0001 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：標準

同じ命令セットを実行するCPU AとBがある。Aは平均CPI=2.0、クロック2.4GHz、Bは平均CPI=1.5、クロック2.0GHzである。同一プログラムの命令数が等しいとき、実行時間の比較として正しいものはどれか。

ア．Aの方が約10%短い。
イ．両者の実行時間は等しい。
ウ．クロック周波数が高いAが必ず速い。
エ．Bの方が約10%短い。

答え・解説 正答：エ

ア：CPIだけでなくクロック周波数も含めて評価するとAの方が長い。
イ：2.0/2.4と1.5/2.0は等しくない。
ウ：性能はクロック周波数だけでは決まらず、CPIや命令数にも依存する。
エ：実行時間は命令数×CPI÷クロック周波数に比例する。Aは2.0/2.4=0.833、Bは1.5/2.0=0.75なのでBはAの0.90倍である。
総合解説：CPU性能比較では周波数単独ではなく、命令数・CPI・クロック周期を組み合わせる。アーキテクチャ変更で命令数が変わる場合は、その影響も含めて比較する。

対象：2026年度 / 基準日 2026-09-01

0002 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：応用

キャッシュメモリの効果を高める「空間局所性」の説明として最も適切なものはどれか。

ア．一度参照したデータが短時間のうちに再び参照されやすい性質
イ．主記憶の容量が大きいほどキャッシュミス率が必ず下がる性質
ウ．複数CPUが同じキャッシュを共有すると常に性能が向上する性質
エ．あるデータを参照した直後に、その近くのアドレスのデータも参照されやすい性質

答え・解説 正答：エ

ア：これは時間局所性である。
イ：主記憶容量の増加自体が局所性を生むわけではない。
ウ：共有キャッシュには競合やコヒーレンスの影響もあり、常に向上するとは限らない。
エ：連続する命令や配列要素のアクセスでは近接アドレスが続けて参照されやすく、キャッシュライン単位の読込みが有効になる。
総合解説：局所性はキャッシュ設計の基本で、時間局所性は「同じものを再利用」、空間局所性は「近いものを続けて利用」と整理する。

対象：2026年度 / 基準日 2026-09-01

0003 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：基礎

ある処理時間の80%を完全に並列化でき、残り20%は逐次処理のままである。並列部分を無限に高速化できると仮定した場合、理論上の最大速度向上率はどれか。

ア．逐次部分20%が下限となるので、理論上の最大速度向上率は5倍である。
イ．並列部分80%だけの逆数を用いて、最大1.25倍とみなす。
ウ．並列化率80%から単純に4倍とみなし、逐次部分の影響を無視する。
エ．並列部分を無限に高速化できるので、全体の速度向上率も無限大になる。

答え・解説 正答：ア

ア：並列部分の時間が0に近づいても逐次部分20%は残るため、速度向上率の上限は1/0.2=5倍である。
イ：これは並列部分80%を単純に逆数化した値ではなく、上限評価として不適切である。
ウ：並列化率80%だから4倍という計算にはならない。逐次部分がボトルネックとなる。
エ：逐次部分20%が残るので全体の実行時間は0にはならない。
総合解説：並列化ではコア数を増やしても逐次部分が速度向上の上限を決める。システム設計では並列化可能割合を見積もることが重要である。

対象：2026年度 / 基準日 2026-09-01

0004 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：基礎

命令パイプラインで、直前の命令が計算したレジスタ値を次の命令がすぐ必要とするため待ちが発生している。最も直接的な対策はどれか。

ア．分岐予測を無効にする。
イ．フォワーディングによって演算結果を後続命令へ直接渡す。
ウ．命令キャッシュを無効にする。
エ．クロック周波数を下げる。

答え・解説 正答：イ

ア：分岐予測は制御ハザードへの対策であり、依存データの受渡しとは別である。
イ：結果をレジスタへ書き戻す前に後続段へ転送することで、RAW型のデータハザードによる待ちを減らせる。
ウ：命令キャッシュは命令フェッチ時間に関係し、レジスタ依存の解消にはならない。
エ：タイミング余裕は増えても命令間の論理的なデータ依存は解消しない。
総合解説：パイプラインハザードには構造・データ・制御がある。対策は原因に対応させ、データ依存にはフォワーディングやストール、制御には分岐予測などを用いる。

対象：2026年度 / 基準日 2026-09-01

0005 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：応用

一般的なRISCアーキテクチャの設計思想として最も適切なものはどれか。

ア．1命令で複雑な処理を完結させる命令をできるだけ増やす。
イ．主記憶アクセスを全命令で自由に行えるようにする。
ウ．比較的単純で規則的な命令を用い、パイプライン処理を行いやすくする。
エ．命令長を必ず可変長にしてコード密度を最優先する。

答え・解説 正答：ウ

ア：これは典型的にはCISC寄りの考え方である。
イ：RISCではロード/ストア型など、メモリアクセス命令を限定する設計が多い。
ウ：RISCは命令形式や実行を規則化し、単純な命令を高速に処理しやすい設計思想である。
エ：RISCは固定長や規則的な形式を採用することが多く、「必ず可変長」ではない。
総合解説：現代CPUはRISC/CISCの要素を内部で組み合わせる場合もある。試験では設計思想として、命令の単純化・規則化・パイプライン親和性を押さえる。

対象：2026年度 / 基準日 2026-09-01

0006 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：基礎

32ビット値0x12345678を、最下位アドレスから順にメモリへ格納したとき、リトルエンディアンのバイト列はどれか。

ア．最下位バイトから並べるため、メモリ順は0x78→0x56→0x34→0x12となる。
イ．最上位バイトから並べる0x12→0x34→0x56→0x78はビッグエンディアンである。
ウ．最上位と最下位だけを入れ替え、12、78、34、56の順に格納する。
エ．16ビット単位だけを入れ替え、56、78、12、34の順に格納する。

答え・解説 正答：ア

ア：リトルエンディアンでは最下位バイト0x78を最小アドレスに置き、その後0x56,0x34,0x12の順になる。
イ：これはビッグエンディアンの並びである。
ウ：バイト順序を端から順に並べる規則になっていない。
エ：16ビット単位の入替えであり、リトルエンディアンの定義とは異なる。
総合解説：異機種間でバイナリデータを交換するときはエンディアン差を考慮する。ネットワークプロトコルやファイル形式ではバイト順を仕様で固定する。

対象：2026年度 / 基準日 2026-09-01

0007 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：標準

DMA（Direct Memory Access）を用いる主な目的として最も適切なものはどれか。
ア．CPUの命令セットを実行時に書き換える。
イ．I/O装置と主記憶のデータ転送をDMAコントローラ等に任せ、CPUの転送処理負荷を減らす。
ウ．主記憶の内容を常に暗号化する。
エ．キャッシュヒット率を必ず100%にする。

答え・解説 正答：イ

ア：DMAはI/O転送機構であり、命令セット変更とは無関係である。
イ：CPUが1語ずつ転送する代わりに、DMA機構がまとまった転送を行い、CPUは他の処理に使える。
ウ：暗号化はDMAの本来目的ではない。
エ：DMAとキャッシュには整合性の考慮が必要な場合があるが、ヒット率を保証するものではない。
総合解説：DMAは大量I/O転送の効率化に有効だが、CPUキャッシュとDMA先メモリの整合性など、アーキテクチャ上の注意もある。

対象：2026年度 / 基準日 2026-09-01

0008 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：標準

一般的なメモリ階層について、CPUに近い側から遠い側へ移るときの傾向として正しいものはどれか。
ア．平均アクセス時間は短くなり、容量も小さくなる。
イ．平均アクセス時間は長くなり、容量は大きくなる傾向がある。
ウ．平均アクセス時間は一定で、容量だけが大きくなる。
エ．容量は一定で、ビット単価だけが必ず高くなる。

答え・解説 正答：イ

ア：CPUから遠い階層ほど一般に低速で、容量は大きくなる。
イ：レジスタやキャッシュは高速・小容量、主記憶や補助記憶は相対的に低速・大容量という階層を構成する。
ウ：階層ごとに技術・コストが異なるためアクセス時間も変化する。
エ：容量は一定ではなく、遠い階層ほど大きくなる傾向がある。
総合解説：メモリ階層は高速性・容量・コストを両立するための設計である。局所性を利用して、見かけ上の平均アクセス時間を短縮する。

対象：2026年度 / 基準日 2026-09-01

0009 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：標準

8コアCPUへ移行したが、共有ロックで保護された処理が頻繁に競合し、CPU使用率も一部コアだけ高い。性能改善策として最も適切なものはどれか。

ア．クロックを下げてロック待ち時間を短縮する。
イ．全処理を1本のグローバルロックで保護する。
ウ．キャッシュを無効化してコア間共有を避ける。
エ．共有状態とロック粒度を見直し、並列実行可能な区間を増やす。

答え・解説 正答：エ

ア：クロック低下は処理時間を延ばす方向で、競合そのものを解消しない。
イ：直列化範囲が増え、並列性能はさらに悪化しやすい。
ウ：キャッシュ無効化は大きな性能低下につながり、共有ロックの論理的競合を解消しない。
エ：コア数だけ増やしても共有ロックが直列化点になる。競合を減らし並列度を高める設計が必要である。
総合解説：マルチコア性能はスレッド数だけでなく、共有資源・ロック・メモリ帯域などで制約される。ボトルネックを測定し、直列部分を減らす。

対象：2026年度 / 基準日 2026-09-01

0010 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：標準

一般的なECCメモリの説明として、単純な1ビットパリティと比べた特徴はどれか。

ア．すべてのビット誤りを必ず訂正できる。
イ．方式によっては1ビット誤りを訂正し、複数ビット誤りを検出できる。
ウ．誤り検出はできるが訂正は一切できない。
エ．記憶容量を増やす技術であり信頼性とは無関係である。

答え・解説 正答：イ

ア：ECCにも訂正・検出能力の限界があり、任意個数の誤りを保証して訂正できるわけではない。
イ：代表的なSEC-DED型ECCでは1ビット訂正・2ビット検出が可能で、単純パリティより高い保護能力を持つ。
ウ：これは単純パリティに近い説明で、ECCは方式により訂正も可能である。
エ：ECCの主目的はメモリ内データの誤り検出・訂正による信頼性向上である。
総合解説：信頼性設計では「検出」と「訂正」を区別する。ECCの能力は符号方式に依存するため、要求信頼性に合わせて選定する。

対象：2026年度 / 基準日 2026-09-01

0011 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：標準

オンライン注文システムで「1秒当たり500件の注文を処理できること」と「各注文の95%が2秒以内に応答すること」を要求している。前者と後者の指標の組合せとして正しいものはどれか。

ア．前者はレスポンスタイム、後者はスループット。
イ．前者はスループット、後者はレスポンスタイム。
ウ．前者も後者も可用性。
エ．前者も後者もMTBF。

答え・解説 正答：イ

ア：定義が逆である。
イ：単位時間当たりの処理件数はスループット、個々の要求に対する応答時間はレスポンスタイムである。
ウ：可用性はサービスが利用可能な時間割合などを表す指標である。
エ：MTBFは故障間平均時間であり、性能指標ではない。
総合解説：性能要件は処理量と待ち時間を分けて定義する。高スループットでも個々の応答が遅い場合があり、両方の測定が必要である。

対象：2026年度 / 基準日 2026-09-01

0012 コンピュータ・システム技術 > コンピュータアーキテクチャ・性能 ｜ 難易度：基礎

単一のサーバで平均到着率が処理能力に近づいている。処理時間のばらつきもある場合、一般に利用率が100%へ近づくと何が起こりやすいか。

ア．待ち行列が急激に伸び、平均応答時間が大きく悪化する。
イ．待ち時間は0へ近づく。
ウ．処理能力が自動的に到着率を上回る。
エ．スループットは必ず無限に増加する。

答え・解説 正答：ア

ア：余裕容量が小さくなると一時的な到着集中を吸収できず、待ちが蓄積しやすい。
イ：利用率上昇は空き時間を減らすため、待ち時間は通常増加する。
ウ：固定された単一サーバの処理能力は利用率上昇だけでは増えない。
エ：処理能力には上限があり、到着率がそれを超えるとキューが発散する。
総合解説：性能設計ではピーク時に一定の余裕を持たせる。平均値だけでなく、到着のばらつき・サービス時間・キュー長・パーセンタイル応答時間を確認する。

対象：2026年度 / 基準日 2026-09-01

0013 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：標準

同一プロセス内の複数スレッドについて、一般的に正しい説明はどれか。
ア．各スレッドは必ず完全に独立した仮想アドレス空間を持つ。
イ．スレッド間では共有メモリを一切利用できない。
ウ．同一プロセスのアドレス空間などを共有しつつ、各スレッドは独自の実行コンテキストを持つ。
エ．スレッド切替えは必ずプロセス切替えより重い。

答え・解説 正答：ウ

ア：独立アドレス空間は通常プロセス単位であり、同一プロセス内スレッドは共有する。
イ：同一プロセスのスレッドはメモリを共有できる。
ウ：スレッドはコードやヒープ等を共有するため通信が軽量だが、共有データの同期が必要になる。
エ：一般に同一プロセス内スレッドの切替えはプロセス切替えより軽量になりやすい。
総合解説：スレッド化は並行性を高める一方、データ競合やデッドロックを招く。設計では共有範囲と同期方式を明確にする。

対象：2026年度 / 基準日 2026-09-01

0014 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：標準

デマンドページング方式で、参照した仮想ページが主記憶に存在しないときに発生する処理として最も適切なものはどれか。
ア．CPUキャッシュだけをクリアしてそのまま命令を継続する。
イ．必ずプロセスを異常終了する。
ウ．ページテーブルを使わず物理アドレスを直接生成する。
エ．ページフォールトを発生させ、必要なページを補助記憶から主記憶へ読み込む。

答え・解説 正答：エ

ア：主記憶にページがないため、キャッシュクリアだけでは参照できない。
イ：正当な仮想ページであれば通常はページインして継続する。無効アドレスの場合は別である。
ウ：仮想記憶ではページテーブル等の変換機構を用いる。
エ：OSはページテーブルを確認し、未在のページなら例外を処理してページを読み込んだ後、命令を再開する。
総合解説：ページフォールトは通常アクセスより桁違いに高コストである。ワーキングセット不足による頻発はスラッシングにつながる。

対象：2026年度 / 基準日 2026-09-01

0015 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：基礎

対話型システムで、各プロセスに短いCPU時間を順番に割り当て、一定時間を使い切ると次のプロセスへ切り替える方式はどれか。

ア．FCFSだけを用いた非プリエンプティブ方式
イ．ラウンドロビン
ウ．デッドラインモノトニック
エ．LRU

答え・解説 正答：イ

ア：先着順で長い処理がCPUを占有すると後続の応答が悪化しやすい。
イ：タイムクォンタムを用いて順番にCPUを割り当てる方式で、対話型処理の応答性を確保しやすい。
ウ：リアルタイムスケジューリングの一種で、設問の説明とは異なる。
エ：LRUは主にページ置換やキャッシュ置換の方針であり、CPUのラウンドロビンスケジューリングではない。
総合解説：スケジューリング方式は、応答時間・スループット・公平性・期限保証などの要件に応じて選ぶ。

対象：2026年度 / 基準日 2026-09-01

0016 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：標準

複数スレッドが同時に更新すると不整合になる共有データに対して、同時に1スレッドだけをクリティカルセクションへ入れたい。最も直接的な仕組みはどれか。

ア．ミューテックスによる排他制御
イ．カウンティングセマフォの初期値を10にする。
ウ．ページングを無効化する。
エ．DNSキャッシュを利用する。

答え・解説 正答：ア

ア：ミューテックスは所有権を持つ排他ロックとして、クリティカルセクションへの同時進入を防ぐ。
イ：10スレッドまで同時進入できるため、1スレッドだけという要件に合わない。
ウ：メモリ管理方式は共有データの排他制御を実現しない。
エ：DNSキャッシュは名前解決用で、スレッド同期とは無関係である。
総合解説：排他制御は共有資源の競合を防ぐ基本機構である。ロック範囲が広すぎると性能低下、取得順序が不適切だとデッドロックを招く。

対象：2026年度 / 基準日 2026-09-01

0017 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：標準

多数のオンライン要求を受ける業務システムで、アプリケーションとDBの間に置くトランザクション処理ミドルウェアの役割として適切なものはどれか。

ア．業務データの意味を自動的に決定して要件定義を不要にする。
イ．物理ネットワーク配線を自動的に変更する。
ウ．CPUの命令デコーダを置き換える。
エ．要求の受付・実行資源の管理・トランザクション制御などを共通化する。

答え・解説 正答：エ

ア：ミドルウェアは業務要件そのものを自動決定するものではない。
イ：配線変更はトランザクションミドルウェアの役割ではない。
ウ：CPU内部機構ではなくソフトウェア層の共通基盤である。
エ：TPモニタ等は多数要求の振分け、プロセス/スレッド資源管理、トランザクション境界管理などを提供する。
総合解説：ミドルウェアはOSとアプリケーションの間で共通機能を提供し、アプリ側の実装を簡素化する。可用性・性能・運用性にも影響するため方式設計で評価する。

対象：2026年度 / 基準日 2026-09-01

0018 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：標準

フル仮想化におけるハイパーバイザの主な役割はどれか。

ア．各アプリケーションの業務ルールだけを管理する。
イ．DNSゾーンを権威サーバへ配布する。
ウ．DBの正規化を自動実行する。
エ．物理CPU・メモリ・ネットワーク・ストレージ等を抽象化し、複数VMへ仮想資源として提供する。

答え・解説 正答：エ

ア：業務ルール管理はハイパーバイザの主機能ではない。
イ：DNS運用の機能であり仮想化基盤の役割ではない。
ウ：データモデリングとは無関係である。
エ：ハイパーバイザは物理資源を仲介・抽象化し、複数のゲストOSを分離して実行させる。
総合解説：仮想化では、物理資源→ハイパーバイザ→VM→ゲストOS→アプリという層を理解する。管理面ではハイパーバイザ自体が重要な保護対象となる。

対象：2026年度 / 基準日 2026-09-01

0019 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：基礎

一般的なVMとOSコンテナを比較した説明として最も適切なものはどれか。

ア．コンテナは必ずVMより強い隔離を提供する。
イ．VMでは複数の異なるゲストOSを実行できない。
ウ．コンテナはアプリケーションのパッケージングには利用できない。
エ．VMは通常ゲストOSを含み、コンテナはホストのOSカーネルを共有するため起動や配置が軽量になりやすい。

答え・解説 正答：エ

ア：隔離強度は実装・設定に依存し、共有カーネルを持つ点から一概に「必ず強い」とはいえない。
イ：ハイパーバイザ上で異なるゲストOSを動かせることはVMの一般的な用途である。
ウ：コンテナはアプリと依存関係をまとめて可搬化する用途に広く使われる。
エ：コンテナはOSレベルの仮想化で、完全なゲストOSを各インスタンスに持たない構成が一般的である。
総合解説：方式選定では、起動時間・密度・隔離・OS互換性・運用ツールを比較する。コンテナとVMを組み合わせる構成も一般的である。

対象：2026年度 / 基準日 2026-09-01

0020 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：応用

仮想ディスクのCopy-on-Write方式スナップショットを取得した後にデータを書き換える場合の一般的な動作として適切なものはどれか。

ア．スナップショット取得時に必ず全ディスクを即座に完全コピーする。
イ．取得後の書込みをすべて禁止する。
ウ．元の時点を参照できる情報を保ちながら、変更されたブロックを別領域へ記録する。
エ．CPUレジスタの内容だけを保存し、ストレージは保存しない。

答え・解説 正答：ウ

ア：これはフルコピー方式の説明で、CoWの典型的な特徴とは異なる。
イ：CoWスナップショットは通常、元ボリュームへの書込みを継続しながら差分を管理する。
ウ：CoWは変更時に差分を別の場所へ記録することで、スナップショット時点の状態を再構成できるようにする。
エ：ディスクスナップショットはストレージ状態を対象とする。
総合解説：スナップショットはバックアップと同一ではない。障害ドメインや保存先が同じ場合、基盤障害から復旧できない可能性があるため別バックアップも検討する。

対象：2026年度 / 基準日 2026-09-01

0021 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：応用

管理対象ランタイムを使うAPIサーバで、平均応答は速いが数分ごとに大きな停止時間が発生し、99.9パーセンタイル応答が悪化している。GCログで長い停止が確認された。最初に行うべき対応として適切なものはどれか。

ア．GC方式・ヒープ使用量・割当て率・停止時間を計測し、低遅延要件に合う設定や実装へ調整する。
イ．平均応答時間だけを見て問題なしと判断する。
ウ．すべてのログを停止して原因を隠す。
エ．DBの主キーを削除する。

答え・解説 正答：ア

ア：高パーセンタイル遅延の原因がGC停止と確認できているため、メモリ割当てやGC方式を含めて測定に基づき調整する。
イ：SLOが高パーセンタイルで定義されているなら平均値だけでは評価できない。
ウ：観測性を失い、原因分析と再発防止が困難になる。
エ：GC停止とは直接関係がなく、データ整合性も損ない得る。
総合解説：性能問題では平均値だけでなくテールレイテンシを確認する。ランタイム、I/O、ロック、GCなど層ごとの観測データからボトルネックを切り分ける。

対象：2026年度 / 基準日 2026-09-01

0022 コンピュータ・システム技術 > OS・ミドルウェア・仮想化 ｜ 難易度：基礎

物理ホスト8コアに、各4 vCPUのVMを8台配置している。普段は低負荷だが月末に全VMが同時にCPUを多用し、応答が急激に悪化する。最も妥当な説明はどれか。

ア．平常時の利用率を前提にvCPUをオーバーコミットしており、同時ピーク時に物理CPU競合が顕在化した。
イ．vCPU数は物理CPU性能と無関係なのでCPU競合は起こらない。
ウ．VMを増やすほど1台当たりのCPU性能は必ず向上する。
エ．原因はIPv6アドレスが128ビットであることに限定される。

答え・解説 正答：ア

ア：仮想CPU総数が物理コア数を大きく上回っていても低負荷時は動作するが、同時高負荷ではスケジューリング待ちが増える。
イ：vCPUは最終的に物理CPUで実行されるため競合は発生し得る。
ウ：物理資源を共有するため、競合すれば逆に低下する。
エ：CPUスケジューリング競合の事象とIPv6アドレス長は無関係である。
総合解説：仮想化の集約率は平均利用率だけでなく同時ピーク、CPU ready相当の待ち、メモリ圧迫、I/O競合を考慮して決める。

対象：2026年度 / 基準日 2026-09-01

0023 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：基礎

独立した二つの構成要素AとBが直列に接続され、両方が稼働しているときだけサービス可能である。Aの可用性0.99、Bの可用性0.98のとき、全体可用性はいくらか。

ア．停止確率を掛けた値を可用性とみなし、0.9998とする。
イ．二つの可用性を単純平均し、0.985とする。
ウ．直列構成なので両方が稼働する確率を掛け、0.9702とする。
エ．各値を先に小数第2位へ丸め、全体可用性を0.97とする。

答え・解説 正答：ウ

ア：これは停止確率を単純に掛けた値を可用性と取り違えた計算である。
イ：単純平均では直列依存を表せない。
ウ：直列構成では両方が同時に利用可能である確率なので0.99×0.98=0.9702となる。
エ：概算として近いが、与えられた値からの正確な積は0.9702である。
総合解説：可用性は構成依存で決まる。直列依存は積、独立冗長の並列系は全系停止確率から求めるなど、障害条件を明確にする。

対象：2026年度 / 基準日 2026-09-01

0024 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：応用

MTBFが990時間、MTTRが10時間の修理可能システムについて、単純モデルでの定常可用性として最も近いものはどれか。

ア．停止率を約10%とみなし、可用性を90%とする。
イ．990を稼働率99.9%と読み替え、可用性を99.9%とする。
ウ．MTBF/(MTBF+MTTR)で計算し、990/1000=99%とする。
エ．MTTRの比率だけを用い、可用性を1%とする。

答え・解説 正答：ウ

ア：MTTRの割合を過大に見積もっている。
イ：990/1000は0.99であり0.999ではない。
ウ：可用性はMTBF/(MTBF+MTTR)=990/(990+10)=0.99となる。
エ：これは停止時間比率に近く、可用性ではない。
総合解説：可用性改善には故障間隔を伸ばすだけでなく、検知・切替え・修理・復旧を高速化してMTTRを短くすることも有効である。

対象：2026年度 / 基準日 2026-09-01

0025 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：基礎

Active-Active構成を採用する主な利点と注意点の組合せとして適切なものはどれか。
ア．待機系を一切稼働させないため運用が必ず単純になる。
イ．平常時から複数系を処理に使える一方、状態同期や整合性、障害時の負荷再配分を設計する必要がある。
ウ．データ同期を考慮しなくてよいので障害切替えが常に無停止になる。
エ．冗長化しているので容量設計は不要になる。

答え・解説 正答：イ

ア：これはActive-Standbyに近く、Active-Activeの説明ではない。
イ：複数系を同時利用できるため資源効率や耐障害性を高められるが、共有状態や分散整合性が複雑になる。
ウ：状態を持つシステムでは同期やセッション、DB整合性を検討する必要がある。
エ：片系障害時に残存系が負荷を処理できるか容量設計が必要である。
総合解説：冗長構成では「冗長にしたこと」だけでなく、障害検知、切替え、データー貫性、フェイルバック、片系時容量まで設計する。

対象：2026年度 / 基準日 2026-09-01

0026 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：標準

同容量ディスク4台でRAID 6を構成する場合の一般的な特徴として正しいものはどれか。
ア．2台分相当をパリティに用い、同時に2台までのディスク故障に耐えられる。
イ．全ディスクへ同じ内容を複製するため実効容量は1台分になる。
ウ．パリティを持たないため1台故障でもデータを失う。
エ．1台分のパリティだけを用い、2台同時故障にも必ず耐える。

答え・解説 正答：ア

ア：RAID 6は二重パリティを用い、N台なら概ねN-2台分の実効容量を得る。
イ：これは全台ミラーに近く、RAID 6の方式ではない。
ウ：これはRAID 0の特徴である。
エ：単一パリティはRAID 5に相当し、通常2台同時故障には耐えられない。
総合解説：RAIDは可用性を高めるがバックアップの代替ではない。誤削除、論理破損、ランサムウェア、筐体全損などには別の保護が必要である。

対象：2026年度 / 基準日 2026-09-01

0027 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：応用

災害対策要件で「障害発生から4時間以内に業務を再開」「失ってよいデータは最大15分分」と定めた。RTOとRPOの対応として正しいものはどれか。

ア．RTO=4時間、RPO=15分
イ．RTO=15分、RPO=4時間
ウ．RTO=4時間、RPO=4時間
エ．RTO=15分、RPO=15分

答え・解説 正答：ア

ア：RTOは復旧までの目標時間、RPOは許容するデータ損失を時間軸で表した目標復旧時点である。
イ：二つの定義が逆である。
ウ：RPOはデータ損失許容量15分に対応する。
エ：RTOは業務再開まで4時間という要件に対応する。
総合解説：バックアップやレプリケーション方式はRPO、復旧自動化・待機サイト・切替え手順はRTOに大きく影響する。両方を業務要求から決める。

対象：2026年度 / 基準日 2026-09-01

0028 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：標準

Webサーバは2台、DBサーバも2台で冗長化されているが、全通信が1台のロードバランサを経由する。可用性設計上の最優先課題はどれか。

ア．Webサーバを1台に減らす。
イ．DBの主キーを削除する。
ウ．DNSのTTLを無限にする。
エ．ロードバランサが単一障害点なので、装置や経路を冗長化する。

答え・解説 正答：エ

ア：冗長性を下げるため可用性改善にならない。
イ：単一障害点の解消とは無関係で、整合性も損なう。
ウ：TTLを極端に長くすると切替え反映が遅くなる可能性があり、LBのSPOFを除去しない。
エ：Web/DBが冗長でも入口が1台なら、その故障で全サービスが停止する。
総合解説：エンドツーエンドで障害経路を追い、電源・ネットワーク・LB・認証・ストレージ・監視などの隠れたSPOFも洗い出す。

対象：2026年度 / 基準日 2026-09-01

0029 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：標準

一部機能に障害が発生したとき、安全性を損なわない範囲で重要機能だけを継続し、性能や機能を縮退させてサービスを続ける考え方はどれか。

ア．フェイルセーフだけ
イ．フェイルソフト（縮退運転）
ウ．フールプルーフ
エ．チェックサム

答え・解説 正答：イ

ア：フェイルセーフは故障時に安全側へ移行する考え方で、必ずサービス継続を意味しない。
イ：故障時に全停止せず、機能や性能を落として継続する考え方である。
ウ：誤操作しても危険な状態にしにくい設計で、障害時の縮退運転とは異なる。
エ：データ誤り検出の手段であり、システム運転方針の名称ではない。
総合解説：重要システムでは、安全性・可用性・継続性の優先順位を要件化し、停止・縮退・代替手段の条件を事前に定義する。

対象：2026年度 / 基準日 2026-09-01

0030 コンピュータ・システム技術 > システム構成・信頼性 ｜ 難易度：標準

日次フルバックアップだけではRPOが24時間近くになる業務DBで、RPOを15分へ短縮したい。最も直接的な改善策はどれか。

ア．トランザクションログ等を短い間隔で別領域へ保全し、時点復旧できる構成にする。
イ．フルバックアップを月1回に減らす。
ウ．バックアップを同じ単一ディスクだけに保存する。
エ．RPOを短くする代わりに監視を停止する。

答え・解説 正答：ア

ア：フルバックアップ間の変更をログで継続保全すれば、障害直前に近い時点へ復旧できる。
イ：RPOはさらに悪化する。
ウ：ディスク障害で本体とバックアップを同時に失う危険がある。
エ：監視停止はRPO改善にならず障害検知も遅れる。
総合解説：復旧設計はバックアップ頻度だけでなく、保管先、ログ適用、復旧試験、整合性確認まで含める。RPO/RTOを実測で検証する。

対象：2026年度 / 基準日 2026-09-01

0031 データ・データベース > データモデリング・正規化 ｜ 難易度：基礎

関係表で行を一意に識別できる最小の属性集合を何というか。
ア．外部キー
イ．非キー属性
ウ．導出属性
エ．候補キー

答え・解説 正答：エ

ア：外部キーは他表（または同表）の候補キー等を参照し、参照整合性に用いる。
イ：候補キーを構成しない属性の総称であり、一意識別を保証しない。
ウ：他属性から計算できる属性を指し、キーの定義ではない。
エ：候補キーは一意性を持ち、かつ不要な属性を含まない最小のキーである。候補キーの一つを主キーとして選ぶ。
総合解説：キー設計では一意性だけでなく最小性も確認する。スーパーキーは余分な属性を含んでもよいが、候補キーは最小である。

対象：2026年度 / 基準日 2026-09-01

0032 データ・データベース > データモデリング・正規化 ｜ 難易度：標準

学生は複数の科目を履修でき、各科目も複数の学生が履修する。さらに履修ごとに成績を保持したい。論理データモデルとして最も適切なものはどれか。
ア．学生表に科目1、科目2、科目3という繰返し列を固定で追加する。
イ．学生と科目の間に「履修」エンティティを設け、学生ID・科目IDと成績を持たせる。
ウ．科目表に全学生名を1列の文字列として連結保存する。
エ．成績を学生表の1列だけに保存する。

答え・解説 正答：イ

ア：科目数が可変で拡張性が低く、繰返し項目を生む。
イ：多対多関係を連関エンティティへ分解し、関係自体の属性である成績を履修に保持する。
ウ：原子性や検索・整合性の面で不適切である。
エ：成績は学生と科目の組合せに依存するので学生単独では特定できない。
総合解説：多対多関係は連関エンティティで表現し、その関係に属する属性を持たせる。業務上の一意性や履修時点なども検討する。

対象：2026年度 / 基準日 2026-09-01

0033 データ・データベース > データモデリング・正規化 ｜ 難易度：標準

第1正規形（1NF）にするための基本的な考え方として最も適切なものはどれか。
ア．各属性値を関係モデル上の単一の値として扱い、繰返し項目を別行・別表などへ整理する。
イ．すべての非キー属性を主キーの一部だけに依存させる。
ウ．すべての表を1列だけにする。
エ．外部キーを一切使わない。

答え・解説 正答：ア

ア：1NFでは1セルに複数値を詰め込むような繰返し群を避け、関係として扱える形にする。
イ：これは部分関数従属を残す方向で、2NFの考え方に反する。
ウ：1NFは列数を1にする規則ではない。
エ：1NFと外部キー利用可否は別の論点である。
総合解説：正規化は重複と更新異常を減らすために、関数従属を基に表を分解する。1NFはその出発点である。

対象：2026年度 / 基準日 2026-09-01

0034 データ・データベース > データモデリング・正規化 ｜ 難易度：応用

受注明細(受注番号, 商品番号, 受注日, 商品名, 数量)で、主キーは(受注番号, 商品番号)、受注日は受注番号だけに、商品名は商品番号だけに依存する。2NF化として適切なものはどれか。
ア．受注(受注番号,受注日)、商品(商品番号,商品名)、受注明細(受注番号,商品番号,数量)へ分解する。
イ．元の表のまま、受注日と商品名に索引を追加する。
ウ．受注番号を削除して商品番号だけを主キーにする。
エ．数量を受注表へ移す。

答え・解説 正答：ア

ア：複合主キーの一部だけに依存する属性をそれぞれ独立表へ分け、明細には複合キー全体に依存する数量を残す。
イ：索引は性能対策であり、部分関数従属による更新異常を解消しない。
ウ：同じ商品が複数受注に現れるため明細を一意に識別できない。
エ：数量は受注と商品の組合せに依存するため受注単独には置けない。
総合解説：2NFは複合候補キーの一部への部分関数従属を排除する。キーが単一属性なら部分従属は生じないが、3NF以降の検討は別途必要である。

対象：2026年度 / 基準日 2026-09-01

0035 データ・データベース > データモデリング・正規化 ｜ 難易度：標準

社員(社員ID, 部門ID, 部門名)で社員ID→部門ID、部門ID→部門名が成り立つ。3NF化として適切な分解はどれか。

ア．社員(社員ID,部門ID)と部門(部門ID,部門名)に分ける。
イ．社員表に部門名を2列複製する。
ウ．部門IDを削除し、部門名だけを社員表へ残す。
エ．社員IDと部門名を連結して1列にする。

答え・解説 正答：ア

ア：部門名は社員IDから部門IDを介して推移的に依存しているため、部門情報を分離する。
イ：冗長性と更新異常を増やす。
ウ：部門を安定して識別するキーを失い、名称変更の影響も大きくなる。
エ：関数従属の問題を解決せず、検索・更新もしにくくなる。
総合解説：3NFでは非キー属性間の推移的依存を排除する。マスタ情報を独立させることで部門名変更などの更新箇所を一元化できる。

対象：2026年度 / 基準日 2026-09-01

0036 データ・データベース > データモデリング・正規化 ｜ 難易度：基礎

注文表の顧客IDが顧客表の顧客IDを参照する外部キーである。参照整合性制約の主な目的はどれか。

ア．注文金額が必ず正数であることを保証する。
イ．存在しない顧客を参照する注文など、孤立した参照関係を防ぐ。
ウ．顧客名の重複を必ず禁止する。
エ．全検索を必ず索引アクセスにする。

答え・解説 正答：イ

ア：これはCHECK制約等で表す値域制約の論点である。
イ：外部キー値は参照先の候補キー値として存在するか、許可される場合はNULLであることを要求できる。
ウ：顧客名の一意性は別途UNIQUE等で定義する。
エ：参照整合性は論理的整合性の制約で、実行計画を保証するものではない。
総合解説：データ整合性にはキー制約、参照整合性、NOT NULL、CHECKなどがある。業務ルールをDB制約として表現できる範囲を検討する。

対象：2026年度 / 基準日 2026-09-01

0037 データ・データベース > データモデリング・正規化 ｜ 難易度：応用

顧客を識別する業務コードが将来の統廃合で変更される可能性がある。多数の子表が顧客を参照する設計で、識別子の変更波及を小さくしたい。適切な方針はどれか。

ア．安定したサロゲートキーを主たる参照キーとし、業務コードには必要な一意制約を設ける。
イ．業務コードの変更可能性を無視し、全子表に文字列コードを重複保存する。
ウ．主キーを持たない表にする。
エ．顧客名を主キーにし、同姓同名を禁止する。

答え・解説 正答：ア

ア：内部識別子を業務上の可変コードから分離すると、コード変更の外部キー波及を抑えられる。
イ：変更時の更新範囲が広くなり、整合性リスクも増える。
ウ：行の一意識別や参照整合性が困難になる。
エ：顧客名は一般に安定した一意識別子ではない。
総合解説：サロゲートキーには安定性の利点があるが、業務上の一意性制約を失わないよう自然キー相当のUNIQUE制約も設計する。

対象：2026年度 / 基準日 2026-09-01

0038 データ・データベース > データモデリング・正規化 ｜ 難易度：標準

特定DBMSの表領域、索引方式、パーティション方法まで定めるのは、一般にどの段階のデータモデルか。

ア．概念データモデル
イ．物理データモデル
ウ．論理データモデル
エ．業務フロー図

答え・解説 正答：イ

ア：概念モデルは業務上のエンティティや関係を技術実装から独立して表す。
イ：DBMSや格納方式に依存する実装詳細は物理設計で具体化する。
ウ：論理モデルは属性・キー・関係を整理するが、特定DBMSの物理格納詳細は通常含めない。
エ：業務フローは処理の流れを表すもので、物理DB格納設計そのものではない。
総合解説：概念→論理→物理の順に実装具体度が上がる。要件と実装を混同しないことで、DBMS変更や方式比較がしやすくなる。

対象：2026年度 / 基準日 2026-09-01

0039 データ・データベース > データモデリング・正規化 ｜ 難易度：標準

注文明細の数量×単価から算出できる「明細金額」をDBに保存するか検討している。最も適切な判断はどれか。

ア．再計算コストと参照頻度が高い場合は保存も選択肢だが、元データ変更時の整合性維持方法を必ず設計する。
イ．導出できる値はどんな場合でも保存してはならない。
ウ．保存すれば元の数量と単価は不要になる。
エ．保存すると正規化の影響を一切受けず整合性問題もなくなる。

答え・解説 正答：ア

ア：導出値の保存は読取り性能を高める一方、冗長データの更新整合性というコストを生む。
イ：性能や監査要件などから冗長保持が合理的な場合もある。
ウ：算出根拠や他用途のため元データは通常必要である。
エ：冗長値は不整合を生む可能性があるため更新ルールが必要である。
総合解説：正規化は原則だが、性能や履歴要件で意図的に冗長化する場合がある。非正規化は理由・整合性維持・再計算方法を明示する。

対象：2026年度 / 基準日 2026-09-01

0040 データ・データベース > データモデリング・正規化 ｜ 難易度：標準

関係R(A,B,C,D)で A→B、B→C、AC→D が成り立つ。他の関数従属はこれらから導かれるものだけとする。Aの属性閉包A+として正しいものはどれか。

ア．{A,B}
イ．{A,B,C,D}
ウ．{A,B,C}
エ．{A,D}

答え・解説 正答：イ

ア：Bからも導けるため不足している。
イ：A→B、B→CによりAからB,Cが得られ、AとCがそろうのでAC→DからDも得られる。
ウ：AとCからDも導けるため不足している。
エ：AからB,Cも導けるので不足し、導出過程も不完全である。
総合解説：属性閉包は候補キー判定や正規形判定に使う。ある属性集合の閉包が全属性を含み、かつ最小なら候補キーとなる。

対象：2026年度 / 基準日 2026-09-01

0041 データ・データベース > データモデリング・正規化 ｜ 難易度：基礎

BCNF（Boyce-Codd正規形）の条件として最も適切なものはどれか。

ア．すべての属性が数値型である。
イ．非自明な関数従属X→Yが成り立つなら、Xはスーパーキーである。
ウ．外部キーが存在しない。
エ．候補キーが必ず1個だけである。

答え・解説 正答：イ

ア：データ型はBCNFの条件ではない。
イ：BCNFは決定項が必ずスーパーキーであることを求め、3NFより厳しい。
ウ：外部キーの有無はBCNF判定条件ではない。
エ：複数候補キーがあってもBCNFを満たし得る。
総合解説：3NFでは一部の従属に例外を認めるが、BCNFは決定項がスーパーキーであることを要求する。分解時は無損失結合と従属性保存も考慮する。

対象：2026年度 / 基準日 2026-09-01

0042 データ・データベース > データモデリング・正規化 ｜ 難易度：基礎

分析用データマートで、更新は夜間バッチのみ、日中は同じ集計を大量に参照し、複雑なJOINが応答遅延の主因になっている。設計方針として妥当なものはどれか。

ア．正規化された元データを維持しつつ、分析用途では集計表や非正規化モデルを別途用意することを検討する。
イ．OLTPの全表から主キーと外部キーを削除する。
ウ．JOINがあるという理由だけでDBを使わず全てテキストファイルにする。
エ．正規化は常に最大性能なので物理設計は一切不要である。

答え・解説 正答：ア

ア：更新頻度が低く読取り中心なら、ETLで整合性を管理しながら分析向けに冗長化する合理性がある。
イ：整合性を損ない、分析性能改善の一般解ではない。
ウ：可用性・整合性・検索性など別要件を無視している。
エ：正規化は論理整合性に有効だが、実行性能は索引・集計・パーティション等の物理設計にも依存する。
総合解説：OLTPと分析ではアクセス特性が異なる。正規化原則を理解した上で、目的別ストアやデータマートを設けるのがアーキテクトの判断となる。

対象：2026年度 / 基準日 2026-09-01

0043 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：標準

トランザクションが正常終了した後、その更新結果が障害後も失われないことを表すACID特性はどれか。
ア．Atomicity（原子性）
イ．Durability（永続性）
ウ．Consistency（一貫性）
エ．Isolation（独立性）

答え・解説 正答：イ

ア：全て実行されるか全て取り消されるかを保証する性質である。
イ：コミット済み結果を永続化し、障害復旧後も保持する性質である。
ウ：トランザクション前後で定義された整合性制約を満たす性質である。
エ：並行トランザクションの干渉を制御し、適切な分離を保つ性質である。
総合解説：ACIDはトランザクション設計の基本で、原子性・一貫性・独立性・永続性を区別して理解する。

対象：2026年度 / 基準日 2026-09-01

0044 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：基礎

トランザクションT1が更新したがまだコミットしていない値をT2が読み、その後T1がロールバックした。この現象は何か。
ア．ノンリピータブルリード
イ．ファントムリード
ウ．チェックポイント
エ．ダーティリード

答え・解説 正答：エ

ア：同一行を再読したとき、他のコミット済み更新により値が変わる現象である。
イ：同じ条件検索を再実行した際に行集合が増減する現象である。
ウ：障害回復を効率化するための状態記録であり、分離異常ではない。
エ：未コミットの変更を別トランザクションが読んでしまう現象である。
総合解説：分離レベルは許容する並行実行異常とのトレードオフで選ぶ。アプリ側の整合性要件まで含めて評価する。

対象：2026年度 / 基準日 2026-09-01

0045 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：標準

T1が「残高=100」の行を読み、T2がその同じ行を120へ更新してコミットした後、T1が同じ行を再読すると120になった。最も適切な分類はどれか。

ア．ダーティリード
イ．ファントムリード
ウ．ノンリピータブルリード
エ．デッドロック

答え・解説 正答：ウ

ア：T2はコミット済みなので未コミット値の読取りではない。
イ：条件に一致する行集合の追加・削除ではなく、同一行の値が変わっている。
ウ：同じ行を再読した結果が他トランザクションのコミット済み更新によって変化している。
エ：互いにロック待ちして進めない状態ではない。
総合解説：ノンリピータブルリードは既存行の再読値変化、ファントムは検索条件に合致する行集合の変化と整理する。

対象：2026年度 / 基準日 2026-09-01

0046 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：応用

T1が「金額>=100万円」の注文を検索して10件得た。T2が条件に該当する新規注文を追加しコミットした後、T1が同じ検索を行うと11件になった。この現象は何か。

ア．ダーティリード
イ．ロストアップデート
ウ．チェックサムエラー
エ．ファントムリード

答え・解説 正答：エ

ア：追加行はコミット済みであり未コミット読取りではない。
イ：二つの更新が競合し一方を上書きして失う現象ではない。
ウ：通信・記憶の誤り検出であり、トランザクション分離現象ではない。
エ：同一検索条件を再評価したとき、他トランザクションが追加した行により結果集合が変化している。
総合解説：範囲条件の整合性が重要な業務では、Serializable相当の制御やアプリケーションロックなど、必要な保証を選ぶ。

対象：2026年度 / 基準日 2026-09-01

0047 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：標準

2相ロック（2PL）の基本的な考え方として最も適切なものはどれか。
ア．ロックを取得した直後に必ず全ロックを解放する。
イ．読取りも書込みもロックを一切使わない。
ウ．デッドロックを理論上必ず防止する。
エ．ロックを獲得する成長相と、ロックを解放する縮退相を分け、解放後に新たなロックを取得しない。

答え・解説 正答：エ

ア：必要な保護期間を満たせず2PLの定義でもない。
イ：これはロック方式ではない。
ウ：2PLでも複数資源の取得順によってデッドロックは発生し得る。
エ：2PLはロック取得と解放の順序を制約することで競合直列化可能なスケジュールを実現しやすくする。
総合解説：厳密2PLなど派生方式ではコミットまで排他ロックを保持する。直列化可能性と同時実行性、デッドロックの管理を合わせて考える。

対象：2026年度 / 基準日 2026-09-01

0048 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：標準

T1が資源Aをロックして Bを待ち、T2がBをロックして Aを待って処理が進まない。最も直接的な予防策はどれか。
ア．ロック待ちタイムアウトを無限にする。
イ．全トランザクションで取得順をランダムにする。
ウ．ログを無効化する。
エ．複数資源を取得するときのロック取得順序を全処理で統一する。

答え・解説 正答：エ

ア：待ちが永続化し、解決を遅らせる。
イ：取得順の不一致が循環待ちを起こしやすくする。
ウ：障害回復を損なうだけでデッドロックの根本原因を解消しない。
エ：循環待ちを形成しにくくする代表的な予防策である。
総合解説：デッドロック対策には予防、検出して一方をロールバック、タイムアウトなどがある。ロック時間短縮や一貫した取得順も有効である。

対象：2026年度 / 基準日 2026-09-01

0049 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：基礎

MVCC（Multi-Version Concurrency Control）の一般的な利点として適切なものはどれか。
ア．すべての競合をなくし、更新同士も決して衝突しない。
イ．古いバージョンを保持しないためストレージ管理が不要になる。
ウ．分離レベルの概念が不要になる。
エ．複数バージョンを利用して、読取りと書込みの競合を減らし高い同時実行性を得やすい。

答え・解説 正答：エ

ア：更新競合や一意制約競合などは残る。
イ：不要バージョンの回収や可視性管理が必要になる。
ウ：MVCCでもどのスナップショットを見せるかは分離レベルと関係する。
エ：トランザクションごとに適切なスナップショットを見せることで、読取りが更新ロックに阻害されにくくなる。
総合解説：MVCCは同時実行性を高めるが、長時間トランザクションや不要バージョン蓄積、更新競合などの運用課題もある。

対象：2026年度 / 基準日 2026-09-01

0050 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：基礎

WAL（Write-Ahead Logging）の原則として最も適切なものはどれか。
ア．データページを書いた後で、必要ならログを書く。
イ．ログは性能に影響するのでコミット後すぐ削除する。
ウ．データページの変更を永続化する前に、その変更を表すログを先に永続化する。
エ．読取りSQLだけをログし、更新はログしない。

答え・解説 正答：ウ

ア：先行ログにならず、障害時に変更の回復情報を失う恐れがある。
イ：チェックポイントやバックアップ、レプリケーション等の要件に応じて保持が必要である。
ウ：先にログを安全に保存しておけば、障害後にログを用いて必要なREDO/UNDO等の回復処理を行える。
エ：障害回復に必要なのは更新内容の記録である。
総合解説：ログ先行書込みは永続性と回復の基礎である。ログの同期書込み、チェックポイント、バックアップと組み合わせて復旧方式を設計する。

対象：2026年度 / 基準日 2026-09-01

0051 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：応用

複数トランザクションを同時実行しても、結果が何らかの直列実行順序と同等になることを最も強く求める分離レベルはどれか。

ア．Read Committed
イ．Read Uncommitted
ウ．Serializable
エ．Snapshotという名前なら実装に関係なく常にSerializable

答え・解説 正答：ウ

ア：未コミット読取りは防ぐが、ノンリピータブルリード等が発生し得る。
イ：標準上は最も弱い分離レベルである。
ウ：Serializableは並行実行結果を直列実行可能な結果に制約する最も強い標準分離レベルである。
エ：スナップショット方式の保証は実装・分離レベルに依存し、名称だけでSerializableとは限らない。
総合解説：強い分離は整合性を高めるが、競合時の待ちや再試行が増える場合がある。業務不変条件に必要な分離を選ぶ。

対象：2026年度 / 基準日 2026-09-01

0052 データ・データベース > トランザクション・排他・障害回復 ｜ 難易度：標準

障害発生時、ログ上ではT1はCOMMIT済みだが一部データページがディスクへ未反映、T2は未COMMITで一部変更がディスクへ反映済みだった。典型的な回復方針として適切なものはどれか。

ア．T1をUNDOし、T2をREDOする。
イ．T1もT2も必ず何もしない。
ウ．T1の必要な変更をREDOし、T2の未コミット変更をUNDOする。
エ．両方とも必ずREDOする。

答え・解説 正答：ウ

ア：コミット状態と逆の処理でACID要件に反する。
イ：ディスク反映状態がトランザクション状態と不一致なので回復が必要である。
ウ：コミット済み結果は永続性のため再適用し、未コミット結果は原子性のため取り消す。
エ：未コミットT2を残すと原子性が失われる。
総合解説：ログ回復では「コミット済みか」「永続媒体へ反映済みか」を分けて判断する。実際のアルゴリズムはDBMSによるが、REDO/UNDOの目的はACID維持である。

対象：2026年度 / 基準日 2026-09-01

0053 データ・データベース > DB性能・データ活用 ｜ 難易度：標準

索引を追加したときの一般的な効果として最も適切なものはどれか。

ア．索引を増やすほど全SQLが必ず高速になる。
イ．索引は検索性能に影響せず、データ暗号化だけを行う。
ウ．索引を作るとテーブル本体は不要になる。
エ．条件に合う行を高速に探索できる場合がある一方、INSERT/UPDATE/DELETE時の索引保守コストと容量が増える。

答え・解説 正答：エ

ア：更新コストやオプティマイザ選択、低選択性などにより逆効果もある。
イ：索引の主目的は検索・順序処理等の効率化である。
ウ：索引は通常テーブルデータを補助する構造で、本体の代替ではない。
エ：索引は読取り性能を高める代表的手段だが、更新と格納のオーバーヘッドがある。
総合解説：索引は実際のクエリ、選択性、更新頻度、サイズを見て設計する。実行計画と統計情報で利用状況を確認する。

対象：2026年度 / 基準日 2026-09-01

0054 データ・データベース > DB性能・データ活用 ｜ 難易度：標準

多くの検索が WHERE customer_id = ? AND order_date BETWEEN ? AND ? の形で行われる。B-tree複合索引の候補として一般に優先して検討しやすい列順はどれか。

ア．(customer_id, order_date)
イ．(order_date, customer_id)だけが常に最適
ウ．(amount, status)
エ．複合索引はB-treeでは利用できない

答え・解説 正答：ア

ア：先頭列の等値条件で範囲を絞り、その後のorder_date範囲条件を使いやすい典型的な並びである。
イ：利用できる場合もあるが、典型的な等値＋範囲条件では列順による走査範囲が変わる。常に最適とはいえない。
ウ：設問の主要検索条件に含まれず、直接の候補として弱い。
エ：B-treeは複数列索引をサポートし、先頭列側の条件が特に重要になる。
総合解説：複合索引はSQLパターンと列の選択性に基づいて決める。実DBMSの仕様と実行計画を確認し、不要な多列索引を増やさない。

対象：2026年度 / 基準日 2026-09-01

0055 データ・データベース > DB性能・データ活用 ｜ 難易度：応用

1,000万行の表から約80%の行を読み出す集計処理で、ほぼ全列を参照する。索引が存在していてもオプティマイザが全表走査を選ぶ理由として妥当なものはどれか。

ア．大量の行を読む場合、索引経由のランダムアクセスより連続的な全表走査の方が低コストになることがある。
イ．索引が存在するとDBMSは必ず全表走査しかできない。
ウ．全表走査は常に索引走査より遅い。
エ．オプティマイザは統計情報を一切利用しない。

答え・解説 正答：ア

ア：索引は少数行の選択で有利だが、対象割合が大きいと本表アクセス回数が増え、全走査が合理的になる。
イ：索引があれば候補にできるが、コスト比較で選択される。
ウ：対象行割合やI/O特性によっては全表走査が速い。
エ：多くのDBMSは統計情報を使って行数やコストを推定する。
総合解説：実行計画は「索引を使っているか」だけで良否を決めない。処理対象量、I/O形態、キャッシュ、並列化などを含む総コストで評価する。

対象：2026年度 / 基準日 2026-09-01

0056 データ・データベース > DB性能・データ活用 ｜ 難易度：基礎

検索に必要な条件列と取得列が索引内だけで満たせる場合に期待できる利点として最も適切なものはどれか。

ア．索引更新が不要になる。
イ．トランザクション分離レベルが自動的にSerializableになる。
ウ．条件によってはテーブル本体へのアクセスを減らし、I/Oを削減できる。
エ．参照整合性制約が不要になる。

答え・解説 正答：ウ

ア：INSERT/UPDATE/DELETE時の索引保守は必要である。
イ：索引構造と分離レベルは別の概念である。
ウ：索引だけで検索結果を返せる実行計画では、本体ページの読込みを減らせる。
エ：性能索引は論理整合性制約の代替ではない。
総合解説：カバリング索引は読取り性能に有効だが、索引幅が広がるため容量・キャッシュ効率・更新コストとのトレードオフがある。

対象：2026年度 / 基準日 2026-09-01

0057 データ・データベース > DB性能・データ活用 ｜ 難易度：標準

複数の大表を結合した日次集計を多数の利用者が繰り返し参照し、数分程度のデータ遅延は許容される。適切な性能改善策はどれか。

ア．毎回必ず全表を再集計し、キャッシュや事前計算を禁止する。
イ．トランザクションログを削除して集計を高速化する。
ウ．集計結果をマテリアライズドビュー等として事前計算し、更新タイミングを要件に合わせて設計する。
エ．主キーを削除してJOINを高速化する。

答え・解説 正答：ウ

ア：同じ重い処理を繰り返すため負荷削減にならない。
イ：回復性を損ない、集計性能の適切な改善ではない。
ウ：重い集計を毎回計算せず結果を保持することで読取り負荷を下げられるが、鮮度とのトレードオフがある。
エ：整合性を損ない、JOIN性能改善を保証しない。
総合解説：事前計算系の設計では、更新頻度、許容鮮度、再計算時間、障害時の再構築方法を定義する。

対象：2026年度 / 基準日 2026-09-01

0058 データ・データベース > DB性能・データ活用 ｜ 難易度：基礎

時系列ログが年単位で増え続ける表に対し、月単位パーティションを検討している。期待できる効果として適切なものはどれか。

ア．どのSQLでも必ずO(1)で検索できる。
イ．全パーティションを毎回必ず走査するため性能は常に悪化する。
ウ．期間条件による不要パーティションの除外や、古い期間の保守・削除を単位ごとに行いやすくなる。
エ．バックアップや統計情報の管理が不要になる。

答え・解説 正答：ウ

ア：パーティショニングだけで全検索の計算量を一定にできるわけではない。
イ：プルーニング等により対象パーティションを限定できる場合がある。
ウ：アクセス条件と分割キーが合えば走査範囲を減らし、運用も期間単位で扱いやすくなる。
エ：分割後も各パーティションを含む運用管理は必要である。
総合解説：パーティションキーは主要な検索条件とデータライフサイクルに合わせる。細かく分けすぎると管理コストが増える。

対象：2026年度 / 基準日 2026-09-01

0059 データ・データベース > DB性能・データ活用 ｜ 難易度：標準

大量データ投入後から同じSQLが急に遅くなった。実行計画を見ると実際の行数と推定行数が大きく乖離し、不適切なJOIN方式が選択されている。最初に確認すべきものはどれか。

ア．DNSのMXレコード
イ．CPUのエンディアン
ウ．RPOの値
エ．テーブル・列の統計情報が最新か、データ分布を十分表現しているか。

答え・解説 正答：エ

ア：DB内の行数推定とは無関係である。
イ：実行計画の行数推定誤差の直接原因ではない。
ウ：障害復旧のデータ損失許容値でありクエリ最適化とは別である。
エ：コストベース最適化では行数・選択性推定に統計を使うため、陳腐化や分布偏りは誤った計画につながる。
総合解説：DB性能診断では、待機イベント、I/O、ロック、実行計画、実績行数と推定行数を確認する。設定変更より先に証拠を集める。

対象：2026年度 / 基準日 2026-09-01

0060 データ・データベース > DB性能・データ活用 ｜ 難易度：応用

大量の販売明細を、商品・店舗・日付などの軸で対話的に集計するBI用途を設計する。OLTP本番DBへの重い集計負荷を避けたい場合の方針として最も適切なものはどれか。

ア．全BI利用者に本番DBの管理者権限を与える。
イ．分析クエリのたびに本番DBの索引を全部削除する。
ウ．本番OLTPから分析基盤へデータを連携し、ファクトとディメンション等の分析向けモデルを検討する。
エ．分析用途でもトランザクション表1枚だけに全マスタ情報を毎行重複させ、更新統制を設けない。

答え・解説 正答：ウ

ア：セキュリティ上不適切で、本番負荷も抑えられない。
イ：本番性能と運用を悪化させる。
ウ：更新中心のOLTPと集計中心の分析ではアクセス特性が異なるため、用途別にモデル・基盤を分ける価値がある。
エ：無秩序な重複は品質問題を生む。分析向け非正規化はETL等で統制する必要がある。
総合解説：データ活用基盤では、データ鮮度、連携方式、履歴、品質、権限、メタデータを設計する。単なるDBコピーではなく利用目的に合うモデルを選ぶ。

対象：2026年度 / 基準日 2026-09-01

0061 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：基礎

TCPの特徴として最も適切なものはどれか。
ア．常にブロードキャストだけで通信する。
イ．再送や順序制御を一切行わない。
ウ．コネクション型で、順序制御・再送・フロー制御などにより信頼性のあるバイトストリームを提供する。
エ．アプリケーション層のHTTPと同じプロトコルである。

答え・解説 正答：ウ

ア：TCPはIP上のエンドポイント間通信を行い、ブロードキャスト限定ではない。
イ：これはUDPの軽量性に近い説明である。
ウ：TCPは到達保証のための再送や順序制御等を行い、アプリケーションへ信頼性のあるストリームを提供する。
エ：TCPはトランスポート層、HTTPはアプリケーション層のプロトコルである。
総合解説：TCP/UDPの選択は信頼性、遅延、順序、アプリ側再送制御などの要件で決める。

対象：2026年度 / 基準日 2026-09-01

0062 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：標準

TCP接続開始時に3ウェイハンドシェイクを行う主な目的はどれか。
ア．DNS名をIPアドレスへ変換する。
イ．双方が通信可能であることを確認し、初期シーケンス番号など接続状態を同期する。
ウ．HTTPレスポンス本文を圧縮する。
エ．ルータ間で経路を交換する。

答え・解説 正答：イ

ア：名前解決はDNSの役割である。
イ：SYN、SYN/ACK、ACKを交換して接続状態を確立する。
ウ：HTTPのコンテンツ符号化等の論点で、TCP接続確立とは別である。
エ：ルーティングプロトコルの役割である。
総合解説：TCPでは接続確立後もシーケンス番号、ACK、ウィンドウ、再送タイマ等で信頼性とフローを制御する。

対象：2026年度 / 基準日 2026-09-01

0063 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：基礎

IPv6の基本仕様として正しいものはどれか。
ア．48ビットのMACアドレス長をIPv6のアドレス長とみなす。
イ．IPv4と同様に32ビットのアドレス長を使用するとする。
ウ．IPv6ではルータが全パケットを必ず暗号化するとする。
エ．IPv6のIPアドレス長は128ビットである。

答え・解説 正答：エ

ア：48ビットはEthernet MACアドレスで一般的な長さで、IPv6アドレス長ではない。
イ：32ビットはIPv4である。
ウ：IPv6自体が全通信の暗号化を必須にするわけではない。
エ：IPv6はIPv4の32ビットから128ビットへアドレス空間を拡大した。
総合解説：IPv6ではアドレス拡張に加え、基本ヘッダ簡素化や拡張ヘッダ方式などが導入されている。

対象：2026年度 / 基準日 2026-09-01

0064 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：標準

IPv4ネットワーク192.0.2.0/26について、ホスト部のビット数はいくつか。
ア．プレフィックス長/26そのものをホスト部とみなし、26ビットとする。
イ．/24の場合のホスト部と混同し、8ビットとする。
ウ．IPv4アドレス全体の長さをホスト部とみなし、32ビットとする。
エ．IPv4の32ビットからプレフィックス26ビットを除き、ホスト部を6ビットとする。

答え・解説 正答：エ

ア：これはネットワークプレフィックス長である。
イ：/24ならホスト部8ビットだが、/26では6ビットである。
ウ：アドレス全体の長さであり、ホスト部だけではない。
エ：IPv4は32ビットなので、プレフィックス26ビットを除くと32-26=6ビットがホスト部となる。
総合解説：サブネット設計ではプレフィックス長からアドレス数、ネットワーク境界、経路集約を判断する。IPv4とIPv6ではアドレス長が異なる。

対象：2026年度 / 基準日 2026-09-01

0065 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：基礎

DNSの設計上の特徴として最も適切なものはどれか。

ア．階層的な名前空間を分散管理し、名前サーバやリゾルバのキャッシュを利用できる。

イ．世界中の全ドメイン情報を1台の中央サーバだけで管理する。

ウ．ドメイン名はIPアドレスを必ず文字列へ直接埋め込む。

エ．DNSはTCP接続の輻輳制御を行う。

答え・解説 正答：ア

ア：DNSは単一の中央DBに全情報を置くのではなく、ゾーンへ分割して分散管理しキャッシュを活用する。

イ：DNSはスケーラビリティのため分散構造を採用する。

ウ：名前空間はアドレスや経路を名前自体に要求しない設計である。

エ：輻輳制御はTCP等のトランスポート層機能である。

総合解説：DNS障害は多くのサービスへ波及する。権威DNS冗長化、キャッシュ、TTL、DNSSECなどを要件に応じて設計する。

対象：2026年度 / 基準日 2026-09-01

0066 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：標準

HTTPがステートレスなプロトコルであるという説明として適切なものはどれか。

ア．HTTP接続を一度確立すると利用者状態が永久にサーバへ自動保存される。

イ．各要求は基本的に独立して解釈でき、利用者の継続状態が必要ならCookieやトークン、サーバ側セッション等を別途設計する。

ウ．HTTPでは同じURIへ2回アクセスできない。

エ．ステートレスとは暗号化しないという意味である。

答え・解説 正答：イ

ア：HTTPのステートレス性と逆の説明である。

イ：HTTP自身は要求/応答を独立に扱うため、アプリケーション状態は追加の仕組みで管理する。

ウ：同じリソースへ繰り返し要求できる。

エ：状態管理とTLS等の暗号化は別の性質である。

総合解説：水平スケールするWebシステムでは、セッション状態をどこに持つかが可用性・負荷分散へ影響する。

対象：2026年度 / 基準日 2026-09-01

0067 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：標準

HTTPのGETメソッドについて、プロトコルの意味論として最も適切なものはどれか。

ア．リソース表現の取得を目的とする安全なメソッドとして定義され、同じ要求を繰り返しても意図されたサーバ状態変更を生じさせない。

イ．決済を確定するために使うことが標準で推奨される。

ウ．同じGETを2回送ると必ず異なるURLが生成される。

エ．GETはTCPでしか利用できずHTTP/3では存在しない。

答え・解説 正答：ア

ア：GETはsafeであり、読み取り目的の操作として設計される。

イ：状態変更を目的とする操作をGETに割り当てるのは安全メソッドの意味論に反する。

ウ：そのような仕様ではない。

エ：HTTPメソッド意味論はHTTPバージョン間で共有される。

総合解説：API設計ではHTTPメソッドの標準意味論を尊重すると、キャッシュ、再試行、プロキシとの相互運用性が高まる。

対象：2026年度 / 基準日 2026-09-01

0068 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：標準

IPv4のNAT/NAPTを利用する代表的な目的として適切なものはどれか。

ア．プライベートアドレスを利用する複数端末の通信を、少数のグローバルIPv4アドレスへ変換して外部接続する。

イ．DNSSECの署名を生成する。

ウ．IPパケットの到達順序を必ず保証する。

エ．アプリケーションデータを必ず暗号化する。

答え・解説 正答：ア

ア：アドレス・ポート変換を行うことで、内部アドレス空間と外部IPv4アドレスを対応付ける。

イ：NATは名前解決データの署名機構ではない。

ウ：順序保証はNATの機能ではない。

エ：NATはアドレス変換であり暗号化とは別である。

総合解説：NATはIPv4アドレス節約に寄与する一方、エンドツーエンド到達性やプロトコルへの影響を考慮する。

対象：2026年度 / 基準日 2026-09-01

0069 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：応用

インターネット上の異なるAS（自律システム）間で経路情報を交換する代表的なプロトコルはどれか。

ア．BGP
イ．OSPF
ウ．ARP
エ．HTTP

答え・解説 正答：ア

ア：BGPはAS間の経路制御に用いられるパスベクトル型の代表的ルーティングプロトコルである。
イ：OSPFは主に単一AS内部で用いられるリンクステート型IGPである。
ウ：ARPはIPv4アドレスから同一リンク上のMACアドレスを解決するためのプロトコルである。
エ：HTTPはWeb等で用いるアプリケーション層プロトコルである。
総合解説：ネットワーク方式設計では、IGPとEGP、経路集約、冗長経路、収束性、ポリシー制御を区別して考える。

対象：2026年度 / 基準日 2026-09-01

0070 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：基礎

同一リンク上でIPアドレスに対応するリンク層情報を得る仕組みについて、IPv4とIPv6の組合せとして適切なものはどれか。

ア．IPv4もIPv6も必ずARPだけを使う。
イ．IPv4ではDNS、IPv6ではHTTPを使う。
ウ．IPv4ではARP、IPv6ではICMPv6を用いるNeighbor Discoveryが中心となる。
エ．IPv4ではBGP、IPv6ではSQLを使う。

答え・解説 正答：ウ

ア：IPv6はARPを使用しない。
イ：DNS/HTTPは近隣リンクのL2アドレス解決機構ではない。
ウ：IPv6ではARPを使わず、Neighbor Discovery Protocolがアドレス解決や近隣到達性等を担う。
エ：どちらもリンク層アドレス解決とは無関係である。
総合解説：IPv6移行ではアドレス長だけでなく、近隣探索、ルータ広告、自動設定、ICMPv6の重要性など運用差も理解する。

対象：2026年度 / 基準日 2026-09-01

0071 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：応用

遠隔拠点間の大容量ファイル転送で回線帯域には余裕があるがRTTが大きく、単一TCP接続のスループットが伸びない。調査項目として適切なものはどれか。

ア．DBの第3正規形だけを確認する。
イ．TCPウィンドウサイズ、RTT、パケット損失、輻輳制御の状態を確認する。
ウ．画面の文字色を変更する。
エ．DNSのMXレコード数だけを増やす。

答え・解説 正答：イ

ア：ネットワーク転送性能の直接原因ではない。
イ：高帯域・高遅延経路では帯域遅延積に見合うウィンドウが必要で、損失時の輻輳制御もスループットへ影響する。
ウ：転送スループットへ影響しない。
エ：メール配送用レコードであり、単一TCP転送のウィンドウ制御を改善しない。
総合解説：ネットワーク性能は「帯域が余っている＝速い」とは限らない。RTT、損失、ウィンドウ、アプリの並列性をエンドツーエンドで確認する。

対象：2026年度 / 基準日 2026-09-01

0072 ネットワーク・クラウド > ネットワーク技術・プロトコル ｜ 難易度：標準

音声通信と大容量バックアップが同一WAN回線を共有し、輻輳時に音声品質が悪化する。適切な対策はどれか。

ア．トラフィック分類とQoSを設定し、音声の遅延・ジッタ要件に応じて優先制御や帯域保証を行う。
イ．音声パケットだけ意図的に低優先度にする。
ウ．全トラフィックを同一キューで無制限に投入する。
エ．RPOを0分と定義するだけでネットワークを変更しない。

答え・解説 正答：ア

ア：リアルタイム音声は遅延・ジッタに敏感なため、輻輳時のキュー制御で優先度を設ける。
イ：品質悪化をさらに招く。
ウ：輻輳時にバックアップがキューを占有し音声が遅延しやすい。
エ：RPOはデータ復旧目標でありQoS設定ではない。
総合解説：QoSは帯域そのものを増やす技術ではなく、限られた帯域を要件に応じて配分する。容量増強と併用する場合もある。

対象：2026年度 / 基準日 2026-09-01

0073 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：標準

利用者が仮想マシンのOSやミドルウェアを自ら管理し、クラウド事業者が物理基盤や仮想化基盤を提供する形に最も近いサービスモデルはどれか。

ア．SaaS
イ．PaaS
ウ．DNS
エ．IaaS

答え・解説 正答：エ

ア：SaaSでは完成したアプリケーションを利用する形が中心で、OS管理は通常利用者の責任ではない。
イ：PaaSはアプリ実行基盤まで事業者が管理し、利用者はアプリ開発・データに集中する形が典型的である。
ウ：DNSはクラウドのサービスモデル分類ではない。
エ：IaaSでは計算・ストレージ・ネットワーク等の基盤をサービスとして提供し、利用者はOS以上を管理する形が典型的である。
総合解説：クラウド選定ではサービスモデルごとの管理責任境界を明確にする。上位サービスほど運用負担が減る一方、制御範囲は狭くなる。

対象：2026年度 / 基準日 2026-09-01

0074 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：標準

NISTのクラウド定義における「オンデマンド・セルフサービス」の説明として適切なものはどれか。

ア．全ての資源を年単位の固定量でしか契約できない。
イ．利用者が必要に応じて計算資源などを、人手による事業者との個別調整なしに自動的に確保できる。
ウ．利用者が物理データセンターへ毎回出向いてサーバを設置する。
エ．サービス事業者の人手承認が全操作で必須である。

答え・解説 正答：イ

ア：需要に応じたオンデマンド供給と逆である。
イ：クラウドの本質的特性の一つで、必要時にセルフサービスで資源を供給できることを指す。
ウ：クラウドのセルフサービス特性とは異なる。
エ：人手による個別やり取りを最小化する特性に反する。
総合解説：NISTはクラウドを5本質的特性、3サービスモデル、4展開モデルで整理している。用語を方式判断に結び付けて理解する。

対象：2026年度 / 基準日 2026-09-01

0075 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：基礎

負荷増加時にインスタンス数を自動的に増やし、負荷低下時に減らす仕組みが最も直接的に実現するクラウド特性はどれか。

ア．Measured Serviceだけ
イ．単一障害点の固定化
ウ．Rapid Elasticity（迅速な弾力性）
エ．第3正規形

答え・解説 正答：ウ

ア：利用量の計測は重要だが、増減そのものを表す特性ではない。
イ：インスタンス増減はSPOF固定化を目的としない。
ウ：需要に合わせて資源を迅速に拡張・縮小する性質である。
エ：DBの論理設計上の正規形でクラウド特性ではない。
総合解説：オートスケーリングには監視指標、閾値、起動時間、状態管理、DBや外部サービスの上限も含めて設計する。

対象：2026年度 / 基準日 2026-09-01

0076 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：基礎

機密性の高い既存基幹系は専用環境に残しつつ、季節変動が大きいWebフロントをパブリッククラウドで拡張したい。最も近い構成はどれか。

ア．パブリッククラウドだけ
イ．ハイブリッドクラウド
ウ．プライベートクラウドだけ
エ．クラウドを一切利用しない構成

答え・解説 正答：イ

ア：専用環境に残す要件を含まない。
イ：異なるクラウド/専用環境を組み合わせ、データやアプリの連携を行う構成に該当する。
ウ：パブリック側の弾力的利用を含まない。
エ：Webフロントでパブリッククラウドを利用する要件と矛盾する。
総合解説：ハイブリッド構成ではネットワーク、ID連携、監視、データ整合性、障害境界、運用責任をまたいで設計する必要がある。

対象：2026年度 / 基準日 2026-09-01

0077 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：標準

コンテナ運用で「本番稼働中のコンテナへ手作業でパッチを当てず、修正版イメージをビルドして置き換える」方針の主な利点はどれか。

ア．コンテナ内の脆弱性が自動的にゼロになる。
イ．永続データを必ずコンテナの書込み層だけに保存する必要がある。
ウ．実行環境の再現性と変更履歴を保ちやすく、同じイメージを検証後に展開できる。
エ．ホストOSの管理が一切不要になる。

答え・解説 正答：ウ

ア：イメージ更新やスキャン、設定管理は別途必要である。
イ：永続データは外部ボリュームやDBへ分離する設計が一般的である。
ウ：イメージを成果物として扱うと、手作業による構成ドリフトを減らしやすい。
エ：コンテナはホストカーネルを共有するため、ホストOSの保護・更新も重要である。
総合解説：コンテナは可搬性を高めるが、イメージ供給網、レジストリ、ランタイム、ホスト、オーケストレータまで含む運用設計が必要である。

対象：2026年度 / 基準日 2026-09-01

0078 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：応用

Linuxホスト上の一般的なOSコンテナで、Windowsカーネルを必要とするアプリをそのまま実行したい。方式判断として妥当なものはどれか。

ア．コンテナは必ず独自カーネルを含むので問題ない。
イ．コンテナは通常ホストのカーネルを共有するため、そのままでは異なるカーネルを必要とするOS環境を提供できず、VM等を検討する。
ウ．コンテナならCPU命令セットも自動的に任意変換される。
エ．DNSを設定すればOSカーネル差は解消する。

答え・解説 正答：イ

ア：一般的なOSコンテナはカーネルを共有し、VMのような独立ゲストOSではない。
イ：OSコンテナはホストカーネル依存があるため、異なるOSカーネル要件ではVMの方が適する場合がある。
ウ：CPUアーキテクチャ互換性も別途考慮が必要である。
エ：名前解決はカーネル互換性を変えない。
総合解説：コンテナとVMの選択は、OS/カーネル互換性、隔離、起動速度、密度、運用標準化を含めて評価する。

対象：2026年度 / 基準日 2026-09-01

0079 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：応用

コンテナオーケストレータで「Webコンテナを常時5個稼働させる」と宣言し、1個停止したら自動的に新しい1個を起動して5個へ戻す考え方はどれか。

ア．バックアップのフルコピー
イ．DNSの再帰問い合わせ
ウ．宣言したdesired stateと実状態の差を継続的に調整するリコンシリエーション
エ．DBの正規化

答え・解説 正答：ウ

ア：稼働レプリカ数の自動調整とは異なる。
イ：名前解決処理でありワークロード状態管理ではない。
ウ：望ましい状態を定義し、コントローラが実状態を監視して差分を修正する管理方式である。
エ：データモデルの論理設計である。
総合解説：オーケストレーションは配置・再起動・スケーリングを自動化できるが、状態を持つデータ、ヘルスチェック、ローリング更新を別途設計する。

対象：2026年度 / 基準日 2026-09-01

0080 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：標準

クラウド利用量をCPU時間、ストレージ容量、転送量などで計測し、利用者別に可視化・課金配賦する考え方に最も関係する特性はどれか。

ア．Resource Poolingだけ
イ．フェイルセーフ
ウ．2相ロック
エ．Measured Service（計測可能なサービス）

答え・解説 正答：エ

ア：資源プール共有の特性であり、利用量計測そのものを指さない。
イ：障害時に安全側へ移行する設計原則である。
ウ：DB同時実行制御の方式である。
エ：資源利用を計測・監視・報告し、利用量に応じた透明性を提供する特性である。
総合解説：クラウドコスト最適化では利用量計測、タグ/アカウント設計、予算、予約容量、スケール制御を組み合わせる。

対象：2026年度 / 基準日 2026-09-01

0081 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：基礎

イベント発生時だけ短時間処理し、平常時はほぼ要求がない画像メタデータ変換処理を新規開発する。運用負担とアイドルコストを抑えたい。候補として適切なのはどれか。

ア．最大負荷用の物理サーバを常時100台固定稼働することだけを選ぶ。
イ．処理要件を無視してどんな長時間処理も必ずサーバレスにする。
ウ．イベント駆動のサーバレス実行基盤を検討し、実行時間・同時実行・コールドスタート等の制約を確認する。
エ．イベント処理の代わりにDNSのAレコードだけ追加する。

答え・解説 正答：ウ

ア：要求が少ない時間の資源効率が悪く、運用負担も大きい。
イ：実行時間やメモリ、接続、状態管理等の制約があるため適合性確認が必要である。
ウ：断続的イベント処理では自動スケールと従量課金の利点が出やすいが、制約確認が必要である。
エ：DNSは実行ロジックを提供しない。
総合解説：サーバレスはPaaS的な高マネージド方式の一つとして、イベント駆動・急変負荷に向く。ロックイン、観測性、実行制限も評価する。

対象：2026年度 / 基準日 2026-09-01

0082 ネットワーク・クラウド > クラウド・仮想化・コンテナ ｜ 難易度：標準

IaaS上の仮想マシンで業務アプリを運用する。一般的な責任分界の考え方として最も妥当なものはどれか。

ア．クラウド事業者が存在するので利用者側のセキュリティ作業は一切不要である。
イ．利用者がクラウド事業者の物理施設の電源設備を直接保守するのが一般的である。
ウ．責任分界はサービスモデルに関係なく常に同一である。
エ．物理データセンターや基盤は事業者が管理しても、ゲストOSのパッチ、アプリ設定、ID権限、データ保護は利用者側責任が残る。

答え・解説 正答：エ

ア：利用者が管理すべき設定・OS・アプリ・データの責任が残る。
イ：物理基盤は通常事業者側の責任範囲である。
ウ：IaaS/PaaS/SaaSで利用者と事業者の管理範囲は異なる。
エ：IaaSでは利用者がOS以上の多くを管理する。クラウド利用でセキュリティ責任が消えるわけではない。
総合解説：クラウド方式設計では責任分界をコントロール単位で表にし、パッチ、鍵、バックアップ、監視、ログ、インシデント対応の担当を明確にする。

対象：2026年度 / 基準日 2026-09-01

0083 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：標準

注文受付APIが、後続のメール送信サービスの一時停止中でも注文自体は受理したい。最も適切な連携方式はどれか。

ア．注文APIがメール送信完了まで同期的に無期限待機する。
イ．メールサービス停止中は注文DBの主キーを削除する。
ウ．全注文をDNS TXTレコードへ書き込む。
エ．注文確定後にメッセージキューへ通知イベントを登録し、メールサービスが非同期に処理する。

答え・解説 正答：エ

ア：メール障害が注文受付へ直接波及する。
イ：整合性を損ない、連携可用性の解決にならない。
ウ：DNSは業務メッセージキューの代替ではない。
エ：後続サービスの一時停止をキューで吸収し、注文受付とメール送信の時間的結合を弱められる。
総合解説：非同期連携は可用性とスパイク吸収に有効だが、配送保証、順序、重複、失敗時の再処理を設計する必要がある。

対象：2026年度 / 基準日 2026-09-01

0084 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：応用

ネットワークタイムアウトで「決済要求がサーバへ届いたか不明」になることがある。クライアント再試行で二重決済を防ぎたい。適切な設計はどれか。

ア．タイムアウトしたら新しい決済を無条件に毎回実行する。
イ．再試行のたびに金額を変更する。
ウ．クライアントが一意な冪等性キーを送り、サーバが同じキーの処理結果を再利用して重複実行を防ぐ。
エ．HTTPステータスを無視して成功扱いにする。

答え・解説 正答：ウ

ア：最初の要求が成功済みなら二重決済になる。
イ：業務要求を変えてしまい、重複防止にならない。
ウ：同一操作を識別して一度だけ業務効果を発生させれば、通信失敗時に安全に再試行しやすい。
エ：処理結果不明を解決せず整合性リスクが残る。
総合解説：分散システムでは「応答がない＝処理されていない」とは限らない。冪等性、重複排除、照会APIを組み合わせて再試行可能にする。

対象：2026年度 / 基準日 2026-09-01

0085 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：応用

既存の注文リソース /orders/123 の現在表現を取得する操作として、HTTP意味論に最も自然なものはどれか。

ア．GET /orders/123
イ．DELETE /orders/123
ウ．POST /orders/123 を取得専用として使うことだけが標準である
エ．CONNECT /orders/123

答え・解説 正答：ア

ア：GETは対象リソースの表現を取得する安全なメソッドとして定義される。
イ：DELETEは対象リソースとの関連を削除する意味論で、取得操作ではない。
ウ：POSTは汎用処理に使えるが、単純な取得ならGETの意味論が自然である。
エ：CONNECTはトンネル確立に関係するメソッドで、一般的なリソース取得ではない。
総合解説：APIはURI設計だけでなく、HTTPメソッド、状態コード、キャッシュ、冪等性、エラー表現まで整合させると相互運用性が高まる。

対象：2026年度 / 基準日 2026-09-01

0086 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：標準

メッセージング基盤がat-least-once配送を採用している。コンシューマ設計で特に必要な考慮はどれか。

ア．同じメッセージは仕様上絶対に1回しか届かないと仮定する。
イ．メッセージIDを削除して重複判定を不可能にする。
ウ．同一メッセージが複数回配送される可能性を前提に、処理を冪等化または重複排除する。
エ．失敗時もACKを先に返してメッセージを捨てる。

答え・解説 正答：ウ

ア：at-least-onceは重複を許容する配送保証である。
イ：重複排除を難しくする。
ウ：ACK喪失や再配信により少なくとも1回の保証では重複が起こり得る。
エ：少なくとも1回処理したい要件に反する可能性がある。
総合解説：配送保証はブローカだけで完結しない。業務DB更新との整合、ACKタイミング、再試行、DLQ、重複排除をエンドツーエンドで設計する。

対象：2026年度 / 基準日 2026-09-01

0087 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：基礎

「注文確定」イベントを、在庫更新、分析、通知の3システムがそれぞれ独立して受け取りたい。適切なメッセージングモデルはどれか。

ア．単一ワーカーだけが受け取るPoint-to-Pointキューにし、他の2システムへは一切通知しない。
イ．Publish/Subscribeで複数の購読者へイベントを配信する。
ウ．イベントを送信せず、全システムを同じプロセスに統合することだけを選ぶ。
エ．DNSラウンドロビンだけで業務イベントを配信する。

答え・解説 正答：イ

ア：全3システムが独立受信する要件を満たさない。
イ：1つのイベントを複数の独立コンシューマへ届ける用途に適する。
ウ：独立したシステムへ配信する要件に反する。
エ：DNSはサービス宛先解決で、イベント配信モデルではない。
総合解説：Pub/Subは疎結合を高めるが、イベントスキーマ互換性、購読者ごとの再処理、順序、保持期間を管理する必要がある。

対象：2026年度 / 基準日 2026-09-01

0088 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：標準

下流APIが高率でタイムアウトしているとき、呼出し元が毎回長時間待ち続けてスレッド枯渇する連鎖障害を防ぎたい。適切なパターンはどれか。

ア．タイムアウトを無限大にする。
イ．再試行を待ち時間なしで無限回行う。
ウ．Circuit Breakerを用い、失敗率が閾値を超えたら一時的に呼出しを遮断し、回復確認後に再開する。
エ．全ログを削除する。

答え・解説 正答：ウ

ア：待ち資源が長時間占有され、連鎖障害を悪化させる。
イ：下流負荷を増やし障害を増幅し得る。
ウ：失敗中の下流へ無制限に要求を送り続けるのを防ぎ、呼出し元の資源枯渇を抑える。
エ：観測性を失うだけで障害伝播を防げない。
総合解説：分散耐障害設計ではタイムアウト、限定再試行、指数バックオフ、Circuit Breaker、Bulkhead等を組み合わせる。

対象：2026年度 / 基準日 2026-09-01

0089 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：基礎

複数拠点に分散したデータストアでネットワーク分断が発生しても各拠点の書込み受付を継続したい。その間、一時的に拠点間で異なる値を返すことは許容する。重視している性質の組合せとして最も近いものはどれか。

ア．Consistencyだけを優先し、分断中も両拠点が必ず書込み成功する。
イ．Partition toleranceとAvailabilityを優先し、分断中の強いConsistencyを緩める。
ウ．Partition toleranceを不要とし、物理ネットワーク分断は起きないと仮定する。
エ．CAPはCPUキャッシュの置換方式を表す。

答え・解説 正答：イ

ア：強い一貫性を保つなら分断時に一部要求を拒否する必要が生じる。
イ：分断中も各側で応答を返すため、同一時点の単一値を保証する強い一貫性を犠牲にする設計に近い。
ウ：分散拠点では分断を設計上考慮する必要がある。
エ：CAPは分散システムの一貫性・可用性・分断耐性の関係を論じる概念である。
総合解説：CAPは通常時の性能比較ではなく、ネットワーク分断が起きたときの選択を考える枠組みである。業務上どの整合性をどこまで緩められるかを定義する。

対象：2026年度 / 基準日 2026-09-01

0090 ネットワーク・クラウド > 分散処理・API・メッセージング ｜ 難易度：標準

注文・在庫・配送が別サービスで、それぞれ独立DBを持つ。長時間の分散ロックを避けつつ、途中失敗時には業務的に元へ戻したい。適切な考え方はどれか。

ア．全サービスのDBを必ず同じテーブルへ統合しない限り処理できない。
イ．各サービスのローカルトランザクションを順に実行し、失敗時は補償トランザクションを行うSagaを検討する。
ウ．途中失敗時も既に確定した処理を放置し、業務整合性を考えない。
エ．各サービスが互いのDBへ管理者権限で直接更新することだけを採用する。

答え・解説 正答：イ

ア：分散サービスでも調整方式を設計できる。
イ：Sagaは長いグローバルトランザクションの代わりに局所コミットと補償処理を組み合わせる。
ウ：補償や再処理など整合性維持策が必要である。
エ：境界を崩し結合度と障害影響を高める。
総合解説：Sagaでは補償不能処理、再試行、冪等性、イベント順序、監視が重要になる。強いACIDが必要な範囲と業務的整合性でよい範囲を分ける。

対象：2026年度 / 基準日 2026-09-01

0091 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：基礎

顧客の注文データについて、権限のない利用者から内容を見られないようにする対策が主に守る情報セキュリティの特性はどれか。

ア．機密性
イ．完全性
ウ．可用性
エ．真正性

答え・解説 正答：ア

ア：機密性は、認可されていない者への情報開示を防ぐ性質である。
イ：完全性は、情報が不正に変更・破壊されず正確であることに関係する。
ウ：可用性は、必要なときに情報やサービスを利用できることに関係する。
エ：真正性は、主体や情報が主張どおり真正であることの確からしさに関係する。
総合解説：セキュリティ設計では、守るべき特性を機密性・完全性・可用性などに分け、要求に応じた管理策を選ぶ。閲覧権限制御は主に機密性を守る。

対象：2026年度 / 基準日 2026-09-01

0092 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：基礎

インターネット公開サーバに既知の脆弱性が残っており、その脆弱性を悪用する攻撃が観測されている。ここで「脆弱性」に該当するものはどれか。

ア．攻撃者が脆弱性を悪用しようとしていること
イ．サーバに未修正の既知脆弱性が残っていること
ウ．攻撃により顧客情報が漏えいする可能性と影響
エ．漏えい後に実施する顧客への通知

答え・解説 正答：イ

ア：これは脅威事象・攻撃活動に該当する。
イ：脆弱性は、脅威によって悪用され得る弱点であり、未修正のソフトウェア欠陥が該当する。
ウ：これはリスクの評価対象となる可能性・影響である。
エ：これはインシデント対応・事後対応であり、脆弱性そのものではない。
総合解説：リスク評価では、脅威が脆弱性を悪用して望ましくない結果を生む可能性と影響を考える。用語を混同しないことが重要である。

対象：2026年度 / 基準日 2026-09-01

0093 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：標準

ある社内システムの小規模障害について、対策費が想定損失を大きく上回り、事業への影響も限定的である。経営責任者が評価結果を確認した上で、追加対策を行わず定期的に監視することにした。この対応として最も適切なものはどれか。

ア．リスク回避
イ．リスク移転
ウ．リスク受容
エ．リスク低減

答え・解説 正答：ウ

ア：回避は、リスクの原因となる活動自体を中止・変更してリスクをなくす方向の対応である。
イ：移転は、保険や契約などにより損失負担の一部を第三者へ移す対応である。
ウ：評価された残余リスクを権限ある責任者が許容し、必要に応じ監視するのはリスク受容である。
エ：低減は、管理策を追加して発生可能性や影響を下げる対応である。
総合解説：リスク対応は、回避・低減・移転・受容などから、リスク水準、コスト、事業上の便益を踏まえて選ぶ。受容には責任所在と継続監視が必要である。

対象：2026年度 / 基準日 2026-09-01

0094 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：標準

ある障害の年間発生確率を5%、1回当たりの損失額を2,000万円と見積もった。単純化して年間期待損失を「発生確率×損失額」で評価する場合、年間期待損失はいくらか。

ア．5%を0.005と誤って換算し、年間期待損失を10万円とする。
イ．損失額2,000万円を別の割合で割り、年間期待損失を400万円とする。
ウ．発生確率を50%と誤って扱い、年間期待損失を1,000万円とする。
エ．発生確率5%=0.05を掛け、2,000万円×0.05=100万円とする。

答え・解説 正答：エ

ア：発生確率5%を0.005として計算した値であり、5%=0.05である。
イ：発生確率と損失額の積ではなく、別の割合で計算している。
ウ：発生確率50%として扱った値であり、条件と一致しない。
エ：2,000万円×0.05=100万円である。
総合解説：定量評価では、前提を明示した上で発生可能性と影響を金額等に落とすことがある。実務では不確実性や複数シナリオも考慮する。

対象：2026年度 / 基準日 2026-09-01

0095 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：標準

受注登録だけを担当する利用者に、顧客マスタの全件削除権限まで付与されている。最小権限の原則に沿う改善として最も適切なものはどれか。

ア．担当業務に必要な登録・参照権限だけを付与し、不要な削除権限を外す。
イ．全利用者に管理者権限を付与し、操作ログだけを監視する。
ウ．削除権限は残したまま、パスワードの最小文字数だけを増やす。
エ．権限を個人ごとに増やし、担当変更のたびに手作業で付け足す。

答え・解説 正答：ア

ア：最小権限では、業務遂行に必要な最小限の権限のみを付与する。
イ：ログ監視だけでは過大権限による誤操作・悪用の機会を減らせない。
ウ：認証強化は重要だが、不要な権限の削減という問題を解決しない。
エ：必要以上の権限が残りやすく、最小権限と継続的な権限管理に反する。
総合解説：権限設計では、利用者・役割・業務に必要な操作範囲を定義し、不要な権限を与えない。定期的な棚卸しも重要である。

対象：2026年度 / 基準日 2026-09-01

0096 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：標準

Webシステムの防御を一つの対策だけに依存しない方針として、最も多層防御の考え方に沿う構成はどれか。

ア．境界FWだけを高性能化し、アプリケーション側の対策は行わない。
イ．WAF、アプリケーションの入力検証、最小権限のDBアカウント、監視を組み合わせる。
ウ．利用者教育だけを実施し、技術的対策は設けない。
エ．ログを保存せず、障害が起きたときだけ手動確認する。

答え・解説 正答：イ

ア：単一の境界対策への依存であり、多層防御にならない。
イ：異なる層に複数の管理策を置くことで、一つの対策が破られても他の対策で影響を抑えられる。
ウ：人的対策だけに依存しており、層が不足している。
エ：検知・追跡能力が不足し、多層的な防御・監視にならない。
総合解説：多層防御は、境界、アプリケーション、データ、権限、監視など複数層で相互補完する設計である。単一対策への依存を避ける。

対象：2026年度 / 基準日 2026-09-01

0097 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：応用

認可サーバとの通信が切断されたときの業務システムの挙動を検討している。機密データへの不正アクセス防止を最優先する場合、フェイルセキュアの考え方に最も適合するものはどれか。

ア．認可サーバが復旧するまで、全利用者を管理者として扱う。
イ．最後に成功した利用者の権限を全利用者へ適用する。
ウ．認可結果を確認できない要求は原則拒否し、復旧後に再試行させる。
エ．監査ログを停止し、性能低下を避ける。

答え・解説 正答：ウ

ア：障害時に権限を拡大するため、重大な不正アクセスにつながる。
イ：主体ごとの認可が失われ、最小権限にも反する。
ウ：セキュリティ判断ができない状態で許可せず、安全側に倒す設計である。
エ：障害時こそ追跡性が重要であり、セキュリティ上不適切である。
総合解説：フェイルセキュアは、障害時に保護が弱まる方向ではなく、安全側へ遷移する考え方である。ただし可用性要件とのトレードオフは明示する必要がある。

対象：2026年度 / 基準日 2026-09-01

0098 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：標準

高額な支払処理について、申請者が自分の申請をそのまま承認できる。内部不正のリスクを低減する設計として最も適切なものはどれか。

ア．申請者にさらに会計管理者権限を付与する。
イ．承認処理を省略し、後から月次で集計だけ確認する。
ウ．申請画面の配色を変更する。
エ．申請と承認を別の権限・別の担当者に分離する。

答え・解説 正答：エ

ア：権限集中が進み、不正リスクが高まる。
イ：予防的な統制がなくなり、不正な支払を事前に止められない。
ウ：UI改善であり、職務分離による統制にはならない。
エ：重要な処理を複数主体に分けることで、単独で不正を完結しにくくする。
総合解説：職務分離は、申請・承認・実行・監査などの重要機能を分け、権限集中による不正や誤りのリスクを低減する代表的な統制である。

対象：2026年度 / 基準日 2026-09-01

0099 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：基礎

ランサムウェア被害時にも業務データを復旧できる可能性を高めるバックアップ設計として最も適切なものはどれか。

ア．本番環境から直接書換えできない世代バックアップを保管し、復元テストも定期的に行う。
イ．本番サーバと同じ権限で常時マウントした1世代だけを保存する。
ウ．バックアップは取得するが、復元手順は障害発生後に初めて確認する。
エ．バックアップの代わりにアクセスログだけを長期保存する。

答え・解説 正答：ア

ア：本番侵害時にバックアップまで同時に暗号化・削除されるリスクを下げ、実際に復元できることも検証できる。
イ：本番資格情報が侵害されるとバックアップも改変される可能性が高い。
ウ：取得できても復元不能では意味がなく、事前検証が必要である。
エ：ログは調査に有効だが、失われた業務データそのものを復旧できない。
総合解説：バックアップは取得だけでなく、隔離・世代管理・権限分離・復元テストまで含めて設計する。可用性と事業継続の観点が重要である。

対象：2026年度 / 基準日 2026-09-01

0100 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：応用

外部SaaSを中核業務に採用する。サプライチェーンリスクへの対応として最も適切なものはどれか。

ア．SaaSであれば自社にセキュリティ責任は残らないとみなす。
イ．委託先のセキュリティ管理、障害時対応、再委託、データ返却・移行条件を契約・設計の両面で確認する。
ウ．サービス料金だけを比較し、障害時の代替手段は検討しない。
エ．導入後の設定変更を禁止すれば、提供者側のリスクもなくなる。

答え・解説 正答：イ

ア：クラウド利用でも利用者側の設定・データ・契約上の責任は残る。
イ：外部依存では提供者の管理態勢や供給停止・再委託等が自組織のリスクになるため、技術・契約の双方で扱う必要がある。
ウ：供給停止や事業継続のリスクを評価できていない。
エ：提供者の運用、脆弱性、障害、再委託等のリスクは残る。
総合解説：外部サービス利用では、技術仕様だけでなく、提供者の統制、契約、可用性、データ可搬性、終了時対応などを含むサプライチェーンリスク管理が必要である。

対象：2026年度 / 基準日 2026-09-01

0101 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：基礎

同じ評価尺度で、リスクAは発生可能性が高く影響も大きい。リスクBは発生可能性が低く影響も小さい。他の条件が同じなら、通常どちらを優先的に対応すべきか。

ア．リスクB
イ．必ず両方同じ優先度にする
ウ．リスクA
エ．評価せず発見順に対応する

答え・解説 正答：ウ

ア：可能性・影響とも小さいため、通常はAより優先度が低い。
イ：評価結果の差を無視すると、限られた対策資源を適切に配分できない。
ウ：可能性と影響の双方が大きいリスクは、一般に優先度が高い。
エ：リスクベースの優先順位付けにならない。
総合解説：リスク評価では、可能性と影響などを用いてリスク水準を見積もり、対応資源の優先順位を決める。ただし法令・契約要件等で優先度が変わることもある。

対象：2026年度 / 基準日 2026-09-01

0102 セキュリティ・UI/UX > セキュリティ設計・リスク ｜ 難易度：標準

アクセス制御や監視を導入した後も、ゼロにはならず残っているリスクを何と呼ぶか。

ア．固有リスク
イ．移転リスク
ウ．ゼロリスク
エ．残余リスク

答え・解説 正答：エ

ア：一般に管理策を考慮する前のリスクを指す概念であり、設問の状態とは異なる。
イ：一般的な標準用語として設問の状態を表すものではない。
ウ：実務上、管理策を導入してもリスクが完全にゼロになるとは限らない。
エ：管理策を適用した後になお残るリスクを残余リスクという。
総合解説：管理策導入後の残余リスクが組織の許容水準以内かを確認し、必要なら追加対策、移転、回避などを検討する。

対象：2026年度 / 基準日 2026-09-01

0103 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：基礎

大量データを高速に暗号化する用途で、一般に公開鍵暗号より共通鍵暗号が適している主な理由はどれか。

ア．同程度の安全性を狙う処理では、共通鍵暗号の方が一般に計算負荷が小さく高速だから。
イ．共通鍵暗号では鍵管理が不要だから。
ウ．共通鍵暗号では復号鍵を公開してよいから。
エ．公開鍵暗号では暗号化ができず署名しかできないから。

答え・解説 正答：ア

ア：共通鍵暗号は大量データの暗号化に向き、公開鍵暗号は鍵配送や署名などと組み合わせることが多い。
イ：送受信者が秘密鍵を共有する必要があり、鍵管理は重要である。
ウ：共通鍵は秘密に保つ必要があり、公開してはならない。
エ：公開鍵暗号方式は暗号化用途にも利用される。
総合解説：実際のセキュア通信では、公開鍵技術で鍵交換・認証を行い、データ本体は高速な共通鍵暗号で保護するハイブリッド方式が一般的である。

対象：2026年度 / 基準日 2026-09-01

0104 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：標準

送信者が文書のハッシュ値に対して自分の秘密鍵でデジタル署名し、受信者が対応する公開鍵で検証する。主に確認できることはどれか。

ア．文書内容を第三者から秘匿できること。
イ．文書が署名後に改ざんされていないことと、署名鍵の保有者による署名であること。
ウ．受信者だけが文書を復号できること。
エ．通信経路の可用性が保証されること。

答え・解説 正答：イ

ア：署名だけでは内容の機密性は提供しない。秘匿には暗号化が必要である。
イ：署名検証により完全性と署名者の真正性を確認できる。
ウ：これは受信者向け暗号化の性質であり、署名の主目的ではない。
エ：署名はネットワークやサーバの可用性を保証しない。
総合解説：デジタル署名は暗号化と目的が異なる。署名は主に完全性・真正性・否認防止の根拠に関係し、機密性が必要なら別途暗号化する。

対象：2026年度 / 基準日 2026-09-01

0105 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：基礎

パスワードをハッシュ化して保存するとき、利用者ごとに異なるソルトを加える主な目的はどれか。
ア．ハッシュ値から元のパスワードを必ず復号できるようにするため。
イ．パスワードをネットワーク上で平文送信しても安全にするため。
ウ．同じパスワードでも異なるハッシュ値になり、事前計算表などを使った一括解析を困難にするため。
エ．利用者全員のパスワードを同じ値に統一するため。

答え・解説 正答：ウ

ア：ハッシュは本来一方向関数であり、ソルトは復号可能にするためのものではない。
イ：保存方式と通信路保護は別問題であり、通信は適切に保護する必要がある。
ウ：ランダムなソルトは同一パスワードのハッシュ一致を隠し、事前計算攻撃の効率を下げる。
エ：ソルトは利用者ごとに異なる値を使うのが基本である。
総合解説：パスワード保存では、ソルト付きの適切なパスワードハッシュ/KDFを用い、平文や復号可能な形での保存を避ける。

対象：2026年度 / 基準日 2026-09-01

0106 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：標準

次の組合せのうち、異なる種類の認証要素を組み合わせた多要素認証に最も適切なものはどれか。
ア．パスワードと秘密の質問
イ．パスワードとPIN
ウ．指紋と顔認証だけ
エ．パスワードと、利用者が所持するハードウェア認証器

答え・解説 正答：エ

ア：どちらも主に知識要素であり、異なる要素の組合せではない。
イ：どちらも知識要素である。
ウ：どちらも生体に基づく同種要素であり、通常は異なる要素の組合せとは扱わない。
エ：知識要素と所持要素という異なる種類を組み合わせている。
総合解説：MFAは、知識・所持・生体など異なる種類の認証要素を組み合わせる。単に認証ステップを二回行うこととは異なる。

対象：2026年度 / 基準日 2026-09-01

0107 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：標準

数千人の従業員に対し、職務ごとに必要な権限がほぼ共通している。異動時の権限変更を簡素化したい。最も適したアクセス制御方式はどれか。

ア．役割に権限を割り当て、利用者には役割を付与するRBAC
イ．すべての利用者へ同じ管理者権限を付与する方式
ウ．利用者ごとに毎回ゼロから権限を個別設定し、役割は使わない方式
エ．ネットワーク帯域だけでアクセス可否を決める方式

答え・解説 正答：ア

ア：職務に対応する役割単位で権限を管理すると、大規模な人事異動にも対応しやすい。
イ：業務上不要な権限まで与えるため、最小権限に反する。
ウ：可能だが大規模組織では運用負荷が高く、職務単位の共通権限管理に不向きである。
エ：利用者の職務権限管理とは無関係である。
総合解説：RBACは、権限を役割に束ねて利用者へ割り当てる。職務単位で権限が共通する組織で、権限管理・棚卸しを効率化しやすい。

対象：2026年度 / 基準日 2026-09-01

0108 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：応用

文書へのアクセス可否を、利用者の所属・職位、文書の機密区分、アクセス時刻、端末の管理状態など複数条件から動的に判定したい。最も適した考え方はどれか。

ア．利用者IDだけを見て固定権限を直接割り当てる方式
イ．主体・資源・環境などの属性を評価して認可するABAC
ウ．全員に同じ権限を与える共有アカウント方式
エ．IPアドレスだけで利用者本人を認証する方式

答え・解説 正答：イ

ア：複数の環境・資源属性を動的に評価する要件に適しにくい。
イ：ABACは複数属性とポリシーを用いて、状況に応じた細粒度の認可を実現しやすい。
ウ：個人識別や最小権限が損なわれる。
エ：アクセス元情報だけでは本人性や文書属性を十分に評価できない。
総合解説：細粒度な認可では、役割だけでなく主体・資源・環境の属性を組み合わせるABACが有効である。ルール複雑化への管理策も必要になる。

対象：2026年度 / 基準日 2026-09-01

0109 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：標準

利用者の写真サービスに保存された画像の一部を、別の印刷アプリへ限定的に利用させたい。利用者の写真サービス用パスワードを印刷アプリへ渡さずに認可したい。この用途に合う仕組みはどれか。

ア．DNSによる名前解決
イ．NTPによる時刻同期
ウ．OAuth 2.0によるアクセストークンを用いた認可
エ．RAIDによる冗長化

答え・解説 正答：ウ

ア：ホスト名とIPアドレス等の名前解決であり、第三者アプリへの認可を行わない。
イ：時刻同期の仕組みであり、認可委譲とは関係しない。
ウ：OAuth 2.0は、資源所有者の資格情報そのものを第三者アプリへ渡さず、限定されたアクセス権を委譲するための枠組みである。
エ：ストレージ可用性の技術であり、APIアクセス認可ではない。
総合解説：OAuthは認可の枠組みである。『ログイン=OAuth』と単純化せず、本人認証を担う仕組みとの役割分担を理解する。

対象：2026年度 / 基準日 2026-09-01

0110 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：基礎

Webサービスのパスワード保存方法として最も適切なものはどれか。

ア．すべてのパスワードを平文のCSVファイルで保存する。
イ．全利用者のパスワードを一つの共通暗号鍵で可逆暗号化し、アプリに鍵を埋め込む。
ウ．パスワードの先頭3文字だけを保存する。
エ．利用者ごとのソルトを用い、パスワード保存向けの一方向KDF/ハッシュで検証可能な形にする。

答え・解説 正答：エ

ア：漏えい時に即座に認証情報が露出する。
イ：鍵漏えい時に一括復号される。通常のパスワード検証に可逆性は不要である。
ウ：照合精度も安全性も不十分であり、適切な認証情報管理ではない。
エ：認証時に入力値から同じ処理を行って照合し、平文や容易に復号できる保存を避ける。
総合解説：パスワードは、復号して元に戻す必要がないため、専用KDF/ハッシュで保存する。ソルト、計算コスト、漏えい対策を組み合わせる。

対象：2026年度 / 基準日 2026-09-01

0111 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：応用

受け取ったサーバ証明書を信頼できるか判断する際の確認として、最も不適切なものはどれか。
ア．証明書に記載された公開鍵と同じ秘密鍵を、利用者側で入手して保持する。
イ．信頼する認証局までの証明書チェーンを検証する。
ウ．証明書の有効期間や失効状態を確認する。
エ．接続先ホスト名と証明書の対象名が一致するか確認する。

答え・解説 正答：ア

ア：秘密鍵は証明書主体が秘密に保持すべきもので、利用者が入手する必要はない。
イ：証明書が信頼点へ連なることの確認は重要である。
ウ：期限切れ・失効証明書を受け入れないために必要である。
エ：なりすまし防止の観点で重要である。
総合解説：PKIでは、公開鍵と主体の結び付きを証明書で検証する。信頼チェーン、対象名、有効期間、失効などを確認し、秘密鍵は主体側で保護する。

対象：2026年度 / 基準日 2026-09-01

0112 セキュリティ・UI/UX > 暗号・認証・アクセス制御 ｜ 難易度：標準

過去に盗聴された正しい認証メッセージを攻撃者がそのまま再送して認証を通過するリプレイ攻撃を抑止する方法として最も適切なものはどれか。
ア．全利用者で同じ固定パスワードを使う。
イ．チャレンジ用のnonceや時刻など、認証のたびに変化する新鮮性情報を検証する。
ウ．認証ログを削除する。
エ．暗号アルゴリズム名を利用者へ秘密にする。

答え・解説 正答：イ

ア：資格情報の共有により安全性が低下する。
イ：以前のメッセージを再送しても、現在のチャレンジや許容時刻と一致せず拒否できる。
ウ：追跡性を失うだけで、再送自体を防げない。
エ：安全性を方式名の秘匿に依存すべきでなく、リプレイ防止にもならない。
総合解説：認証プロトコルでは、nonce、カウンタ、時刻、ワンタイム値などでメッセージの新鮮性を確認し、過去の有効メッセージの再利用を防ぐ。

対象：2026年度 / 基準日 2026-09-01

0113 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：基礎

入力エラーを赤色だけで示しているWebフォームを改善する。アクセシビリティの観点から最も適切な方法はどれか。

ア．赤色をさらに鮮やかにするだけにする。
イ．正常項目もすべて赤色にする。
ウ．色に加えて、エラーアイコンやテキストメッセージでもエラーを示す。
エ．エラー表示自体をなくす。

答え・解説 正答：ウ

ア：色だけへの依存は解消されない。
イ：区別が失われ、問題を悪化させる。
ウ：WCAGでは色だけを情報伝達の唯一の手段にしないことが求められる。
エ：利用者が修正箇所を把握できず、入力支援を損なう。
総合解説：アクセシビリティでは、色、音、位置など一つの感覚だけに依存せず、複数の手掛かりで同じ意味を伝えることが重要である。

対象：2026年度 / 基準日 2026-09-01

0114 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：基礎

商品画像そのものが商品の種類を伝える重要な情報になっている。スクリーンリーダ利用者にも同等の意味を伝えるための対応として最も適切なものはどれか。

ア．画像ファイル名だけを必ず表示する。
イ．画像をCSSで拡大するだけにする。
ウ．画像をすべて背景画像に変更して説明を削除する。
エ．画像の目的や内容を表す適切なテキスト代替を提供する。

答え・解説 正答：エ

ア：ファイル名が意味を適切に説明するとは限らない。
イ：視認性の一部には寄与し得るが、スクリーンリーダへ意味を伝えない。
ウ：意味のある情報が支援技術に伝わりにくくなる。
エ：非テキストコンテンツには、その目的を満たすテキスト代替を提供するのが基本である。
総合解説：代替テキストは画像の見た目を逐語的に説明するのではなく、その画像がページ上で果たす目的を利用者へ伝えることが重要である。

対象：2026年度 / 基準日 2026-09-01

0115 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：標準

マウスを使えない利用者も操作できるようにするWeb UIの設計として最も適切なものはどれか。

ア．主要な操作をキーボードで実行でき、現在のキーボードフォーカス位置を視覚的に確認できるようにする。
イ．クリック可能な要素をdivだけで作り、キーボード操作は考慮しない。
ウ．フォーカス枠をCSSで完全に消し、見た目を統一する。
エ．全操作をドラッグ操作だけに限定する。

答え・解説 正答：ア

ア：キーボード操作可能性とフォーカスの可視性は、運動障害者などの利用に重要である。
イ：標準的なキーボード操作や支援技術との互換性を損ないやすい。
ウ：現在位置が分からなくなり、キーボード利用を困難にする。
エ：代替操作がなく、操作可能性を損なう。
総合解説：標準HTMLコントロールを適切に使い、Tab移動、Enter/Space操作、フォーカス表示などを確認することがアクセシブルなUIの基本である。

対象：2026年度 / 基準日 2026-09-01

0116 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：標準

会員登録フォームで必須項目が未入力のまま送信された。利用者が修正しやすいエラー表示として最も適切なものはどれか。

ア．画面上部に『エラー』とだけ表示し、該当項目は示さない。
イ．エラー箇所を特定し、何が問題かをテキストで示し、可能なら修正方法も提示する。
ウ．送信ボタンを反応しないようにするだけにする。
エ．未入力項目を自動的に推測値で確定し、利用者へ知らせない。

答え・解説 正答：イ

ア：どこを直すか分からず、修正負荷が高い。
イ：入力エラーの特定と説明は、利用者が自力で修正するために重要である。
ウ：理由が分からず、利用者が次の行動を判断できない。
エ：誤入力や意図しない登録につながる。
総合解説：良いフォーム設計では、ラベル、入力例、エラー箇所、理由、修正方法を明確にし、支援技術にも状態が伝わるようにする。

対象：2026年度 / 基準日 2026-09-01

0117 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：標準

項目の並べ替えをドラッグ操作で提供している。WCAG 2.2を踏まえた改善として最も適切なものはどれか。

ア．ドラッグ速度を速くするだけにする。
イ．マウスカーソルの色だけを変更する。
ウ．ドラッグに加えて、上下移動ボタンなど単一ポインタ操作で実行できる代替手段も用意する。
エ．ドラッグ以外の操作を禁止する。

答え・解説 正答：ウ

ア：ドラッグ自体が困難な利用者への代替にならない。
イ：操作方法の代替にはならない。
ウ：ドラッグ動作が必要な機能には、ドラッグを必要としない代替操作を提供することが求められる場合がある。
エ：アクセシビリティを低下させる。
総合解説：操作方法を一つに固定すると、運動能力や入力機器の違いで利用できない人が生じる。代替操作を設けることで利用可能性が高まる。

対象：2026年度 / 基準日 2026-09-01

0118 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：応用

スマートフォン画面で、削除ボタンと保存ボタンが非常に小さく密接しており、誤タップが多発している。改善として最も適切なものはどれか。

ア．ボタンをさらに小さくし、画面内に多く並べる。
イ．保存と削除を同じ位置に重ね、長押し時間だけで区別する。
ウ．誤タップが起きても確認なしで即時削除する。
エ．操作対象のサイズと間隔を十分に確保し、誤操作しにくい配置にする。

答え・解説 正答：エ

ア：誤操作が増えやすく、アクセシビリティとUXの双方を悪化させる。
イ：操作が複雑になり、誤操作リスクも高まる。
ウ：重大な結果を招く操作でのエラー防止に反する。
エ：小さすぎる操作対象は運動障害者だけでなく一般利用者にも誤操作を招くため、適切なサイズと間隔が重要である。
総合解説：タッチUIでは、ターゲットの大きさ、間隔、重要操作の分離、確認などを組み合わせて誤操作を減らす。WCAG 2.2でもターゲットサイズの基準が追加されている。

対象：2026年度 / 基準日 2026-09-01

0119 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：標準

新しい業務画面を設計する際、実利用者の作業効率と誤操作を改善したい。人間中心設計の考え方に最も沿う進め方はどれか。

ア．利用状況と利用者ニーズを把握し、試作・評価・改善を実利用者のフィードバックを得ながら反復する。
イ．利用者には見せず、開発者だけで完成まで仕様を固定する。
ウ．見た目の色だけを先に決め、業務手順は評価しない。
エ．公開後の問い合わせ件数が増えても画面は変更しない。

答え・解説 正答：ア

ア：利用者・利用状況を起点に反復評価することで、設計者の思い込みだけに頼らず改善できる。
イ：実利用状況とのずれを早期に発見しにくい。
ウ：UXは外観だけでなく、目的達成のしやすさや効率・満足度を含む。
エ：評価結果を改善へ反映する反復性がない。
総合解説：UI/UX設計では、利用者・タスク・利用環境を理解し、プロトタイプやユーザビリティ評価を通して反復的に改善する。

対象：2026年度 / 基準日 2026-09-01

0120 セキュリティ・UI/UX > UI/UX・アクセシビリティ ｜ 難易度：応用

Ajaxでファイルをアップロードした後、画面上に『アップロード完了』と表示するが、スクリーンリーダには変化が伝わらない。改善として最も適切なものはどれか。

ア．完了メッセージの文字色を緑にするだけにする。
イ．状態メッセージを支援技術がプログラム的に認識して通知できるよう、適切なroleやライブ領域等を用いる。
ウ．画面全体を毎回点滅させる。
エ．完了メッセージをDOMから削除してログだけに残す。

答え・解説 正答：イ

ア：色の変更だけではスクリーンリーダに状態変化を伝えられない。
イ：フォーカスを強制移動しなくても、状態変化を支援技術へ伝えられる設計が望ましい。
ウ：アクセシビリティ上の問題を生み、状態通知の適切な方法ではない。
エ：利用者へ状態を通知できなくなる。
総合解説：動的Web UIでは、視覚的な変化だけでなく、支援技術が変化を検出できる意味構造を実装する必要がある。

対象：2026年度 / 基準日 2026-09-01

0121 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：基礎

要求が長期間ほぼ固定され、工程ごとの正式承認と文書化が契約上強く求められるシステム開発で、予測型のウォーターフォールを採用する利点として最も適切なものはどれか。

ア．要求変更が頻繁なほど、変更を無条件で吸収できる。
イ．利用者確認を最後まで一切行う必要がない。
ウ．工程・成果物・承認点を事前に明確化しやすく、契約上の進捗管理と整合させやすい。
エ．テスト工程を省略できる。

答え・解説 正答：ウ

ア：予測型では大きな変更が後工程ほど高コストになりやすい。
イ：要件・設計レビューや受入れ確認は必要である。
ウ：要求の安定性が高い場合、予測型は計画・成果物・承認の管理をしやすい。
エ：開発方式にかかわらず品質確認は必要である。
総合解説：開発方式は、要求の変動性、規制・契約、技術的不確実性、組織能力などに合わせて選ぶ。ウォーターフォールが常に優れるわけではない。

対象：2026年度 / 基準日 2026-09-01

0122 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：標準

アジャイルソフトウェア開発宣言の考え方に最も沿うものはどれか。

ア．計画は一切作らず、その日の思いつきだけで開発する。
イ．顧客との対話を避け、契約書の文言だけで仕様を判断する。
ウ．動くソフトウェアより文書量を最大化することを最優先する。
エ．計画には価値を認めつつ、学習や環境変化に応じて計画を見直し、顧客と協調して価値あるソフトウェアを継続的に届ける。

答え・解説 正答：エ

ア：アジャイルは無計画を意味しない。
イ：顧客との協調を重視する価値観に反する。
ウ：文書にも価値はあるが、動くソフトウェアをより重視する。
エ：アジャイルは計画を否定せず、変化への対応、顧客協調、動くソフトウェアをより重視する。
総合解説：アジャイルでは、短いフィードバックループで学習しながら価値を届ける。『文書不要』『計画不要』という理解は誤りである。

対象：2026年度 / 基準日 2026-09-01

0123 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：基礎

インクリメンタル開発の説明として最も適切なものはどれか。
ア．システムを機能のまとまりごとに段階的に追加し、各段階で利用可能な範囲を拡張していく。
イ．全機能を最後まで統合せず、設計書だけを段階的に増やす。
ウ．要求を一切分割せず、一度だけ全機能をリリースする。
エ．過去のリリースを毎回捨て、全てをゼロから作り直す。

答え・解説 正答：ア

ア：完成機能を増分として順次追加するのがインクリメンタル開発の基本である。
イ：利用可能な製品・機能の増分を届ける考え方ではない。
ウ：段階的な機能追加ではない。
エ：増分を積み上げる考え方に反する。
総合解説：反復型と増分型は組み合わせて使われることが多い。反復は同じ対象を改善し、増分は利用可能な機能範囲を追加する観点で整理できる。

対象：2026年度 / 基準日 2026-09-01

0124 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：標準

利用者自身も業務画面の要件を具体化できていない。要件の誤解を早期に減らす方法として最も適切なものはどれか。
ア．本番完成まで利用者に一切見せない。
イ．操作可能なプロトタイプを早期に作り、利用者の評価を受けて要件を具体化する。
ウ．画面要件を開発者だけで推測し、変更を禁止する。
エ．プロトタイプを本番品質だと誤認させ、そのまま無検証で本番投入する。

答え・解説 正答：イ

ア：要件誤解が後工程まで残る可能性が高い。
イ：抽象的な要求を具体物で確認することで、認識差や使い勝手の問題を早期に発見できる。
ウ：利用者ニーズを確認できない。
エ：試作と本番成果物の目的・品質を混同している。
総合解説：プロトタイピングは、要求獲得・UI評価・技術検証に有効である。一方、試作品をそのまま本番化する場合は品質・保守性を別途確認する必要がある。

対象：2026年度 / 基準日 2026-09-01

0125 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：標準

V字モデルの考え方として最も適切なものはどれか。
ア．テストは実装完了後に初めて考えればよいとする。
イ．要求定義と受入れテストは無関係とする。
ウ．要求・設計の各段階に対応する検証・妥当性確認のテスト段階を意識し、上流成果物からテスト観点を早期に準備する。
エ．単体テストで全業務要件の妥当性を確認する。

答え・解説 正答：ウ

ア：上流成果物とテストの対応を早期に考える意義に反する。
イ：要求が受入れ判断の基礎になる。
ウ：V字モデルは、左側の仕様化・設計と右側のテスト段階の対応関係を明確にする。
エ：単体テストだけではシステム全体の業務要件を十分確認できない。
総合解説：V字モデルでは、要求と受入れ、システム設計とシステムテスト、詳細設計と単体/結合テストなどの対応を意識し、早期にテスト可能性を高める。

対象：2026年度 / 基準日 2026-09-01

0126 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：応用

新規性の高い技術を使う大規模システムで、性能実現性など重大な技術リスクを早期に評価しながら段階的に開発したい。最も適した考え方はどれか。
ア．技術リスクを最後の総合テストまで評価しない。
イ．要求・設計・実装を一度だけ実施し、見直しを禁止する。
ウ．リスクの大小に関係なく、最も容易な機能だけを先に作る。
エ．各反復でリスクを識別・評価し、プロトタイプ等で高リスク事項を先に検証するスパイラル型

答え・解説 正答：エ

ア：重大な実現性問題の発見が遅れ、損失が大きくなる。
イ：不確実性が高い案件でのリスク駆動反復に合わない。
ウ：高リスク項目の早期検証という目的を満たさない。
エ：スパイラル型はリスク分析を反復ごとの重要活動とし、高リスク事項の早期低減に向く。
総合解説：不確実性が高い案件では、技術検証、プロトタイプ、段階的コミットメントなどで重大リスクを早期に減らすことが有効である。

対象：2026年度 / 基準日 2026-09-01

0127 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：標準

DevOpsの実践として最も適切なものはどれか。
ア．開発と運用が共通目標を持ち、ビルド・テスト・デプロイを自動化し、運用フィードバックを開発へ継続的に戻す。
イ．開発と運用の責任境界を完全に断絶し、情報共有を禁止する。
ウ．本番リリースのたびに手作業を増やし、再現性を下げる。
エ．監視データを開発チームへ共有しない。

答え・解説 正答：ア

ア：組織間の協働と自動化、継続的フィードバックがDevOpsの中心的な狙いである。
イ：協働を阻害し、DevOpsの狙いに反する。
ウ：自動化・再現性・短いフィードバックを損なう。
エ：運用からの学習が開発へ戻らない。
総合解説：DevOpsは単なるツール導入ではなく、開発・運用・品質・セキュリティの協働、継続的な自動化とフィードバックを通じて価値提供を改善する考え方である。

対象：2026年度 / 基準日 2026-09-01

0128 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：基礎

リファクタリングの説明として最も適切なものはどれか。
ア．新しい業務機能を追加することだけを指す。
イ．外部から見える振る舞いを原則変えずに、内部構造を改善して保守性などを高める。
ウ．障害を隠すためにテストを削除する。
エ．設計を一切変えず、変数名も含めて永久に固定する。

答え・解説 正答：イ

ア：機能追加とは目的が異なる。
イ：リファクタリングは既存動作を保ちながらコード内部の設計を改善する活動である。
ウ：品質を低下させる行為であり、リファクタリングではない。
エ：内部構造改善という目的に反する。
総合解説：安全なリファクタリングには、自動テストや小さな変更単位を活用して振る舞いが変わっていないことを継続確認するのが有効である。

対象：2026年度 / 基準日 2026-09-01

0129 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：応用

短納期を優先して重複コードや暫定実装を多数残した結果、後の機能追加で修正範囲が広がり、テスト工数も増えている。この状態の説明として最も適切なものはどれか。

ア．機能要求が一切存在しない状態である。
イ．テストを増やせば内部構造の問題は必ず自動的に消える。
ウ．短期的な開発速度と引き換えに将来の変更コストを増やす技術的負債が蓄積している。
エ．技術的負債は必ずゼロにしなければ開発してはいけない。

答え・解説 正答：ウ

ア：機能要求の有無ではなく、内部品質と将来コストの問題である。
イ：テストは安全な変更に役立つが、設計負債そのものを自動除去しない。
ウ：暫定実装を放置すると、将来の保守・変更で利息のように追加コストが発生する。
エ：事業判断で意図的に負う場合もあるが、把握・返済計画・リスク管理が必要である。
総合解説：技術的負債は必ず悪ではないが、可視化せず放置すると変更速度・品質・障害リスクを悪化させる。返済優先度を事業価値と合わせて管理する。

対象：2026年度 / 基準日 2026-09-01

0130 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：標準

新システムの設計段階で運用・保守性を高める取組として最も適切なものはどれか。

ア．運用設計は本番稼働後に初めて考える。
イ．保守担当者には設計情報を共有しない。
ウ．廃棄時のデータ消去や移行は考慮しない。
エ．監視、バックアップ、ログ、障害時手順、更新方法、廃棄・移行までを設計要件として早期に検討する。

答え・解説 正答：エ

ア：監視・復旧・権限などが設計に織り込まれず、手戻りが増える。
イ：保守性・障害対応力を低下させる。
ウ：ライフサイクル終端のリスクが残る。
エ：ライフサイクルは開発だけでなく運用・保守・廃棄まで含み、後付けでは高コストになりやすい。
総合解説：システムアーキテクトは、構想・開発・導入だけでなく、運用、保守、変更、終了までを見通して設計判断する必要がある。

対象：2026年度 / 基準日 2026-09-01

0131 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：基礎

アジャイル開発で要求変更が発生したときの考え方として最も適切なものはどれか。
ア．変更の価値・影響・優先順位を評価し、バックログ等へ反映して次の反復で扱う。
イ．一度決めた要求は状況が変わっても絶対に変更しない。
ウ．変更要求は全て即時に作業中へ割り込み、計画や優先順位を無視する。
エ．変更内容を記録せず、口頭だけで実装する。

答え・解説 正答：ア

ア：変化を歓迎することは無制限に受け入れることではなく、価値と影響を見て優先順位を調整することである。
イ：変化への対応という価値観に反する。
ウ：チームの集中や予測可能性を損なう。
エ：認識差や追跡不能を生みやすい。
総合解説：アジャイルでも変更管理は必要である。短い計画サイクルで価値とリスクを見直し、関係者で優先順位を調整する。

対象：2026年度 / 基準日 2026-09-01

0132 開発・マネジメント基礎 > 開発手法・ライフサイクル ｜ 難易度：標準

規制対応部分は仕様・承認手続が厳格だが、利用者向け画面は要求変更が多い案件で、最も合理的な開発方針はどれか。
ア．全てを同一方式に固定し、領域差は考慮しない。
イ．規制対応部分は予測型で管理し、UI部分は反復・適応型で進めるなど、領域特性に応じて方式を組み合わせる。
ウ．規制要件も文書化せず、全て口頭だけで進める。
エ．UI要件も初日に完全固定し、利用者評価を禁止する。

答え・解説 正答：イ

ア：案件特性に合わない部分で過剰な制約や変更コストが生じる。
イ：プロジェクト全体を一方式に固定せず、成果物や不確実性に応じて適切なアプローチを組み合わせられる。
ウ：正式承認・追跡性が必要な領域に不適切である。
エ：変化が多い領域の学習機会を失う。
総合解説：ISO 21502も予測型、反復型、適応型、ハイブリッド等をプロジェクト特性に応じて選べる考え方を示している。重要なのは方式名ではなく適合性である。

対象：2026年度 / 基準日 2026-09-01

0133 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：基礎

複数のモジュールを接続したとき、インタフェースのデータ形式や呼出し順序の不整合を主に検出するテストはどれか。

ア．単体テスト
イ．受入れテスト
ウ．結合テスト
エ．運用監視

答え・解説 正答：ウ

ア：個々のモジュールや関数の内部動作を主に確認する。
イ：利用者・発注者の要求を満たすかを確認する最終段階のテストである。
ウ：モジュール間・コンポーネント間のインタフェースや相互作用を確認するのが結合テストの主目的である。
エ：本番運用中の状態把握であり、テストレベルではない。
総合解説：テストレベルごとに目的が異なる。単体、結合、システム、受入れを、どの欠陥をどの段階で見つけたいかで整理する。

対象：2026年度 / 基準日 2026-09-01

0134 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：標準

年齢入力が18〜65歳なら有効、それ以外は無効という仕様がある。同値分割法の観点で、最小限の代表クラスとして最も適切なものはどれか。

ア．18と65の2値だけ
イ．0〜100を1歳刻みの101クラス
ウ．18〜65の有効クラスだけ
エ．18未満、18〜65、66以上の3クラス

答え・解説 正答：エ

ア：境界値としては重要だが、同値クラス全体の分割を表していない。
イ：同じ動作をする入力まで細分化しており、同値分割の効率化目的に反する。
ウ：無効側のクラスを検証できない。
エ：入力領域を、同じ結果になる無効・有効・無効の三つの同値クラスに分けて代表値を選べる。
総合解説：同値分割法は、同じ処理結果が期待される入力領域をクラス化し、各クラスから代表値を選んでテスト数を抑えつつ網羅性を確保する。

対象：2026年度 / 基準日 2026-09-01

0135 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：基礎

入力可能な数量が1〜100個である。境界値分析で特に確認すべき値の組合せとして最も適切なものはどれか。

ア．0、1、2、99、100、101
イ．25、50、75だけ
ウ．1と100だけ
エ．10、20、30、40だけ

答え・解説 正答：ア

ア：有効範囲の直前・境界・直後を確認する典型的な境界値の組合せである。
イ：範囲中央の代表値であり、境界の不具合を狙っていない。
ウ：境界そのものは確認できるが、境界の直外側を確認できない。
エ：境界条件を重点的に確認する組合せではない。
総合解説：境界付近は比較演算子や配列上限などの欠陥が生じやすい。境界値分析では、境界の直前・境界・直後を重点的に確認する。

対象：2026年度 / 基準日 2026-09-01

0136 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：標準

障害修正で一つの計算ロジックを変更した。修正箇所以外の既存機能が壊れていないか確認するために実施するテストはどれか。

ア．負荷テストだけ
イ．回帰テスト
ウ．受入れ判定の省略
エ．データ移行だけ

答え・解説 正答：イ

ア：性能特性は確認できるが、既存機能全般の退行を確認する目的とは異なる。
イ：変更によって既存機能に予期しない副作用が生じていないかを確認する。
ウ：テストではなく、品質確認を弱める行為である。
エ：変更影響による既存機能の破壊を検証するテストではない。
総合解説：変更が局所的でも依存関係を通じて別機能へ影響することがある。自動化された回帰テスト群は変更頻度が高い開発で特に有効である。

対象：2026年度 / 基準日 2026-09-01

0137 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：標準

プログラムを実行せず、設計書やソースコードをレビューして欠陥を見つける活動はどれか。
ア．負荷テスト
イ．回帰テスト
ウ．静的テスト
エ．フェイルオーバテスト

答え・解説 正答：ウ

ア：実際にシステムを動作させて性能を確認する動的テストである。
イ：変更後に既存機能が壊れていないか実行して確認する動的テストである。
ウ：成果物を実行せずにレビューや静的解析で欠陥を検出する活動が静的テストに含まれる。
エ：障害切替を動作させて確認する動的な検証である。
総合解説：欠陥を早期に見つけるには、レビュー・静的解析と、実行による動的テストを組み合わせる。上流欠陥の早期除去は手戻り削減にもつながる。

対象：2026年度 / 基準日 2026-09-01

0138 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：応用

あるモジュールの欠陥密度が他より高いことが分かった。品質判断として最も適切なものはどれか。
ア．欠陥密度が高いので、そのモジュールを無条件で削除する。
イ．欠陥密度が高くても、測定結果は一切見ない。
ウ．欠陥密度が低ければテストは今後不要と判断する。
エ．欠陥密度だけで結論を出さず、変更量、複雑度、テスト量、欠陥重大度など他の情報と合わせて重点レビュー対象を判断する。

答え・解説 正答：エ

ア：事業上必要な機能か、原因が何かを評価せず結論を飛躍させている。
イ：品質リスクのシグナルを無視している。
ウ：欠陥未検出と欠陥不存在は同じではなく、テスト十分性も別途評価が必要である。
エ：メトリクスは状況を示す手掛かりであり、単独値だけで品質の良否を断定すると誤る可能性がある。
総合解説：品質メトリクスは、トレンド、基準値、モジュール特性、テストカバレッジ等と組み合わせて解釈し、改善行動へつなげる。

対象：2026年度 / 基準日 2026-09-01

0139 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：標準

正式レビューで承認したリリース対象の設計書・ソース・設定ファイルを基準として固定し、その後の変更を管理手続の対象にした。この基準状態を表す用語はどれか。

ア．ベースライン
イ．スパイク
ウ．ペルソナ
エ．ハッシュ衝突

答え・解説 正答：ア

ア：正式に合意された構成の基準状態を定め、以後の変更を統制するために用いる。
イ：アジャイルで不確実性を調査する短期の検証活動を指すことがある。
ウ：UX設計で想定利用者像を表す概念である。
エ：異なる入力が同じハッシュ値になる現象であり、構成管理の基準状態ではない。
総合解説：構成管理では、識別、版管理、変更管理、状態記録、監査などを通じて、何が正式版かを追跡できるようにする。

対象：2026年度 / 基準日 2026-09-01

0140 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：基礎

複数チームが同じソフトウェアを並行開発する際、版管理システムを使う主な利点として最も適切なものはどれか。

ア．ソースコードの品質を自動的に100%保証する。
イ．変更履歴を追跡し、分岐した変更を比較・統合し、必要に応じ過去版へ戻せる。
ウ．全開発者が常に同じファイルを同時編集し、履歴を残さない。
エ．リリース後の変更履歴を削除する。

答え・解説 正答：イ

ア：版管理は履歴・統合を支援するが、品質保証にはレビューやテスト等が必要である。
イ：版管理は誰が何をいつ変更したかを記録し、並行作業やリリース管理を支援する。
ウ：版管理の利点を失う。
エ：追跡性・監査性を損なう。
総合解説：構成管理と版管理は、再現可能なビルドや障害調査、リリース比較にも重要である。履歴だけでなくタグ・ブランチ・変更承認と組み合わせる。

対象：2026年度 / 基準日 2026-09-01

0141 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：応用

本番リリース直前に、開発者が承認済み設定ファイルを独断で変更した。構成管理上、最も問題となる点はどれか。

ア．設定ファイルはソースコードではないので構成管理対象にできないこと。
イ．承認済み成果物は永久に変更してはいけないこと。
ウ．ベースライン化された構成に対する変更の評価・承認・記録を経ておらず、再現性と追跡性が損なわれること。
エ．変更内容が小さいほど記録してはいけないこと。

答え・解説 正答：ウ

ア：設定もシステム動作を左右する重要な構成項目であり、管理対象になり得る。
イ：変更自体は禁止ではなく、統制された手続で行うことが重要である。
ウ：正式版への変更は影響評価と承認を経て記録し、何が本番に入ったか追跡できる必要がある。
エ：小さな変更でも本番影響や再現性のため記録が必要な場合がある。
総合解説：構成管理では、ソース、ライブラリ、設定、インフラ定義、文書など本番を再現するのに必要な構成項目を特定し、変更を統制する。

対象：2026年度 / 基準日 2026-09-01

0142 開発・マネジメント基礎 > 品質・テスト・構成管理 ｜ 難易度：標準

要求仕様の各項目に対して対応する設計要素とテストケースを関連付けて管理する主な目的はどれか。

ア．要求変更を技術的に不可能にする。
イ．テストケース数を必ずゼロにする。
ウ．設計書を読めないよう暗号化することだけを目的とする。
エ．要求の実装漏れやテスト漏れを検出し、変更時の影響範囲を追跡しやすくする。

答え・解説 正答：エ

ア：トレーサビリティは変更を禁止するのではなく、影響把握を支援する。
イ：品質確認と逆である。
ウ：機密性対策とは別の目的である。
エ：双方向のトレーサビリティは、要求がどこで実装・検証されているかを確認するのに役立つ。
総合解説：要求→設計→実装→テストの関連を追跡できると、カバレッジ確認や変更影響分析、受入れ根拠の説明がしやすくなる。

対象：2026年度 / 基準日 2026-09-01

0143 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：基礎

プロジェクトのネットワーク図でクリティカルパス上の作業が、他の条件を変えずに3日遅延した。一般に最も直接的に起こり得る影響はどれか。

ア．プロジェクト全体の完了が3日遅れる。
イ．必ず3日前倒しになる。
ウ．全体納期には絶対に影響しない。
エ．品質が必ず向上する。

答え・解説 正答：ア

ア：クリティカルパス上の作業は通常、全体日程の余裕がなく、遅延が全体納期に直結する。
イ：遅延しているため前倒しにはならない。
ウ：クリティカルパスには余裕がないため、通常は影響する。
エ：日程遅延と品質向上に必然的な関係はない。
総合解説：クリティカルパスはプロジェクト所要期間を決める経路である。遅延対策では、クリティカル作業の短縮や順序・資源の見直しを検討する。

対象：2026年度 / 基準日 2026-09-01

0144 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：標準

EVMで、出来高EVが800万円、実コストACが1,000万円である。CPI=EV/ACとすると、CPIと状況の組合せとして正しいものはどれか。

ア．CPI=1.25で、コスト効率が良い。
イ．CPI=0.8で、投入コストに対して出来高が少なくコスト超過傾向である。
ウ．CPI=0.8で、必ず納期が前倒しである。
エ．CPI=1.0で、計画どおりである。

答え・解説 正答：イ

ア：EV/ACではなく逆数を計算している。
イ：800/1000=0.8。CPIが1未満ならコスト効率が計画より悪いことを示す。
ウ：CPIはコスト効率であり、納期はSPI等別指標で見る。
エ：EVとACが等しくないため1.0ではない。
総合解説：EVMでは、CPIはコスト効率、SPIはスケジュール効率を表す。単一指標だけでプロジェクト全体を判断せず、原因と見通しも分析する。

対象：2026年度 / 基準日 2026-09-01

0145 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：標準

プロジェクトで識別したリスクを継続管理する方法として最も適切なものはどれか。
ア．リスクは発見した人の記憶だけに残し、共有しない。
イ．発生済みの問題だけを記録し、将来リスクは扱わない。
ウ．リスクの内容、原因、影響、発生可能性、対応策、責任者、状態などを登録し、定期的に見直す。
エ．対応策を決めたら、その後の状況変化は確認しない。

答え・解説 正答：ウ

ア：属人化し、対応漏れが起きやすい。
イ：リスク管理は未発生の不確実事象も対象とする。
ウ：リスクを可視化し、責任者と対応状況を追跡することで継続的な管理ができる。
エ：発生可能性や影響は変化するため、継続監視が必要である。
総合解説：リスク管理は一度きりの作業ではない。識別、分析、対応計画、実施、監視をプロジェクト全体で繰り返す。

対象：2026年度 / 基準日 2026-09-01

0146 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：標準

ITサービスの可用性や応答時間、サポート時間など、提供者と顧客の間で合意したサービス目標を明文化するものはどれか。
ア．WBS
イ．ER図
ウ．公開鍵証明書
エ．SLA

答え・解説 正答：エ

ア：プロジェクト作業を階層的に分解するための構造である。
イ：データ構造・エンティティ間関係を表すモデルである。
ウ：公開鍵と主体の結び付きを証明するもので、サービスレベル合意ではない。
エ：SLAはサービスレベルに関する合意を明確にし、測定・報告・改善の基準になる。
総合解説：SLAでは、測定可能な指標、測定方法、報告、例外条件、責任分担などを明確にすることが重要である。

対象：2026年度 / 基準日 2026-09-01

0147 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：応用

同じ障害が毎週繰り返し発生し、そのたびにサービス再起動で復旧している。サービス復旧とは別に、再発防止のため根本原因を調査・除去する活動として最も適切なものはどれか。

ア．問題管理
イ．インシデント管理だけ
ウ．キャパシティ管理だけ
エ．アクセス制御だけ

答え・解説 正答：ア

ア：反復するインシデントの根本原因を特定し、恒久対策・既知の誤り管理などにつなげる活動である。
イ：インシデント管理は早期のサービス復旧を重視する。根本原因の恒久除去は問題管理の主目的である。
ウ：容量不足が原因なら関係するが、一般に根本原因分析そのものを指す用語ではない。
エ：原因が権限問題とは限らず、再発分析のプロセスを表していない。
総合解説：サービス管理では、インシデントの迅速な復旧と、問題の根本原因分析・再発防止を分けて考える。両者は連携するが目的が異なる。

対象：2026年度 / 基準日 2026-09-01

0148 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：標準

重要サービスの災害対策で、復旧設計を具体化するために最も重要な組合せはどれか。

ア．画面の背景色とロゴサイズだけを定める。
イ．どの時点までのデータ損失を許容するかと、どれだけの時間でサービスを復旧するかを定める。
ウ．開発者の人数だけを増やし、復旧目標は決めない。
エ．障害発生後に初めて復旧時間を考える。

答え・解説 正答：イ

ア：災害復旧の要件にはならない。
イ：RPOとRTOに相当する目標を明確にすると、バックアップ頻度や冗長化方式の設計根拠になる。
ウ：必要な復旧水準が不明では適切な対策を選べない。
エ：事前の継続計画にならない。
総合解説：継続性設計では、事業影響を踏まえ、RTO・RPO、代替拠点、バックアップ、連絡体制、復旧手順、訓練等を整合させる。

対象：2026年度 / 基準日 2026-09-01

0149 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：基礎

個人情報を扱う業務システムの設計・運用として最も適切なものはどれか。
ア．収集できる個人情報は目的に関係なく全て永久保存する。
イ．個人情報であればアクセス権限管理は不要である。
ウ．利用目的や業務上の必要性を踏まえて取り扱う情報を定め、アクセス制御や漏えい防止など必要な安全管理措置を講じる。
エ．本番データを無制限に開発者個人端末へコピーする。

答え・解説 正答：ウ

ア：必要性を超えた収集・保有はリスクを増やす。
イ：漏えい・不正利用防止のため適切な安全管理が必要である。
ウ：個人情報は目的・必要性に沿って適切に取り扱い、安全管理を行う必要がある。
エ：アクセス統制や持出し管理がなく、漏えいリスクが高い。
総合解説：個人情報保護では法令だけでなく、システム設計上も必要最小限のデータ、権限、ログ、保管期間、委託管理などを具体化することが重要である。

対象：2026年度 / 基準日 2026-09-01

0150 開発・マネジメント基礎 > プロジェクト・サービス・法務・標準 ｜ 難易度：応用

他人の利用者IDとパスワードを本人やアクセス管理者の承諾なく使い、インターネット経由でアクセス制御されたシステムにログインして利用した。法的な評価として最も適切なものはどれか。
ア．本人のパスワードが正しかったので、常に適法である。
イ．社外からのアクセスだけが対象で、社内ネットワーク経由なら必ず適法である。
ウ．不正アクセス禁止法はパスワード取得だけを禁止し、実際の不正ログインは対象外である。
エ．不正アクセス行為に該当し得る。

答え・解説 正答：エ

ア：正しい識別符号でも、承諾なく他人のものを使う場合は問題となる。
イ：法の要件は単純な社内外区分ではなく、電気通信回線とアクセス制御等に基づく。
ウ：同法は不正アクセス行為そのものを禁止している。
エ：他人の識別符号を無断で入力し、アクセス制御で制限された利用を可能にする行為は、不正アクセス行為の典型に該当する。
総合解説：不正アクセス禁止法は、不正アクセス行為に加え、不正アクセス目的の識別符号取得・提供・保管や不正な入力要求等も規定する。問題作成時は適用時点の現行法を確認する。

対象：2026年度 / 基準日 2026-09-01