々のオブジェクトの独立性を高め、再利用や保守も容易になる。継承は、既存のクラスの機能や性質を受け継いだ新しいクラスを作成することである。クラスは同じ性質を持つオブジェクトの集まりがである。

アは、仮想化、構造化、投影が異なる。

イは、構造化、連続が異なる。

ウは、正規化、分割が異なる。

エの抽象化、カプセル化、継承、クラスが基本概念を示す。求める答えはエとなる。

#### 問22 エ

オブジェクト指向に関する問題である。

アのオブジェクトの設定は、クラスを利用して特定のオブジェクトを設定することができる。操作はインタフェースを介して行うメッセージであり、あらかじめ指定する必要がない。

イのカプセル化は、データと振る舞いを一体化してオブジェクトを形成することである。カプセル化によってオブジェクトの独立性を高めることができる。相互依存性を高めるは誤りである。

ウのクラスの変更は関係する上位のクラスを変更するかまたは差分プログラムで特定の内容を変更するかのいずれかで可能である。上位にあるすべてのクラスの変更が必要は誤りである。

エのモデルの拡張や変更は、差分プログラムを利用して可能であり、局所化できる。求める答えはエとなる。

#### 問23 ウ

オブジェクト指向に関する問題である。

オブジェクトには存在や状態を表す属性と振舞いを表すメソッドが存在する。属性とメソッドを一体化したカプセル化されたオブジェクトを使用してオブジェクトモデルを作成する。属性はオブジェクトに格納されており、属性やメソッドはクラスで定義されている。各オブジェクトの属性に対しては、メソッドからしかアクセスできない。オブジェクトの内部に属性は隠され、外部に対してはメソッドだけを公開する。外部からは、属性が隠された状態となり、それをアクセスするためのメソッドだけが見えることになる。これがオブジェクトモデルの特徴である。求める答えはウとなる。

#### 問24 ア

カプセル化の効果に関する問題である。

オブジェクトの状態や振舞いをカプセル化して、ブラックボックス化して隠すことを情報隠蔽という。各オブジェクトの属性に対しては、メソッドからしかアクセスできない。オブジェクトの内部に属性は隠され、外部に対してはメソッドだけを公開する。外部からは、属性が隠された状態となり、それをアクセスするためのメソッドだけが見えることになる。カプセル化によって、オブジェクトの独立性が高まり、再利用がし易くなる。

アはカプセル化の効果、イはユーザ定義型の追加や差分プログラムによる仕様の拡張、ウは継承、エは多様性である。求める答えはアとなる。

#### 問25 ウ

オブジェクト指向の汎化・特化に関する問題である。

スーパークラスとサブクラスの関係は汎化と特化の関係である。

アの会社と社員は所有の関係である。

イの自動車とエンジンの関係は集約関係である。

ウの図形と三角形はスーパークラスが図形であり、サブクラスが三角形である。スーパークラスとサブクラスの関係で、求める答えはウとなる。

エのデータベースとウィンドウは特に関係というものが存在しない。

#### 問26 ウ

オブジェクト指向の集約化に関する問題である。

集約化は、オブジェクト同士の関連のうち全体と部分を表す関係である。車体は自動車の部品であり、タイヤも自動車の部品である。自動車は、駆動装置、車体、車輪から構成されており、このうち1つでも欠けると自動車とは言わない。

ア、イは所有の関係、ウは集約化、エは汎化である。求める答えはウとなる。

#### 問27 エ

オブジェクト指向の振る舞いに関する問題である。

オブジェクトには存在や状態を表す属性と振舞いを表すメソッドが存在する。属性とメソッドを一体にしたものがオブジェクトである。属性とメソッドを一緒にすることをカプセル化と呼ぶ。属性はオブジェクトに格納されており、属性やメソッドはクラスで定義されている。

オブジェクトはその内部データを持つとともに、外部からの動作指示によって、その振舞いを実行するという機構を持つ。外部からの動作指示はメッセージによって与えられる。メッセージは、あるオブジェクトに対してその振舞いを動作させるための依頼の単位であり、オブジェクト間のメッセージのやり取りの結果としてメソッドが実行される。

アのインスタンスは、クラスに属するオブジェクトを具現化したものである。クラスの属性に具体的なデータが与えられたオブジェクトがインスタンスである。

イのクラスは、オブジェクトのうち、共通する性質を持つもの同士をまとめて、新たに名前を付けたものをクラスと呼ぶ。クラスによってオブジェクトを抽象化する。

ウの属性は、オブジェクトの内部的な状態を決定するための情報である。一つのクラスは、一つ以上の属性から構成される。

エのメソッドは、オブジェクトのもつ振舞いを記述したものがメソッドである。求める答え

はエとなる。

### 問28 ウ

オブジェクト指向のUMLに関する問題である。

UMLはオブジェクト指向分析で用いられるモデリング言語で、クラス図、オブジェクト図、状態チャート図、ユースケース図などがある。

アのオブジェクト指向データベースを操作するための言語はSQLがある。

イのオブジェクト指向言語としては、JavaやC++などの言語がある。

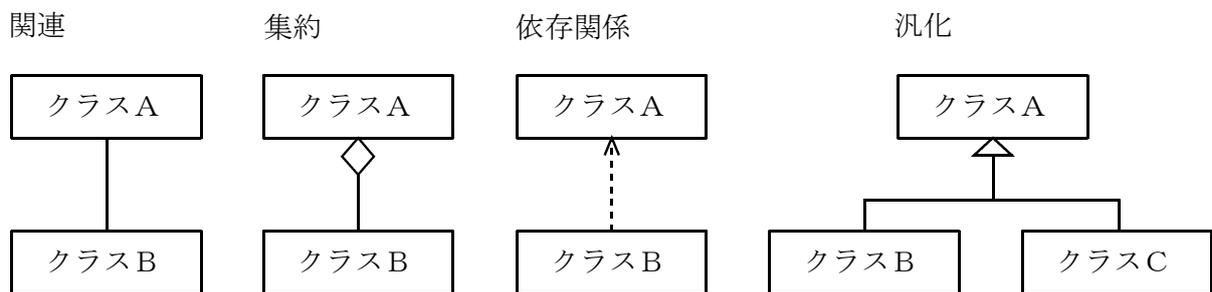
ウのモデルを記述するための言語はUMLの内容を表わしている。求める答えはウとなる。

エはクラス・ライブラリである。

### 問29 イ

UMLに関する問題である。

クラス図は、システム内に存在するクラス同士の静的な構造を記述する。パッケージ単位の表現、システム全体での表現、機能単位での表現など様々な視点で作成することができる。クラスは長方形の中にクラス名、属性、操作が記述される。構成する要素には、汎化、集約、依存関係、多重度など各種の表現が可能である。関連、集約、依存関係、汎化の図的表現法を示すと図のようになる。



集約の関係を表すのは、クラスAとクラスCである。求める答えはイとなる。

### 問30 イ

クラス図に関する問題である。

クラスは、オブジェクト指向プログラミングにおいて、データとその操作手順であるメソッドをまとめたオブジェクトの雛型を定義したもので、これを定義することによって、同種のオブジェクトをまとめて扱うことができるようになる。クラス図は、システム内に存在し、システムを構成するクラス同士の静的な構造を記述したものであり、長方形の中にクラス名、属性、操作などが記述される。

アの関連は、2つの実体間の関係を捉えたものであり、その関連に名前を付けたのが関連名である。

イのクラス名は、クラスに付けた名前、クラス図の最も簡単な表記法は、長方形の中にクラス名を記入する。求める答えはイとなる。

ウの集約は、オブジェクトが複数のオブジェクトで構成されるとき、そのオブジェクトが上位

のオブジェクトの構成要素となる関係である。この集約に名を付けたのが集約名である。

エのユースケースは、外部から見た、システムが提供する機能を定義したもので、それに名前を付けたのがユースケース名である。

### 問31 イ

UMLの多重度に関する問題である。

UML、制約条件から考えて、プロジェクト、社員、プロジェクト内役割分担の3条件で一意に定まり、1社員が1プロジェクトに複数の役割分担で参画し、社員とプロジェクト参画は0個以上の1対多の対応関係になることも可能である。

アの複数のプロジェクトに参画する場合は同じ役割分担になるとは言えない。

イの同じ社員が同じプロジェクトに異なる役割分担で参画することができる。求める答えはイとなる。

ウの社員が1つ以上のプロジェクトに参画しているとは言えない。

エの社員は1部門にのみ所属する。複数の部門には所属しない。

### 問32 ウ

オブジェクト指向設計の手順の問題である。

オブジェクト指向設計の手順

- ① 業務プロセスを検討し、明確化して業務モデルを作成する。
- ② 業務モデルに基づいて、オブジェクトの識別、オブジェクトのデータ属性とメソッドの識別、オブジェクト間の基本対応の分析、メッセージの分析、オブジェクトモデルの作成の順序でオブジェクトのモデリングを行う。
- ③ クラスやオブジェクトを作成するカプセル化の業務になる。オブジェクトの必要な属性を設定し、オブジェクトが必要とするメソッドを決定し、属性とメソッドを組み合わせるカプセル設計を行う。
- ④ オブジェクト間に必要なメッセージやオブジェクトのインタフェースが検討される。必要なオブジェクトが決定されると、これらのオブジェクトを利用して業務を処理する制御設計を行うことになる。メッセージのやり取りやその間に必要な各種ルールが検討されることになる。

設計順序は、業務モデリング→オブジェクトモデリング→カプセル設計→制御設計の順序になる。c→a→b→dとなり、求める答えはウとなる。

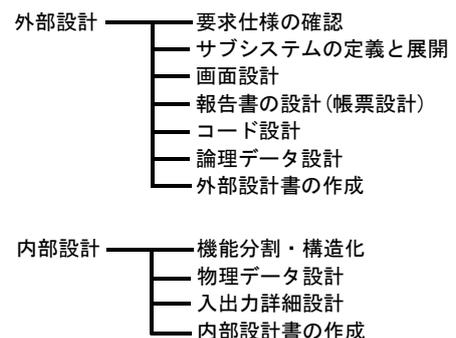
### 問33 イ

外部設計・内部設計の作業に関する問題である。

右の図に、外部設計、内部設計工程で実施される主要な作業内容を示す。

アのシステムをプログラムに分割するのは内部設計であり、DFDは両者の工程で使用する。

イの外部設計で論理データ構造、内部設計で物理データ構造を決定する記述は適切である。求める答えはイと



なる。

ウの遂行内容・順序は、外部設計は基本計画のユーザ要求、内部設計は外部設計の結果に基づいて決定される。

エの外部設計はユーザ側から見た設計であり、内部設計はコンピュータ側から見た設計である。

### 問34 イ

内部設計に関する問題である。

外部設計の成果物に基づいて、システム開発者の立場から進める設計作業であるから、内部設計作業である。アの画面フロー設計、ウのコード設計、エの論理データ設計は外部設計の作業であり、イの機能分割・構造化が内部設計の作業である。求める答えはイとなる。

### 問35 エ

プログラム間のインタフェースの決定に関する問題である。

アの業務フローは、業務処理の流れを示すもので、インタフェースには伝票・帳票を用いる。

イのシステムフローは、業務の流れをコンピュータシステム化した場合の処理の流れを表すもので、インタフェースには帳票類に相当する情報が利用される。

ウのプロセスフローは、業務処理の流れをプログラムの処理順序で表したもので、ファイルやデータの受け渡しを明らかにする。インタフェースには媒体が利用される。

エのデータフローは、データ処理の流れを表現したもので、源泉・吸収や処理プロセス、データの保存、取り扱うデータの関係が表現され、インタフェースはデータになる。

プログラム間のインタフェースの決定は、プログラム間でのデータの受渡を明確にすることであり、データフローを明確にすることである。求める答えはエとなる。

### 問36 エ

チェックディジット法に関する問題である。

数値データ 2 1 3 1 に付加する検査文字を求める。

$$2 \times 4 + 1 \times 3 + 3 \times 2 + 1 = 18$$

$$18 \div 11 = 1 \dots 7$$

検査文字は7となる。求める答えはエとなる。

### 問37 ウ

コード設計のチェックディジット法に関する問題である。

チェックディジットは、コード入力誤りをチェックするために、コードに付加する検査文字である。検査用文字は、コードの各文字を所定のアルゴリズムで計算した英数字が使われる。入力されたコードから計算されたチェックディジットとコードに付加されたチェックディジットを照合してエラーを検出する。

アのコードの実際の付番はコードの処理方法に詳しいシステム利用者が行う方がよい。システム設計者は必ずしも詳しいとは言えない。

イのコードの属性と桁数はコードの使用条件に従って設計すべきである。コンピュータの内部処理効率ではない。

ウのコードの入力ミスを防止するためには、検査文字を採用すべきである内容は適切な記述である。求める答えはウとなる。

エのコードの保守方法の検討は外部設計段階で決めるべきである。運用テスト段階では遅い。

### 問38 エ

注文日の論理チェックに関する問題である。

アのシーケンスチェックは、データの順序が決められた通りかどうかを検査する。

イの重複チェックは、許されない重複データがあるかどうかをチェックすることである。

ウのフォーマットチェックは、データの各項目の桁位置、けた数が決められた形式で入力されているかどうかをチェックすることである。

エの論理チェックは、項目の内容が決められている値のいずれかに該当するかを検査する。例えば、昭和の年数は1～63の範囲内にあるかどうかをチェックする。注文日が入力日以前の営業日かを検査するのは論理チェックである。求める答えはエとなる。

### 問39 ウ

データの誤りチェックのリミットチェックに関する問題である。

リミットチェックは、限度検査、範囲検査のことで、データの値が指定された値を超えていないかどうかや、指定された範囲内にあるかどうかを検査する方法である。

アは重複チェック、イは論理チェック、ウはリミットチェック、エは照合チェックである。求める答えはウとなる。

### 問40 ア

入力漏れのチェックに関する問題である。

アの定められた項目数と入力された項目数の比較は入力漏れのチェックになる。求める答えはアとなる。

イの入力された項目のデータ形式の検査は入力内容のチェックであって、入力漏れのチェックではない。

ウのデータ形式の確認は入力漏れのチェックにはならない。

エのマスタファイルの突き合わせは、内容のチェックであって、入力漏れのチェックではない。

### 問41 エ

マスタファイルの利用に関する問題である。

顧客マスタファイル、商品マスタファイル、担当者マスタファイル、当月受注ファイルを使用して、当月と前3ヶ月の受注実績を求める問題である。

ア、イは、前3ヶ月の顧客と商品に関連づけるデータがないため不能である。

ウは、商品と担当者を関連づけるデータがないため不能である。

エは顧客コードから担当者コードを決定することが可能であり、当月と前3ヶ月の担当者別の顧客別受注実績を顧客マスタ、担当者マスタ、当月受注ファイルから作成可能である。求める答えはエである。

#### 問42 エ

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。

#### 問43 ウ

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。

#### 問44 ウ

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、データの流に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、データの流に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、データの流にに着目した分割技法で、データの流に岐がある場合に用いる。

#### 問45 ウ

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、繰り返してデータ構造を表現する。

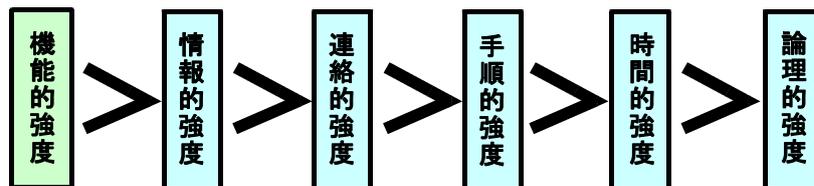
階層構造図では、基本給の更新、手当の更新、控除の更新とモジュールが処理機能別に分けられている。これはトランザクション分割である。求める答えはウとなる。

#### 問46 ア

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、情動的強度、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。

#### 問47 ウ

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 하나가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 情動的強度は、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、情動的強度、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。

#### 問48 イ

モジュール設計の独立性に関する問題である。

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。

#### 問49 ウ

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。

#### 問50 ウ

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。

#### 問51 ア

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。