

開発企画設計演習解説

問1 イ

ウォーターフォール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、基本計画からテストまで、川の流れのように一定方向に作業を進めるのを原則としている開発モデルである。システム開発を基本計画、外部設計、内部設計、プログラム設計、プログラミング、テストの工程順に進め、後戻りせずに開発を進める方式である。開発工程を設計、実装、テストなどに分け、前の工程が完了してから、その成果物を使用して次の工程を行う開発法である。

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

問2 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3 イ

ウォーターフォール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、基本計画からテストまで、川の流れのように一定方向に作業を進めるのを原則としている開発モデルである。システム開発を基本計画、外部設計、内部設計、プログラム設計、プログラミング、テストの工程順に進め、後戻りせずに開発を進める方式である。分割された各工程の成果を段階的に確認しながら開発を進める段階的アプローチ法である。上流工程ほど厳しい基準の設定が必要となるが、ユーザの要求分析に曖昧さが残り開発を進める上でその後の各工程に多くの問題を発生させるケースが多い。

ウォーターフォールモデルの長所・短所

- ① 目的がはっきりしているときは、その目的に合った効率的なシステム開発ができる。
- ② 大規模なシステム開発に向いている。

- ③ メンバの役割分担が明確であり、作業に習熟できる。
- ④ 開発費用や工数が多くなる。
- ⑤ 開発期間が長期にわたるため、その間にユーザの要求が変化することも多く、それに対してタイムリーに対応できない。
- ⑥ 抽象的な仕様を段階的に具体化していくという手順を踏むため、最終工程が終わるまで、仕様が100%実現できるのか判らない場合がある。
- ⑦ 要求仕様の段階で不具合が入り込みやすく、上流工程で入ったバグはユーザの目に触れる下流工程まで隠れてしまうことが多い。
- ⑧ 現実のプロジェクトでは反復作業が常に生じるが、工程間のフィードバックが簡単に行えない仕組みになっている。

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

問4 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5 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。

プロト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6 イ

プロトタイプモデルに関する問題である。

プロトタイプモデルは、開発工程の早い段階で目に見える形でユーザが要求を確認できるように、試作品を作成しながらシステムを構築していく手法である。プロトタイプはユーザの要求や意見を明確にするために作成する画面イメージやデザインなどの試作品のことである。

求める答えはイとなる。

問7 ウ

開発モデルのスパイラルモデルに関する問題である。

アのE-Rモデルは、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。

イのウォーターフォールモデルは、システムの開発工程を、基本計画、外部設計、内部設計、プログラム設計、プログラミング、テストのフェーズに分け、上流から順次作業を進めていく方法で前のフェーズに戻ることがない開発手法である。

ウのスパイラルモデルは、ウォーターフォールモデルを何度か繰り返して、機能を追加していくインクリメンタルモデルを改良したモデルで、ユーザの要望を取り入れながら、サブシステムごとに開発プロセスモデルを繰り返して進めていく方法である。開発単位が独立している場合に適している。求める答えはウとなる。

エのプロトタイピングは、試作品をユーザに見せ、それをたたき台にユーザの要求を聞きながら、システムを完成させていくソフトウェア開発手法である。早い段階でユーザの要求を明確に理解し、開発者との認識のずれをなくすことができる。

問8 ア

スパイラルモデルに関する問題である。

スパイラルモデルは、ウォーターフォールモデルとプロトタイプモデルの両方のよい部分を取り入れた開発手法である。システムの範囲を限定して利用者の要求を聞き取り、システムの仕様をまとめ、プログラミング、テストを行い、利用者の評価と確認をとる。利用者が満足するまで設計、プログラミング、テストを繰り返し開発する。リスクを評価しながら開発するので、リスクが最小になる。

スパイラルモデルの特徴

- ① 検証フェーズでフィードバックの機構が実現し、仕様の変更に柔軟に対応する。
- ② プロトタイピングによるソフトウェア開発に適応できる。
- ③ プロトタイプの目的は、顧客の要求を早い段階に確認することやシステムの実現を事前に予測することにある。
- ④ 稼働の容易さ、導入の効率性、変更の迅速性、進化性の特性がある。
- ⑤ 機能を順次追加しながら短期間で開発することができる。
- ⑥ 開発リスクを最小にできる。

アはスパイラルモデル、イ、ウ、エはウォーターフォールモデルである。求める答えはアである。

問9 イ

スパイラルモデルに関する問題である。

スパイラルモデルは、ウォーターフォールモデルにプロトタイプモデルの手法を取り入れた開発モデルで、一方向を維持しながら、何回も繰り返す手順をらせん状に実現する。目標設定の最適化フェーズ、要求定義の分析開発フェーズ、レビューの検証フェーズ、ソフトウェア開発の計画フェーズで1サイクルを完了する。再分析、設計、プログラミング、テストとサイクルをあげるように移動する。次の段階の開発内容を検討し、開発リスクを特定し、リスクを最小にしながら、開発サイクルを繰り返す、システムの完成度を高めていく手法である。

アのウォーターフォールモデルは、システムの開発工程を、基本計画、外部設計、内部設計、プログラム設計、プログラミング、テストのフェーズに分け、上流から順次作業を進めていく方法で前のフェーズに戻ることがない開発手法である。

イのスパイラルモデルは、ウォーターフォールモデルを何度か繰り返して、機能を追加していくインクリメンタルモデルを改良したモデルで、ユーザの要望を取り入れながら、サブシステムごとに開発プロセスモデルを繰り返して進めていく方法で、開発リスクを最小化する。開発単位が独立している場合に適している。求める答えはイとなる。

ウの成長モデルは、ソフトウェアの変更要求の度に、モデリング、要求定義、実装、評価、再モデリングを繰り返す方法である。仕様変更要求に柔軟に対応できるが、プロジェクト管理がやりにくい問題がある。

エのプロトタイピングは、試作品をユーザに見せ、それをたたき台にユーザの要求を聞きながら、システムを完成させていくソフトウェア開発手法である。早い段階でユーザの要求を明確に理解し、開発者との認識のずれをなくすことができる。

問10 イ

スパイラルモデルに関する問題である。

スパイラルモデルは、ウォーターフォールモデルにプロトタイプモデルの手法を取り入れた開発モデルである。一方向を維持しながら、何回も繰り返す手順をらせん状に実現する。目標設定の最適化フェーズ、要求定義の分析開発フェーズ、レビューの検証フェーズ、ソフトウェア開発の計画フェーズで1サイクルを完了する。再分析、設計、プログラミング、テストとサイクルをあげるように移動する。次の段階の開発内容を検討し、開発リスクを特定し、リスクを最小にしながら、開発サイクルを繰り返す、システムの完成度を高めていく手法である。

アのウォーターフォールモデルは、システムの開発工程を、基本計画、外部設計、内部設計、プログラム設計、プログラミング、テストのフェーズに分け、上流から順次作業を進めていく方法で前のフェーズに戻ることがない開発手法である。

イのスパイラルモデルは、ウォーターフォールモデルを何度か繰り返して、機能を追加していくインクリメンタルモデルを改良したモデルで、ユーザの要望を取り入れながら、サブシステムごとに開発プロセスモデルを繰り返して進めていく方法で、開発リスクを最小化する。開発単位が独立している場合に適している。求める答えはイとなる。

ウのプロトタイピングは、試作品をユーザに見せ、それをたたき台にユーザの要求を聞きながら、システムを完成させていくソフトウェア開発手法である。早い段階でユーザの要求を明確に理解し、開発者との認識のずれをなくすことができる。

エのリレーショナルモデルは、リレーションは関係データベースで、関連ある項目で作られた行をまとめ、2次元の表にしたものである。形式の異なる複数の表の集まりで構成されて、各表

がリレーションあるいは関係と呼ばれる構造を持ったモデルである。

問11 エ

システム開発モデルに関する問題である。

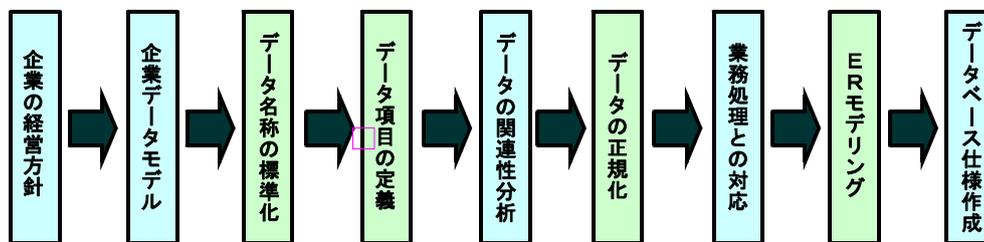
ウォーターフォールモデルは、要求分析、システム設計、製造、テスト、運用の一連の段階的工程からなる。プロトタイプモデルは、初期段階の試作、ユーザインタフェースの確定、手戻りによるリスクの防止などのキーワードが重要である。スパイラルモデルは、成長型モデルで、開発コストや品質からリスクを評価し、リスクを最小にするモデルである。

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12 ウ

データ中心アプローチに基づくデータ標準化作業の問題である。

データ標準化、データベース設計の手順



この問題は正規化の手順を求める問題であり、

データ名称の標準化→データ項目の定義→データの正規化→E-Rモデリングの順になり、求める答えはウとなる。

問13 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14 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15 ア

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は、データや情報を組織体の共有の経営資源と見なし、他の経営資源と同様に計画・管理・統制していく考え方である。データや情報資源をシステムの骨組みと考え、対象業務が持つデータとそれらの関連やデータの発生・流通の仕組みを定義していく。

アはデータ中心アプローチ、イはオブジェクト指向、ウはモジュール分割の考え方、エは機能中心アプローチである。求める答えはアとなる。

問16 ウ

データ中心設計の特徴に関する問題である。

データ中心アプローチは、データや情報を組織体の共有の経営資源と見なし、他の人、もの、金と同様に計画・管理・統制していこうという考え方である。データをシステムの骨組みと考え、対象業務がもつデータとそれらの関連やデータの発生・流通の仕組みを定義していく。

データ中心アプローチの設計手順は次の通りである。

- | | |
|--------------|---------------|
| ① 企業目標の設定 | ② 企業活動の分析 |
| ③ 情報システムの体系化 | ④ データ体系の確立 |
| ⑤ データモデルの作成 | ⑥ 個別業務システムの開発 |

アはデータ中心設計の設計手順、イはプロセス中心設計の特徴、ウはデータ中心設計の特徴、エはプロセス中心アプローチの考え方の特徴である。求める答えはウである。

問17 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間を明確にする。
- 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間を分析して、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18 エ

I T 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19 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20 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21 ア

システム化構想立案に関する問題である。

企画プロセスのシステム構想立案時に業務の新全体像を作成する。企業で将来必要となる最上位の業務機能と業務組織のモデルを検討する。検討の結果、目標とする業務の新しい全体像を描き、新システムの全体イメージを作成し、業務機能と組織モデル、新システムとが整合しているかを確認する。

アは企画プロセスのシステム化構想立案時に作成、イ、エはシステム方式設計時、ウはソフトウェア方式設計時で作成され、いずれも開発プロセスで行う作業である。求める答えはアとなる。

問22 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23 イ

システム開発工程に関する問題である。

システム開発工程で最初に行う作業はモデリングのためのシステム要件を明確にするために、現状業務を分析することである。従って、現状業務を分析し、システム要件を整理することになる。求める答えはイとなる。

システム要件整理後、モデリングを行い、その後モデルに従って、システムの構築、評価の順に展開される。モデリングの段階で、システム分割、コンポーネント設計、ユーザインタフェース設計が行われ、システム構築段階にプログラムの内部構造が設計される。

問24 ア

企画プロセス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、システムを実現するための実施計画を得ることである。求める答えはアとなる。

アは主ライフサイクルプロセスの企画プロセス、イ、ウ、エは支援ライフサイクルプロセスのイは検証プロセス、ウは問題解決プロセス、エは文章化プロセスである。

問25 ウ

共通フレームによる企画プロセスの定義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。

ウの経営事業の目的、目標を達成するために必要なシステムに関係する経営上のニーズなどは、企画プロセスで行われる。求める答えはウとなる。

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26 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27 エ

要件定義プロセスに関する問題である。

要件定義プロセスの目的は、新たに構築する業務、システムの仕様を明確化し、それをベースにIT化範囲とその機能を明示することである。活動内容として、利害関係者の識別、要件の識別、要件の評価、要件の合意などがある。

アは利害関係者の識別、イは要件の評価、ウは要件の合意、エは要件の識別である。求める答えはエとなる。

問28 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29 ア

要件定義で行う作業に関する問題である。

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。

イの費用と投資効果は、企画プロセスのシステム化計画の立案で行う。

ウの事業環境の分析、事業目標、業務目標の作成は、企画プロセスのシステム化構想の立案で

行う。

エのソフトウェア、データベースのテストは開発プロセスで行う。

問30 イ

システム計画を立案する場合の考慮事項に関する問題である。

共通フレームは、コンピュータ・システムの開発において、システム発注側と受注側の間で相互の役割や業務の範囲・内容、契約上の責任などに対する誤解がないように、双方に共通して利用できるよう用語や作業内容を標準化するために作られたガイドラインである。システム構築・運用の受発注において、契約上のトラブル防止、作業内容の確認、役割分担の明確化、社内作業標準の策定や人員計画、見積もり精度の向上、品質確保などに利用される。

アの場合、運用の方法や運用に関係する開発関連業務において、他社との関係やユーザ・ベンダーの関係が発生することがあり、自社の社員だけで開発することを前提に検討することは好ましくない。

イの開発、保守、運用に関する費用と投資効果を明確にする作業はシステム化計画立案時に行う。求める答えはイとなる。

ウの同業他社を調査しても、それぞれの企業の特徴、業務上の問題、プロジェクト上の特徴が異なり、必ずしも同じシステムにはならない。参考にする程度になる。

エのシステム計画の段階で、運用マニュアル、障害対策を具体的にすべて示すことは不可能に近い。通常は不能である。

問31 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32 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、エは、機能要件であり、イは非機能要件である。求める答えはイとなる。

問33 ウ

トレーサビリティに関する問題である。

アのインターオペラビリティは、異なるシステムから同じソフトウェアやデータを利用する際の容易さを表す。パソコンで利用するデータに携帯電話からアクセスできるなど、手段や端末の違いを意識せずに利用できれば、インターオペラビリティは高いと考える。

イのセキュリティは、システム内外に潜在する脅威を抑制するために、システムの機密性、完全性、可用性を維持することである。

ウのトレーサビリティは、対象とする物品とその部品や原材料の流通履歴を確認できることである。定義された要件に対して、発生した変更要求の実装までの経過を明らかにできることである。求める答えはウとなる。

エのユーザビリティは、使い勝手とは、使いやすさの程度を表す。

問34 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35 エ

E-R図に関する問題である。

アの商品と仕入れ先は $m:n$ の関係であるから、同一の商品は複数の仕入れ先から仕入れる。

イの発注明細と納品明細は1対1の対応ではなく、 $m:n$ の関係になる。

ウの発注と商品との対応は、1つの商品は複数の発注明細に含まれており、1つの発注明細には1つの商品であるから、商品と発注明細は1:多の関係にある。発注と発注明細は1:多の関係にあるから、1つの発注に複数種類の商品が含まれることがある。

エの発注と仕入れ先は、多:1の関係であるから、1つの発注で複数の仕入れ先に発注することはない。求める答えはエとなる。

問36 ウ

E-R図に関する問題である。

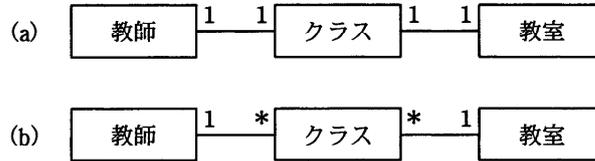
E-R図は、E-Rモデルでユーザ要求定義の分析に使用されるダイアグラムで、データ構造をエンティティ、アトリビュート、エンティティ間の関連という概念で捕らえて分析した上でモデル化する際に、これらの内容を図式化して表現するために用いる。対象とする世界を実体と関連の二つの概念で表現する図である。

アはUML、イは状態遷移図、ウはER図、エはDFDである。求める答えはウとなる。

問37 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38 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れ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ることができる、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39 ウ

DFDに関する問題である。

DFDは、業務やシステムの機能を視覚的に表現するための図解法で、データの流れに着目し、データフロー、プロセス、データストア、源泉の4つの構成要素を組み合わせで作成する。

DFDは要求された機能をいくつかの単位に分割して、その単位間のデータの流れを表す図である。求める答えはウとなる。

問40 イ

DFDの要素に関する問題である。

DFDは、データ源泉・吸収、データストア、データフロー、プロセスの4要素からなる。答

えはデータストアで、求める答えはイとなる。

問41 エ

D F Dに関する問題である。

D F Dの要素は、○はプロセス、＝はデータストア、→はデータストア、□は源泉・吸収である。求める答えはエとなる。

問42 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43 ウ

D F Dの構成要素に関する問題である。

D F Dはデータフロー、プロセス、データストア、源泉の4つの要素で構成される。Aの要素はストアを表している。

アの売上ファイルは売上状況を記載したファイルで、売上年月日、商品コード、売上数量などのデータ項目が含まれている。

イの受注ファイルは商品を受注する際の情報を記載したファイルで、受注年月日、商品コード、受注数量、顧客コード、納期などから構成される。

ウの単価ファイルは、商品の単価を記載したファイルで、商品コード、単価で構成される。

エの入金ファイルは、入金状況を記載したファイルで、入金年月日、顧客コード、金額などで構成される。

受注情報に基づいて売上計算を行い売上台帳を作成するプロセスであるから、受注数量に単価を掛けて売上金額の計算が必要になり、その際に利用されるファイルであるから、単価ファイルである。求める答えはウとなる。

問44 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DFDは、直下のレベルのDFD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DFD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45 ウ

DFDの記述内容に関する問題である。

アのアルゴリズムは、処理手順であり、流れ図に関係する。

イのイベントは、状態変化を引き起こす出来事で、状態遷移図における遷移のきっかけになる出来事である。

ウのデータの流れは、DFDにおけるデータ処理を行う機能間の流れである。DFDに記述されるものはデータの流れで、求める答えはウである。

エの入出力装置や外部記憶装置は、中央処理装置の周辺装置に相当するものである。

問46 エ

DFDの表記法に関する問題である。

DFD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、データの源泉・吸収は四角、データフローは矢線、データ蓄積は二重線、処理は円の記号を用いる。

アの平行線はデータの蓄積である。

イの円は処理を表す。

ウの四角は、源泉・吸収であり、データの発信元や送信先の顧客などを指す。

エの矢印は、データの流れで、矢線の上にデータ名を付ける。求める答えはエとなる。

問47 ア

構造化分析に関する問題である。

構造化分析技法は、デマルコの提案したユーザのソフトウェアへの要求を表現する手法で、データフローダイアグラム、データディクショナリ、ミニ仕様書を使った構造化仕様書を作成する技法である。

アの構造化分析は、開発対象の業務をモデル化し、概要モデルから詳細モデルへと段階的に分析してシステムを構造的にとらえる。分析・設計作業では、DFDや階層構造図、E-R図、状態遷移図などの図記号を使った表記法を使用する。求める答えはアである。

イの複合設計は、構造化をベースとしたモジュール分割のための設計技法で、プログラムを階層化したモジュール構造に変換するための方法である。

ウのプロトタイプングは、試作品をユーザに見せ、それをたたき台にユーザの要求を聞きながら、システムを完成させていくソフトウェア開発手法である。早い段階でユーザの要求を明確に

理解し、開発者との認識のずれをなくすることができる。

エの要求仕様分析は、顧客の要求を洗い出し、それらをどのような範疇のもとにまとめたり、類型化したらよいかを明らかにすることである。

問48 イ

状態遷移図、状態遷移表に関する問題である。

状態遷移図から判断して、Aは状態2、Bは状態3、Cは状態1、Dは状態3となる。求める答えはイである。

問49 イ

状態遷移図に関する問題である。

状態遷移図は、状態の遷移を表現した図であり、とり得る状態を円で、状態の変化と変化させる条件を矢印で表現する。タスクの状態遷移やリアルタイム処理の状態変化などを表現するのに利用する。状態遷移図は、世の中の機構はとり得る状態の中から一時点では必ずどれかの状態をとって安定するが、それぞれの状態に対応するイベントが発生するとそのイベントの条件に従って別の安定した状態に遷移するという有限状態マシンの理論に基づいている。時間の経過や状況の変化に基づいて、そのときの動作を記述したものであ。

アは階層構造図、イは状態遷移図、ウは段階的詳細化、エはDFDである。求める答えはイとなる。

問50 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51 ウ

H I P Oに関する問題である。

H I P Oは、システム設計技法の一つで、機能分割やプログラム構造の設計を行うときに用いられる手法である。システムの機能を3つの構成で階層的に記述する。

図式目次は、システムを構成するモジュールの関係を階層的に表現する。総括ダイヤグラム、詳細ダイヤグラムは各モジュールを入力・処理・出力の形で表現する。

図式目次、総括ダイヤグラム、詳細ダイヤグラムがH I P Oを構成する組み合わせで、求める答えはウとなる。

問52 ウ

決定表に関する問題である。

判断表は問題を処理するための条件と、その行動の関係を表形式で表すものである。条件と行動との対応関係を表で表現することによって、複雑な論理を一目でわかるように表現できる。

条件表題欄は判定すべき、すべての条件を記入する。行動表題欄は特定の条件が満足された場

合に実行すべき処理のすべてを列挙する。条件記入欄は条件表題欄の記入条件に対する判定結果を記入する。行動記入欄は条件記入欄に記入した判定が満足された場合にとるべき行動の該当欄にX印を入れる。

条件表題欄	条件記入欄
行動表題欄	行動記入欄

アはERダイアグラム、イはDFD、ウは決定表、エは流れ図である。求める答えはウとなる。

問53 エ

オブジェクト指向に関する問題である。

アのオブジェクト間の情報交換はメッセージで行う。インスタンスはオブジェクトの実現値そのものである。

イは、同じ性質を持つオブジェクトの集まりがクラスであり、逆にオブジェクトはクラスの性質を継承したものであり、クラスの性質を記述したものではない。

ウのカプセル化はある単位のデータと操作をまとめて一つのモジュールとし、外部との間にインタフェースを設けることによって他と独立分離し、モジュール間の依存性を最小にすることである。ライブラリは共通で使用できる関数やデータを一つにまとめたもので、複数のアプリケーションで共有するものである。カプセル化されたものはオブジェクト独自のものである。

エは、スーパークラスの持つ性質やメソッドはそのサブクラスの性質やメソッドとして通用する。クラスはその親のクラスから性質やメソッドを継承する。求める答えはエである。

問54 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55 エ

オブジェクト指向技術に関する問題である。

アのクラス全体で共有されるデータはクラス変数である。インスタンス変数はオブジェクトの

性質を表すもので、同じクラスに属するオブジェクトは共通のインスタンス変数を持っている。
イのクラスの中のメソッドは手続きであって、共有データではない。
ウの複数の上位クラスからの性質の継承は多重継承である。
エのオブジェクトが他のオブジェクトに処理を依頼する場合、メッセージ通信によってのみ可能である。求める答えはエとなる。

問56 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57 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に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58 ア

オブジェクト指向の継承に関する問題である。
アの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。求める答えはアとなる。
イの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。
ウの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り一般化したクラスを定義する考え方である。
エの多様性はオブジェクト間のメッセージのやり取りにおいて、複数のオブジェクトに対して同一名のメッセージを送信しても、それぞれのオブジェクトが固有の振る舞いを実行することである。

問59 ウ

オブジェクト指向の継承に関する問題である。

アはクラスの定義であり、そのためにクラスライブラリを用いて記述される。クラスを定義する場合に、その上位のスーパークラスを利用して定義することはできる。しかし、クラスライブラリの利用が継承を意味することではない。

イはクラスを具現化することで、インスタンスの生成である。

ウのスーパークラスで定義された属性やメソッドを、そのサブクラスが共有することは継承の概念である。求める答えはウである。

エはオブジェクト間のメッセージによるコミュニケーションであって、継承ではない。

問60 イ

オブジェクト指向の継承に関する問題である。

継承は、スーパークラスで定義した性質を、サブクラスへ引き継ぐことができる。これによって、サブクラスでは同じ性質を定義する必要がなくなる。サブクラスで、新たに必要となった性質は、その都度、サブクラスにおいて定義する。このことを差分プログラミングという。プログラミングの生産性向上に、継承は非常に有効な手段である。適当なクラスを見つけ、そのクラスの属性やメソッドを利用して新しいクラスを定義することができる。新しく定義したクラスに必要なに応じて、差分プログラミングを使用して属性やメソッドを負荷することができる。このクラスに属性値を与えることによって、オブジェクトのインスタンスを生成できる。

アの基底クラスで定義されたデータを保護はしない。

イは継承の考え方で適切である。求める答えはイとなる。

ウの基底クラスの変更はサブクラスに影響する。

エの再利用はサブクラスについても可能である。

問61 ア

継承に関する問題である。

継承(インヘリタンス)は、ある性質を引き継ぐことである。オブジェクトの世界では、論理的に同質なものでも、物理的な空間において異質なものとして投影することになる。この投影という考え方が、継承を生み出した。インヘリタンスの機構はクラス間において存在する。スーパークラスで定義した性質を、サブクラスへ継承することができる。これによって、サブクラスでは同じ性質を定義する必要がなくなる。サブクラスで、新たに必要となった性質は、その都度、サブクラスにおいて定義する。このことを差分プログラミングという。サブクラスは複数のスーパークラスから継承することができる。サブクラスから見て、一つのスーパークラスから継承を行うことを単一継承、複数のスーパークラスから継承を行うことを多重継承と呼ぶ。

アは継承(インヘリタンス)、イはカプセル化、ウは情報隠蔽、エはクラスである。求める答えはアとなる。

問62 エ

オブジェクト指向に関する問題である。

オブジェクト指向はデータ構造における共通の特性を示すクラスという概念によってモデル化

したものである。オブジェクトは能動的な振舞いを行い、メッセージパッシングが行われると、そのメッセージに対応したオブジェクトが自律的に動作する現象を利用してモデル化を進める。オブジェクトは存在と状態によってその特性を決定する。また、オブジェクトは振舞いをもっている。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。属性とメソッドを一緒にすることをカプセル化と呼ぶ。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。一つのクラスは、一つ以上の属性から構成される。同一のクラスに属するオブジェクトは同じ属性をもつ。属性やメソッドはクラスで定義されている。

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でできる。操作はオブジェクトの振る舞いに対してインタフェースを介して行うメッセージである。操作を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のであって、相互依存性を高めるは誤りである。

ウのクラスの変更は、継承の概念から考えると変更に関係する上位の特定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63 エ

サブクラスに関する問題である

サブクラスは、スーパークラスで定義した性質を継承することができたため、同じ性質を定義する必要がなくなる。サブクラスで、新たに必要となった性質は、その都度、サブクラスにおいて差分プログラミングを使用して定義する。

スーパークラス自動車の性質を継承しているオブジェクトはトラックである。求める答えはエとなる。

アのエンジンは自動車の構成要素であり、別のオブジェクトであり、集約関係にある。

イの製造番号は自動車の属性である。

ウのタイヤは自動車と集約関係にある別のオブジェクトである。

問64 ウ

オブジェクト指向の汎化・特化に関する問題である。

同じ種類のオブジェクトをまとめたものがクラスである。

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汎化の関係で、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65 ア

情報隠蔽に関する問題である。

オブジェクトの状態と振る舞いを一体化することをカプセル化といい、カプセル化することによって状態と振る舞いをブラックボックス化することを情報隠蔽という。情報隠蔽された振る舞いを動作させるのは外部のオブジェクトからのメッセージによる。カプセル化によってオブジェクトの独立性が高まり、再利用がしやすくなる。また、メッセージ以外の手段ではオブジェクトにアクセスできないため、オブジェクトの独立性と信頼性を一層高める。

アは情報隠蔽、イは集約化、ウは継承、エは多様性を表す。求める答えはアとなる。

問66 エ

カプセル化に関する問題である。

オブジェクトの状態と振る舞いを一体化することをカプセル化といい、カプセル化することによって状態と振る舞いをブラックボックス化することを情報隠蔽という。

アは汎化、イは抽象クラス、ウは継承、エがカプセル化であり、求める答えはエとなる。

問67 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る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68 イ

オブジェクト指向のカプセル化に関する問題である。

アのインスタンスはクラスに属するオブジェクトが具現化することである。

イのカプセル化は、データとメソッドを一体化し、オブジェクトの内部に隠蔽することであり、外部にはメソッドのインタフェースを公開する。求める答えはイとなる。

ウのクラスタ化は、複数のコンピュータを統合して1つのサーバシステムを構築して、システムの耐障害性を高める手段であり、無停止での運用が要求される基幹業務で活用される。

エの抽象化は、外界の対象に対して、最も特徴的な要素だけを抽出して、それ以外のものを捨てることによって、特性を見出す考え方である。具現化の反対である。

問69 ア

カプセル化に関する問題である。

オブジェクトの状態と振る舞いを一体化することをカプセル化といい、カプセル化することによって状態と振る舞いをブラックボックス化することを情報隠蔽という。情報隠蔽された振る舞いを動作させるのは外部のオブジェクトからのメッセージによる。カプセル化によってオブジェ

クトの独立性が高まり、再利用がしやすくなる。また、メッセージ以外の手段ではオブジェクトにアクセスできないため、オブジェクトの独立性と信頼性を一層高める。

アのカプセル化は、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構造にすることである。求める答えはアとなる。

イのクラス化は、オブジェクトに共通するデータ属性とメソッドをカプセル化して、クラスを定義することである。

ウの構造化は、サブシステム、プログラム、モジュールの機能を定義し、その機能間の従属関係を階層的に求める技法である。トップダウンアプローチの考え方をを用いる。各機能を定義し、機能間のインターフェースを明確にして、図式化する。

エのモジュール化は、プログラムを特定の機能を持ったいくつかのモジュールに分割することである。分割されたモジュール間にはインターフェースを定義して、階層構造化する。

問70 ア

カプセル化に関する問題である。

アのカプセル化は、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構造にすることである。求める答えはアとなる。

イのクラス化は、オブジェクトに共通するデータ属性とメソッドをカプセル化して、クラスを定義することである。

ウの集約化は、オブジェクト同士の関連のうち全体と部分を表す関係を作ることである。

エの汎化は、クラスやオブジェクト同士の関係で、複数のクラスまたはオブジェクトに共通する性質に着目して抽象化することである。

問71 ウ

汎化に関する問題である。

アの集約は、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。

イの伝播は、ある地点から別の地点に伝わり広がることである。

ウの汎化は、クラスやオブジェクト同士の関係で、複数のクラスまたはオブジェクトに共通する性質に着目して抽象化することである。上位のクラスをスーパークラスと呼び、汎化関係にあるクラスは継承によって表現される。求める答えはウとなる。

エの役割は、概念間の関係を関連と呼び、関連のそれぞれの終端を役割と呼ぶ。役割には、名前、多重度の表現、誘導可能性などの要素をもたせることができる。

問72 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73 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る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、その振る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る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る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74 イ

オブジェクト指向に関係するクラス間の問題である。

アのクラス間の関連は2つ以上のクラス間でも定義できる。n項関連の定義が可能である。

イのサブクラスでのスーパークラスの操作を再定義することができる。求める答えはイとなる。

ウのサブクラスのインスタンスの操作はスーパークラスからの継承によって自ら操作することができる。スーパークラスに依頼しない。

エの2つのクラスが集約関係にあるときは、集約オブジェクトは部品オブジェクトによって構成されるのであって、属性と操作を共有するのではない。

問75 ア

多相性に関する問題である。

多相性は、オブジェクト指向プログラミング言語の持つ性質のひとつで、同名のメソッドや型などをオブジェクトの種類に応じて使い分けることができる性質のことである。同名のメソッドであっても、その操作を受け取るオブジェクトによって異なる処理内容を実行させることができる。クラスの継承を行う際、上位クラスの関数を新しい関数の定義で置き換えることを「オーバーライド」と呼ぶ。多相性を実現する際にオーバーライドをよく使用する。多相性の利点は、メソッド名を統一することで名前を覚える必要がなくなることや、記述ミスを減らせることなどが挙げられる。

アのオーバーライドは、スーパークラスで定義されたメソッドをサブクラスで定義しなおし、動作を上書きすることである。多相性を実現する際に使用する。求める答えはアとなる。

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。

ウの多重継承は、あるクラスを複数のクラスの性質を受け継ぐ形で定義することである。

エのメッセージパッシングは、オブジェクト間のメッセージによるコミュニケーションのことである。

問76 ウ

UMLに関する問題である。

UMLは、オブジェクト指向モデルを作成するための言語で、UMLを使用することによって、ビジネスのモデル化、開発局面でのソフトウェアのモデル化、更に、静的構造のモデル化、動的振舞いのモデル化など、システム全般のモデル化を可能にするツールである。利用範囲は非常に広く、分析・設計・実装など開発フェーズのあらゆる局面で利用できる。UMLで使用する図式には、ユースケース図、アクティビティ図、クラス図、オブジェクト図、状態図、シーケンス図、コラボレーション図、アクティビティ図などを使用する。

クラス図はUMLで用いられる。求める答えはウとなる。

問77 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78 ウ

UML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。データの流を視覚的に分かりやすく表現する図である。

イのE-R図は、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。対象とする世界を実体と関連の二つの概念で表現する図である。

ウの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を体系化したものである。業務プロセスのモデリング表記法として用いられる。求める答えはウとなる。

エの状態遷移図は、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。時間や行動などに応じて、状態が変化する状況を表現する図である。

問79 エ

UMLに関する問題である。

アのE A Iは異種のアプリケーション間でデータを関係させるミドルウェアである。

イのE J BはJavaアプリケーションのアプレットを組み合わせ、更に大きなアプレットを作成できるようにするインタフェースである。

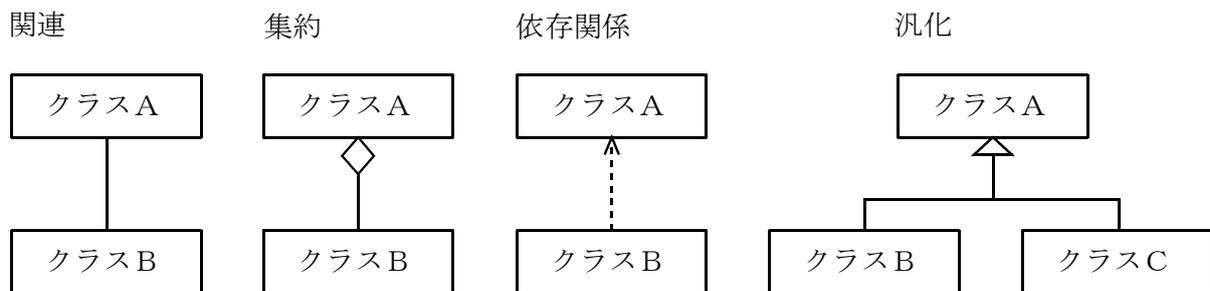
ウのE R Pは経理、生産管理、販売管理、人事管理などの基幹業務を個別に行うのではなく、コンピュータシステムを使って関連付けながら実行することである。

エのU M Lはオブジェクト指向分析／設計で用いられるモデリング言語である。システム開発で利用され、分析から設計、実装、テストまでを統一した表記法である。求める答えはエとなる。

問80 イ

U M L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81 イ

U M Lの汎化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前問に示す。

汎化は、クラスやオブジェクト同士の関係で、複数のクラスまたはオブジェクトに共通する性質に着目して抽象化することである。上位のクラスをスーパークラスと呼び、汎化関係にあるクラスは継承によって表現される。

アは関連の多重度、イは汎化、ウは集約、エは依存関係である。求める答えはイとなる。

問82 ウ

U M L技法に関する問題である。

アのオブジェクト図は、ある特定時点でのオブジェクトのインスタンス間の静的な構造を表現したものである。実行時のある状況でのスナップショットである。

イのクラス図は、オブジェクトとそのオブジェクト間の関係を静的に表現したものである。

ウのシーケンス図は、オブジェクトの動的な振る舞いを時系列に沿って表現した図である。求める答えはウとなる。

エのパッケージ図は、システムの全体構成をパッケージ間の依存関係として図式表現したもので、静的な表現である。

問83 エ

UMLのクラス図に関する問題である。

UMLはオブジェクト指向モデルを作成するためのモデリング言語である。UMLを使用することによって、ビジネスのモデル化、開発局面でのソフトウェアのモデル化、更に、静的構造のモデル化、動的振舞いのモデル化など、システム全般のモデル化を可能にするツールである。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。

アはシーケンス図、イはコラボレーション図、ウはステートチャート(状態図)、エはクラス図である。求める答えはエとなる。

問84 イ

クラスとインスタンスの関係に関する問題である。

クラスはオブジェクトに共通するデータ属性とメソッドをカプセル化して定義したものであり、インスタンスはクラスに具体的なデータを与えて生成したオブジェクトである。

アはクラスがインスタンスの仕様を定義したものである。

イのクラスの定義に基づいてインスタンスを生成するは適切である。求める答えはイとなる。

ウのインスタンスは一つのクラスから生成される。クラスを生成する場合、複数のクラスから継承して生成される場合がある。

エは一つのクラスに対して複数のインスタンスが生成される。

問85 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86 イ

UMLダイアグラムに関する問題である。

アのアクティビティ図は、システムや業務の流れを表現する図、振る舞いを表現する図、アクションの実行順序を定義する図である。

イのオブジェクト図は、インスタンスとその関係を定義する図である。求める答えはイとなる。

ウのコンポーネント図は、コンポーネントを定義する図である。

エのユースケース図は、システムが提供する機能を定義する図である。

問87 ウ

オブジェクト指向言語の特徴に関する問題である。

オブジェクト指向言語は、データと手続きが一体化したオブジェクトをベースとしたプログラム言語で、C++やJavaなどが代表的な言語である。

オブジェクト指向言語の特徴

- ① オブジェクトへの処理の依頼は処理要求メッセージを送ることで実行する。
- ② 外部から直接オブジェクト内のデータ、メソッドにアクセスすることはできない。
- ③ 共通の特性をもつオブジェクトはクラスにまとめられ、クラスは階層構造を形成する。
- ④ 下位のクラスは上位のクラスの性質を継承する。
- ⑤ 下位クラスの機能の充足は差分プログラミングによって可能となる。
- ⑥ プログラミングの冗長さの低下
- ⑦ プログラミング効率の向上

求める答えはウとなる。

問88 ア

オブジェクト指向プログラミングの特徴に関する問題である。

オブジェクト指向プログラミングの特徴は次のようになる。

① インスタンス機能

クラスから効率的にオブジェクトを生成する。同一のクラスに属するオブジェクトは、共通のメソッドを持つ。

② カプセル化機能

データとメソッドを一体化した機能で、オブジェクト指向プログラミングの生産性と信頼性を向上させる。オブジェクトの独立性が高く、開発時や保守時の波及効果が及ばず、効率のよい作業工程が実現する。

③ メッセージパッシング機能

オブジェクト同士が互いに情報をやり取りし、協調的に問題解決を図るために、メッセージを送受信する機能、実行制御の順序やデータ構造について考量する必要なく、協調的に動作する。

④ インヘリタンス機能

親クラスで指定された性質は、子クラスに引き継がれる。子クラスでは、新たなメソッドを追加することができる。これを差分プログラミングという。対象となる適用業務処理毎に、既存のクラス階層をライブラリとして作成し、再利用が可能になる。

アはメッセージパッシング機能で、協調動作し、プログラムの機能を実現する。求める答えはアとなる。

イは、データとメソッドはカプセル化され、外部から直接操作できない。

ウは、インヘリタンス機能で、下位クラスには差分プログラムによって機能や性質が付加できる。

エは、データはカプセル化され、オブジェクト内部に隠蔽されているためデータ辞書に登録する必要は無い。

問89 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90 イ

コンポーネントウェアに関する問題である。

コンポーネントは構成要素を意味する言葉で、情報処理ではハードウェアやソフトウェアなどの構成要素、あるいは一つの領域や機能などを指す。コンポーネントウェアは、標準仕様に基いた再利用可能なソフト部品を組み合わせてアプリケーションを開発する手法またはツールである。ソフト部品はアイコンなどの目に見える形で画面上に表示され、その部品をマウスなどを使って関連付けてアプリケーションを開発する。クラスライブラリやCOM、DCOM、JavaBeans、EJBなどが該当する。

アのグループウェアは共同作業の効率化を実現するための支援ツールである。電子メール、電子掲示板、電子会議室、スケジュール管理、共同文書作成支援、データベースなどがある。

イのコンポーネントウェアは、コンポーネントである部品や部品活用のためのツールを利用してシステムを開発する手法で、既存のコンポーネントや新規に作成したコンポーネントを利用してシステムを構築する。オブジェクト指向によるシステム開発はこの考え方を利用している。求める答えはイである。

ウのマクロコマンドは複雑な操作手順や複数のコマンドを1つのコマンドや簡単な操作で実行させたものである。

エのミドルウェアはオペレーティングシステムとアプリケーションソフトウェアの間にあつ

て、利用者にとってよい操作性、豊富なネットワーク機能、使いやすいデータベース、豊富な図形・画像処理機能などを提供するソフトウェアである。

問91 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ー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92 イ

ソフトウェア方式設計に関する問題である。

ソフトウェアの方式設計は、ソフトウェア要件定義書を基に、ソフトウェアの構造とコンポーネントの設計を行う。ソフトウェアをソフトウェアコンポーネントまで分割し、ソフトウェアコンポーネントの機能、ソフトウェアコンポーネント間の処理手順や関係を明確にする。

実行する業務内容は、ソフトウェア構造とコンポーネントの方式設計、外部およびコンポーネント間のインターフェースの方式設計、データベース最上位レベルの設計、利用者文書の作成、ソフトウェア結合のための要求事項の定義、ソフトウェア方式設計の評価、レビューである。

イの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、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内容が適切である。

アはシステム要件定義、イはソフトウェア方式設計、ウはソフトウェア詳細設計、エはシステム方式設計である。求める答えはイとなる。

問93 エ

外部設計工程の作業内容に関する問題である。

外部設計の主要作業

- | | |
|---------------|----------------|
| ① 要求仕様の確認 | ② サブシステムの定義と展開 |
| ③ 画面設計・報告書の設計 | ④ コード設計 |
| ⑤ 論理データ設計 | ⑥ 外部設計書の作成 |

アの物理設計は内部設計工程、イのプログラムの機能化設計はプログラム設計工程、ウの要求

定義は基本計画工程、論理データ設計は外部設計工程で行われる。求める答えはエとなる。

問94 ア

外部設計工程で行われる作業内容に関する問題である。

アの画面・報告書の設計は外部設計、イのシステム化計画の立案は基本計画、ウのプログラム構造化設計は内部設計、エのモジュールの論理設計はプログラム設計の各工程で行う。求める答えはアとなる。

問95 ア

外部設計におけるデザインレビューに関する問題である。

アの画面レイアウトは外部設計、イのシステム開発計画は基本計画、ウの物理データベース仕様は内部設計、エのプログラム流れ図はプログラム設計またはプログラミングとなる。

外部設計終了段階で承認を受けるのは画面レイアウトで、求める答えはアとなる。

問96 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97 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98 ア

論理データ設計に関する問題である。

論理データ設計で行う作業は、データ項目の洗い出し、データの関連性の分析、ファイル候補の決定、データ項目と処理の対応確認、ファイル仕様の作成である。

アは論理データ設計、イ、ウ、エは物理データ設計である。求める答えはアとなる。

問99 イ

内部設計工程の作業項目に関する問題である。

内部設計で行う作業

- ① 機能分割・構造化は、プログラム詳細機能の洗い出し、図式目次・総括ダイヤグラム・詳細ダイヤグラムの作成、プログラム間インタフェースの作成、プロセスフローの決定などを行う。
- ② 物理データの設計は、主要ファイルの項目／編成方式の決定、レコードレイアウトの設計などを行う。
- ③ 入出力詳細設計は、原票設計、帳票設計、画面設計、入力チェック方式、データ修正法、メッセージの設計などを行う。
- ④ 内部設計書の作成

アのコード設計、エの論理データ設計は外部設計、ウのプログラム構造化設計はプログラム設計、イの物理データ設計は内部設計で行う。求める答えはイとなる。

問100 ア

内部設計の物理データ設計に関する問題である。

物理データ設計の作業内容

- ① ファイル特性の理解
- ② ファイル編成方式の決定
- ③ ファイル媒体の決定
- ④ レコードレイアウトの決定
- ⑤ アクセス時間と容量の見積

イのデータ項目の洗い出し、ウのデータ関連性の分析、エのファイル仕様の作成は外部設計の論理データ設計で行う。

アのアクセス時間と容量の見積は内部設計の物理データ設計で行う。求める答えはアとなる。

問101 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102 イ

チェックディジットに関する問題である。

チェックディジットの特徴

- ① コードの末尾に人為的に数字を付加して、入力誤りの検出に利用する。

- ② 対象とするエラーの種類としては、文字の誤り、隣接文字の逆転誤り、1桁挟んだ文字の逆転誤りなどがある。
- ③ 使用するウェイトによって、エラーの検出力が異なる。
- ④ ランダムエラーの検出力は低い。

アの桁別コード、エのデシマルコードはコード分類の一種類である。

イの入力データの誤りをチェックするために一定の計算法によって数値を算出して利用するのは、チェックディジットの方式である。求める答えはイである。

ウのチェックポイントは、プログラム実行中のある時点で、リスタートのための情報を生成する点である。チェックポイントを通過する時点の主記憶の情報を磁気ディスクに出力しておき、万一システムが停止しても、チェックポイントから処理を継続できるようにする。

エのデシマルコードは、10進分類法である。

問103 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104 ア

チェックディジットに関する問題である。

チェックディジットの計算要領

- ① コードの各桁に指定したウェイトを掛ける
- ② ①で求めた結果の和を求める。(桁別分離法とそのままで和を求める場合がある)
- ③ ②で求めた結果をモジュラス(10または11)で割り、余りを求める。
- ④ ③で求めた余りまたはモジュラスから余りを引いて、検査文字を求める。

チェックディジットコードは、次のように計算する。

$$7 \times 1 + 3 \times 2 + 9 \times 3 + 4 \times 4 = 7 + 6 + 27 + 16 = 56$$

$$56 \div 11 = 5 \text{ 余り } 1$$

$$11 - 1 = 10$$

従って、余り10の1の位であるから、検査文字は0である。求めるコードは73940となり、求める答えはアとなる。

問105 ア

チェックディジットコードに関する問題である。

チェックディジットによるチェック要領

- ① 数字項目の各けたに重み付けを行い、ある計算して得られた1けたの数字をチェックディジットとして数字項目に付加する。
- ② 同じ計算を行って得られた数字とチェックディジットの値を比較し、一致すれば正しいデータ、一致しなければ間違ったデータと判断する。

アの顧客コードの入力誤りのチェックに使用するのはチェックディジットコードを付加する目的である。求める答えはアとなる。

イの配列に使用する、ウの分類に使用する、類推に使用するはいずれも誤りである。

問106 ア

チェックディジット付きコードを求める問題である。

アの場合、810 を計算すると、 $2 \times 4 + 2 = 26 \pmod{(26, 10)} = 6$ 求める答えはアである。

イの場合、 $2 \times 4 + 2 + 2 = 28 \pmod{(28, 10)} = 8$ 当てはまらない。

ウの場合、 $2 \times 4 + 2 + 4 = 30 \pmod{(30, 10)} = 0$ 当てはまらない。

エの場合、 $2 \times 4 + 2 + 6 = 32 \pmod{(32, 10)} = 2$ 当てはまらない。

問107 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108 ウ

チェックディジットに関する問題である。

の数字をXとすると、次の式が成り立つ。

$$(4 \times 3 + 2 \times 2 + X) \pmod{7} = 6$$

$$(12 + 4 + X) \pmod{7} = 6$$

アの場合、 $16 \pmod{7} = 2$

イの場合、 $18 \pmod{7} = 4$

ウの場合、 $20 \pmod{7} = 6$ 求める答えはウとなる。

エの場合、 $22 \pmod{7} = 1$

問109 エ

チェックディジットコードに関する問題である。

手順に従って、検査数字を求めると次のようになる。

偶数桁の合計 $1 + 3 + 5 + 7 + 9 + 1 = 26$

奇数桁の合計 $2 + 4 + 6 + 8 + 0 + 2 = 22$

奇数桁の合計を3倍する $22 \times 3 = 66$

全体の合計を求める $26 + 66 = 92$

検査数字を求める $10 - 2 = 8$

検査数字は8となり、求める答えはエとなる。

問110 ウ

チェックディジットに関する問題である。

アのコード誤りの訂正はできない。チェックディジット法はエラーの検出であって、訂正はできない。

イの誤りの検出はコンピュータの入力時の誤りの検出には利用できるが、出力時の誤り検出には対応しない。

ウの数字以外の文字を含むコードにも対応できる記述は適切な内容であり、求める答えはウとなる。

エのすべてのコードの誤りを検出できない。ランダムエラーの検出力は大幅に低下する。

問111 イ

チェックディジットに関する問題である。

チェックディジットは、データの項目から、ある規則に従って導かれる検査数字をそのデータ項目に付加することによって入力データの誤りをチェックする方法である。

アは数字チェック方式、イはチェックディジット方式、ウは形式チェック、エは範囲チェック方式である。求める答えはイとなる。

問112 イ

商品コードの照合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。

イの照合チェックは、入力された得意先コードや商品コードが実在するかどうかを得意先マスターファイルや商品マスターファイルと照合してチェックする。求める答えはイとなる。

ウのニューメリックチェックは、計算に使用する項目に数字が入っているかをチェックする。数字以外の文字が入っていると計算するときエラーとなる。

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。

問113 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114 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115 エ

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。

アの照合チェックは、入力された得意先コードや商品コードが実在するかどうかを得意先マスターファイルや商品マスターファイルと照合してチェックすることである。

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。

エのリミットチェックは、限度検査、範囲検査のことで、限度検査はデータの値が指定された値を超えていないかどうかを検査することであり、範囲検査は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。求める答えはエである。

問116 イ

CSSに関する問題である。

アのCMSは、Webコンテンツを構成するテキストや画像などのデジタルコンテンツを統合・体系的に管理し、配信など必要な処理を行うシステムの総称である。

イのCSSは、HTMLやXMLの要素をどのように修飾するかを指示する、W3Cによる仕様の一つである。文書の構造と体裁を分離させるという理念を実現する為に提唱されたスタイルシートである。求める答えはイとなる。

ウのPNGは、画像フォーマットの一つである。JPEGやGIFに代わってWWW上で広く使われることを目指して開発された。

エのSVGは、XMLによって記述されたベクターイメージ言語のことである。XMLで記述する為、ブラウザ上で閲覧でき、テキストエディタ等で編集することができる。

問117 イ

入力データのチェックに関する問題である。

ニューメリックチェックは、計算に使用する項目に数字が入っているかをチェックする。数字以外の文字が入っていると計算するときエラーとなる。

アはチェックディジット法、イはニューメリックチェック、ウは論理チェック、エは照合チェックである。求める答えはイとなる。

問118 ウ

客層区分の見直しに必要な帳票の作成要領に関する問題である。

アの売上日をグループキーとして販売金額を集計しても、顧客ID別の情報を得ることができない。

イの客層区分別に希望販売金額を集計しても、顧客別の評価を適切に行うことができないため適正な客層区分の修正にはならない。

ウの顧客ID別に希望販売価格のグループ集計を行い、希望販売価格の合計の集計値を求め、降順に整列すると適正な評価が可能になる。求める答えはウとなる。

エの販売金額別に販売金額を集計しても、顧客別の情報は不明である。

問119 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120 ア

解析条件を満足させる分析表の作成に関する問題である。

優良顧客の定義は、最近購入実績があり、購入回数の多い人である。

判断データは最近の1ヶ月、2ヶ月、…別に期間ごとの購入回数を求めて判定する。期間ごとの購入回数が見目で分かるようになっている表が全体情報と部分情報を同時に見ることができ、適切な判断につながる情報になる。

アの表は、マトリックス上に最近の期間と購入回数が整理されており、期間と購入回数の対応付けが容易であり適切な判断ができる表現である。求める答えはアとなる。

イの情報は、最近の期間と購入回数の関係づけが不十分である。

ウの情報は、購入回数から最近の期間を見る場合には利用できるが、購入回数に最近の期間を加味して考える場合に使用し難い表である。

エの場合はウの場合の逆の現象である。購入回数と最近の期間を関係づけて判断する場合に使用し難い情報提供法である。

問121 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ヶ月の顧客と商品を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122 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123 ウ

プログラムの階層構造化時のインタフェースの個数を求める問題である。

階層構造図におけるインタフェースは下部のモジュールから上位のモジュールに向かって、必ず1本発生することになる。インタフェースを持たないのは一番上位のモジュールのみであるから、モジュールの個数をNとすると、 $N - 1$ 個となる。求める答えはウとなる。

問124 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125 ウ

平均値を求めるプログラムのS T S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。数字選択は入力データの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。

問126 ア

モジュール分割技法のS T S分割に関する問題である。

アのS T S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。

問127 エ

トランザクション分割に関する問題である。

アのS T S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、データの流りに着目した分割技法で、データの流り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。

問128 エ

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K J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O M T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、データの流りに着目した分割技法で、データの流り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。

D F D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。

問129 ウ

トランザクション分割に関する問題である。

アのS T S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130 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、データの流り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、データの流り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、データの流りに着目した分割技法で、データの流りに分岐がある場合に用いる。

問131 ア

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。

問132 イ

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、繰り返して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。

問133 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。

モジュールの独立性を評価する尺度としては、モジュール強度とモジュール間結合度があり、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134 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動的強度である。従って、求める答えは、機動的強度、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135 ウ

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に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低い。

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。

問136 イ

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。

問137 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138 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。

問139 エ

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。

イの情動的強度は、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。

問140 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141 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142 イ

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。

問143 エ

共通フレームの妥当性確認プロセスに関する問題である。

アの監査プロセスは、選ばれた成果物及びプロセスが、要求事項、計画及び合意に対して、適合しているかどうかを独立して決定する。

イの検証プロセスは、プロセス又はプロジェクトのそれぞれのソフトウェア作業成果物及び／又はサービスが規定要求事項を適切に反映しているかを確認する。

ウの使用性向上プロセスは、支援及び教育訓練の最適化、作業の生産性及び品質の増大、並びに人間の作業状態改良を可能にするため、そしてシステムを利用者が拒絶する機会を少なくするために、利害関係者の関心及びニーズの考慮事項を確実にする。

エの妥当性確認プロセスは、ソフトウェア作業成果物の指定された使用方法に対する要件が満たされていることを確認する。

成果物の利用者の視点から意図された正しいものになっているかどうかを確認するプロセスは妥当性確認プロセスである。求める答えはエとなる。

問144 エ

カプセル化に関する問題である。

オブジェクトの状態と振る舞いを一体化することをカプセル化といい、カプセル化することによって状態と振る舞いをブラックボックス化することを情報隠蔽という。

アは汎化、イは継承、ウは抽象クラス、エがカプセル化であり、求める答えはエとなる。

問145 イ

オーバーライドに関する問題である。

オーバーライドはスーパークラスで定義されたメソッドをサブクラスで定義しなおし、動作を上書きすることである。オブジェクト指向プログラミングにおけるポリモーフィズムを実現する際に使われる。オーバーライドを可能とする条件は、引数の型と数が統一されていることである。

オーバーロードは、プログラミング言語において関数や演算子やメソッドの同一名や同一の演算子記号について複数定義し、利用時にプログラムの文脈に応じて選択することで複数の動作を行わせる仕組みである。

アはカプセル化、イはオーバーライド、ウはオーバーロード、エは汎化である。求める答えはイとなる。

問146 ウ

UML技法に関する問題である。

アのオブジェクト図は、ある特定時点でのオブジェクトのインスタンス間の静的な構造を表現したものである。実行時のある状況でのスナップショットである。

イのクラス図は、オブジェクトとそのオブジェクト間の関係を静的に表現したものである。

ウのシーケンス図は、オブジェクトの動的な振る舞いを時系列に沿って表現した図である。求める答えはウとなる。

エ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。時間や行動などに応じて、状態が変化する状況表現する図である。

問147 イ

UML を利用したデータモデルの解釈の問題である。

アは、事業部に所属する部門と所属しない部門が存在するが、部門を管理できるのは事業部であって事業部以外の部門ではない。

イは、社員は部門に所属し、部門の中には事業部に所属するものがあるため、社員は事業部に所属できる。ただし、事業部に所属していない部門の社員は事業部に所属していない。求める答えはイとなる。

ウは、部門には1人以上の社員が所属するため、所属する社員がいない部門は存在しない。

エは、事業部に属さない部門が存在するため部門はいずれかの事業部が管理しているとは言えない。

問148 イ

オブジェクト指向分析のクラスとオブジェクトの関係に関する問題である。

オブジェクト同士の関係で、複数のクラスまたはオブジェクトに共通する性質に着目して抽象化したものがスーパークラスまたはクラスという。これを汎化の関係という。

アの公園とぶらんこは所有の関係である。

イの公園と代々木公園はクラス公園に属するオブジェクト代々木公園であり、クラスとオブジェクトの関係である。汎化の関係とも言う。求める答えはイとなる。

ウの鉄棒とぶらんこは共にクラス公園を構成するオブジェクトである。

エの中之島公園、代々木公園は共にクラス公園に属するオブジェクトである。

問149 イ

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、スカラー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。

アはモジュール強度に関する内容である。

イはデータ結合であり、モジュール結合度が最も弱い。求める答えはイとなる。

ウは外部結合で、共通域を使用して外部宣言したデータを共有する。

エは制御結合で、呼び出すモジュールが呼び出されるモジュールに制御パラメータを渡す。