

正 誤 表
-------

下記の部分に誤りがありましたので訂正させていただきます。  
ご迷惑をおかけし大変申し訳ございません。

## 2022 高度午前 I・応用情報 午前試験対策書 第 1 版 第 1 刷

No	訂正箇所	誤	正
1	P.16 図表 3 午前の 出題範囲	<p>大分類：4 開発技術 / 中分類：12 システム開発技術</p> <p>システム要件定義, システム方式設計, ソフトウェア要件定義, ソフトウェア方式設計・ソフトウェア詳細設計, ソフトウェア構築, ソフトウェア結合・ソフトウェア適格性確認テスト, システム結合・システム適格性確認テスト, 導入, 受入れ支援, 保守・廃棄</p>	<p>大分類：4 開発技術 / 中分類：12 システム開発技術</p> <p>システム要件定義・ソフトウェア要件定義, 設計, 実装・構築, 統合・テスト, 導入・受入れ支援, 保守・廃棄</p> <p>※2021年10月26日, IPAのシラバス改訂に伴う変更</p>
2	P.475 (5)タイトルと 本文上から2 行目	<p>(5) モジュール設計技法とモジュールの独立性</p> <p>モジュールは、コンパイルして実行することができる最小単位のプログラムです。プログラム設計技法は、プログラムをモジュールに分割して、そのプログラムのモジュール構造を決定するための技法です。そして、その結果を、モジュール強度とモジュール結合度という尺度で評価します。分割したモジュールは、モジュール強度が強く、モジュール結合度が弱いものが独立性の高い良いモジュールとされています。この尺度の根本には、保守性という観点があります。仕様変更などでプログラムの修正が発生した場合に、できるだけ修正対象のモジュールを限定し、影響範囲を狭めたいという考え方です。</p>	<p>(5) モジュール分割技法とモジュールの独立性</p> <p>モジュールは、コンパイルして実行することができる最小単位のプログラムです。モジュール分割技法は、プログラムをモジュールに分割して、そのプログラムのモジュール構造を決定するための技法です。そして、その結果を、モジュール強度とモジュール結合度という尺度で評価します。分割したモジュールは、モジュール強度が強く、モジュール結合度が弱いものが独立性の高い良いモジュールとされています。この尺度の根本には、保守性という観点があります。仕様変更などでプログラムの修正が発生した場合に、できるだけ修正対象のモジュールを限定し、影響範囲を狭めたいという考え方です。</p>

## 大分類：4 開発技術 / 中分類：12 システム開発技術

1	システム要件定義	システム要件定義（機能、能力、業務・組織及び利用者の要件、設計制約条件、適格性確認要件ほか）、システム要件の評価 など
2	システム方式設計	システムの最上位の方式確立（ハードウェア・ソフトウェア・手作業の機能分割、ハードウェア方式設計、ソフトウェア方式設計、システム処理方式設計、データベース方式設計ほか）、システム方式の評価 など
3	ソフトウェア要件定義	ソフトウェア要件の確立（機能、能力、インタフェースほか）、ソフトウェア要件の評価、ヒアリング、ユースケース、プロトタイプ、DFD、E-R図、UML など
4	ソフトウェア方式設計・ソフトウェア詳細設計	ソフトウェア構造とコンポーネントの設計、インタフェース設計、ソフトウェアユニットのテストの設計、ソフトウェア結合テストの設計、ソフトウェア品質、レビュー、ウォークスルー、ソフトウェア設計の評価、プロセス中心設計、データ中心設計、構造化設計、オブジェクト指向設計、モジュールの設計、部品化と再利用、アーキテクチャパターン、デザインパターン など
5	ソフトウェア構築	ソフトウェアユニットの作成、コーディング基準、コーディング支援手法、コードレビュー、メトリクス計測、デバッグ、テスト手法、テスト準備（テスト環境、テストデータほか）、テストの実施、テスト結果の評価 など
6	ソフトウェア結合・ソフトウェア適格性確認テスト	テスト計画、テスト準備（テスト環境、テストデータほか）、テストの実施、テスト結果の評価 など
7	システム結合・システム適格性確認テスト	テスト計画、テスト準備（テスト環境、テストデータほか）、テストの実施、テスト結果の評価、チューニング、テストの種類（機能テスト、非機能要件テスト、性能テスト、負荷テスト、セキュリティテスト、リグレッションテストほか） など
8	導入	システム又はソフトウェアの導入計画の作成、システム又はソフトウェアの導入の実施 など
9	受入れ支援	システム又はソフトウェアの受入れレビューと受入れテスト、システム又はソフトウェアの納入と受入れ、利用者マニュアル、教育訓練 など
10	保守・廃棄	システム又はソフトウェアの保守の形態、システム又はソフトウェアの保守の手順、システム又はソフトウェアの廃棄 など

## 大分類：4 開発技術 / 中分類：12 システム開発技術

1	システム要件定義・ソフトウェア要件定義	システム要件定義（機能、境界、能力、業務・組織及び利用者の要件、設計及び実装の制約条件、適格性確認要件ほか）、システム要件の評価、ソフトウェア要件定義（機能、境界、能力、インタフェース、業務モデル、データモデルほか）、ソフトウェア要件の評価 など
2	設計	システム設計（ハードウェア・ソフトウェア・サービス・手作業の機能分割、ハードウェア構成決定、ソフトウェア構成決定、システム処理方式決定、データベース方式決定ほか）、システム統合テストの設計、アーキテクチャ及びシステム要素の評価、ソフトウェア設計（ソフトウェア構造とソフトウェア要素の設計ほか）、インタフェース設計、ソフトウェアユニットのテストの設計、ソフトウェア統合テストの設計、ソフトウェア要素の評価、ソフトウェア品質、レビュー、ソフトウェア設計手法（プロセス中心設計、データ中心設計、構造化設計、オブジェクト指向設計ほか）、モジュールの設計、部品化と再利用、アーキテクチャパターン、デザインパターン など
3	実装・構築	ソフトウェアユニットの作成、コーディング標準、コーディング支援手法、コードレビュー、メトリクス計測、デバッグ、テスト手法、テスト準備（テスト環境、テストデータほか）、テストの実施、テスト結果の評価 など
4	統合・テスト	統合テスト計画、統合テストの準備（テスト環境、テストデータほか）、統合テストの実施、検証テストの実施、統合及び検証テスト結果の評価、チューニング、テストの種類（機能テスト、非機能要件テスト、性能テスト、負荷テスト、セキュリティテスト、回帰テストほか） など
5	導入・受入れ支援	導入計画の作成、導入の実施、受入れレビューと受入れテスト、納入と受入れ、教育訓練、利用者マニュアル、妥当性確認テストの実施、妥当性確認テストの結果の管理 など
6	保守・廃棄	保守の形態、保守の手順、廃棄 など

※2021年10月26日、IPAのシラバス改訂に伴う変更