問5　【解答ウ】

ソフトウェア要件定義は，ソフトウェア構成品目ごとに技術的に実現可能か検証して，ソフトウェア設計が可能な技術要件（ソフトウェア要件）を定義する。代表的なソフトウェア要件に“データ定義及びデータベースに対する要件”があるので，「集計するデータ項目としてどのようなものが必要であるかを洗い出す」ことがソフトウェア要件定義の作業として適切である。

ア：ソフトウェア詳細設計及びソフトウェアコード作成の作業である。

イ：ソフトウェアユニットのテストの作業である。

エ：システム運用の作業である。

問6　【解答ア】

ソフトウェア詳細設計は，ソフトウェアコンポーネントを，ソフトウェアユニットのレベル（コンパイル及びテストを行う単位）に詳細化する工程である。したがって，ソフトウェア詳細設計で初めて決定する項目は，「コーディングを行う単位となる個々のプログラムの仕様」である。

イ，エ：ソフトウェア要件定義で決定する項目である。

ウ：ソフトウェア方式設計で決定する項目である。

問7　【解答ウ】

プログラムの標準的な記述方式（コーディング規約）は，ソフトウェアコード作成（プログラミング）で作成した「プログラム（ソフトウェアコード）の保守性を向上させること」を目的として規定する。作成したプログラムがコーディング規約に従って記述されているかは，コードオーディタというツールで検証する。

3.2　システム開発技術(3)　　ソフトウェア開発管理技術

問1　【解答ア】

プロセス中心アプローチは，ソフトウェアに必要な機能に着目して開発していく手法で，「業務の処理手順に着目して，ソフトウェアを分析する。」代表的なプロセス中心アプローチとして，ソフトウェアに必要な機能と，その機能で使用するデータの流れを分析／設計する構造化手法がある。

イ：データ中心アプローチに関する説明である。

ウ：ボトムアップアプローチに関する説明である。

エ：トップダウンアプローチに関する説明である。

問2　【解答イ】

・インヘリタンス（継承）

：クラス間でデータや手続きを引き継ぐ（継承する）ことである。

・カプセル化

：データを処理するために必要な機能（手続き）とデータを一体化し，オブジェクトとして扱う考え方である。カプセル化によって，利用者はデータがどのように扱われているか知らなくても，利用できるようになる。（正解）

・ポリモフィズム

：同じメッセージに対する振舞い（動作）が，クラスごとに異なることである。

・メッセージ

：オブジェクトに対して送る「こういう結果が欲しい」という処理要求である。

問３　【解答ウ】

・RAD（Rapid Application Development；高速アプリケーション開発）
　：少人数のチームで，開発を支援するツール（道具）を利用して，ソフトウェア（アプリケーション）をできるだけ短期間で開発していく方法である。
・スパイラルモデル
　：ソフトウェア（システム）を独立性の高い部分に分割して，部分ごとに設計／開発／テストを繰り返していく方法である。
・プロトタイピングモデル（プロトタイプモデル）
　：試作品（プロトタイプ）を作成して利用者に試用してもらい，評価／修正を繰り返しながら機能や画面などの仕様を確定していく方法である。（正解）
・リバースエンジニアリング
　：既存のソフトウェアなどを解析し，仕様や設計の情報などを取り出して仕様書を作成する手法である。ソフトウェアを再利用するリエンジニアリングの技術である。

問４　【解答イ】

・CMMI（Capability Maturity Model Integrated；能力成熟度モデル統合）
　：ソフトウェア開発組織のプロセスの成熟度を，５段階で評価するフレームワークである。
・SLCP（Software Life Cycle Process）
　：ソフトウェア開発とその取引の適正化に向けて，それらのベースとなる作業項目を定義し，取得者と供給者に“共通のものさし”を提供する共通フレームである。（正解）
・UML（Unified Modeling Language）
　：オブジェクト指向設計で用いられる表記法である。UMLの主な図式として，ユースケース図，クラス図，シーケンス図，アクティビティ図などがある。
・WBS（Work Breakdown Structure）
　：プロジェクトの目的を達成するために必要な作業を，成果物を主体に段階的に分割した階層構造図である。

問５　【解答イ】

ア：ウォータフォールモデルでは後戻りを想定していないので，前工程に戻って作業をやり直す必要があるソフトウェアの仕様変更に柔軟に対応することは難しい。
イ：ウォータフォールモデルは，ソフトウェアの開発工程を分割して，上流工程から下流工程へと進めていく方法である。そのため，工程単位で進捗状況を容易に把握できる。（正解）
ウ：ウォータフォールモデルでは，上流工程でできるだけ多くのバグを発見・除去し，品質を高める必要があるので，上流工程でのレビュー工数は多くなる。
エ：ウォータフォールモデルでは，開発工程の後半にならなければソフトウェア（プログラム）は作成されない。ソフトウェアを試用できるのはプロトタイピングモデルである。

問６　【解答イ】

リバースエンジニアリングは，既存のソフトウェアなどを解析し，仕様や設計情報などを取り出して仕様書を作成する手法である。ソフトウェアを再利用するリエンジニアリングの技術である。

ア：実験計画法に関する説明である。
ウ：BPR（Business Process Reengineering；ビジネスプロセスリエンジニアリング）に関する説明である。
エ：コンカレントエンジニアリングに関する説明である。

問7　【解答ウ】

アジャイルは，品質の高いソフトウェアを迅速に開発する手法の総称である。アジャイルソフトウェア開発宣言では，アジャイルソフトウェア開発の価値が，次のように宣言されている。

・プロセスやツールより，個人との対応に価値をおく。　･･･（イ）
・包括的なドキュメントより，動くソフトウェアに価値をおく。　･･･（エ）
・契約交渉より，顧客との協調に価値をおく。　･･･（ア）
・計画に従うことより，変化に対応することに価値をおく。　･･･「ウ」（正解）

問8　【解答ア】

ウォータフォールモデルの外部設計は，基本計画で定めたソフトウェアの要件を満たすために，ソフトウェアの機能を利用者の視点で定義する工程である。一般的な外部設計では，サブシステムの定義，入出力概要設計（「画面・帳票レイアウトの設計」），コード設計，論理データ設計などを行う。

イ：基本計画（要件定義）で行われる作業である。
ウ：内部設計で行われる作業である。
エ：プログラム設計で行われる作業である。

3．2　システム開発技術(4)　　テスト工程

問1　【解答ウ】

ホワイトボックステストは，「プログラムのアルゴリズム（内部構造）に着目して実施する」テストである。内部構造と密接な関係がある単体テストだけで利用される。

ア：システム適格性確認テストに関する説明である。
イ：システム結合テストに関する説明である。
エ：ブラックボックステストに関する説明である。

問2　【解答ア】

・インスペクション
　：モデレータという責任者が主体となって実施するレビューである。モデレータは，レビューの実施だけでなく，レビューで指摘された誤りの修正にまで責任を負う。（正解）
・ウォークスルー
　：成果物の作成者が主体となって実施するレビューである。
・共同レビュー
　：開発部門と利用部門の合意を目的とするレビューである。
・コードレビュー
　：ソースコードのバグを早期に発見することを目的とするレビューである。

問3　【解答ア】

単体テストは，「各モジュールが仕様書どおり正しく動作するか検証する」テストである。プログラミング工程を検証するテストとして実施され，ホワイトボックステストとブラックボックステストの考え方を利用してテストケースを設計する。

イ：システムテストの目的である。
ウ：運用テストの目的である。
エ：結合テストの目的である。

問４　【解答イ】

・ソフトウェア結合テスト
　：ソフトウェアユニット及びソフトウェアコンポーネント間のインタフェースを検証する。
・ソフトウェア適格性確認テスト
　：ソフトウェア要件として定義された適格性確認要件（テストケース）を用いて，ソフトウェア品目がソフトウェア要件どおり実現されているか検証する。（正解）
・ソフトウェアユニットテスト
　：ソフトウェアユニットがソフトウェア詳細設計の仕様どおり作成されているか検証する。
・リグレッションテスト（回帰テスト，退行テスト）
　：ある修正が，他の部分に影響を与えていないか検証する。

問５　【解答エ】

ブラックボックステストは，プログラムの仕様（入力と出力の関係）に着目して行うテスト手法である。ブラックボックステストのテストケースの作り方には，設計入力データを幾つかのグループに分割し，各グループからテストデータを選ぶ同値分割法や限界値分析法がある。

ア～ウ：ホワイトボックステストのテストケースの作り方である。

問６　【解答エ】

テストの目的は“バグ（誤り）を見つけること”であって，テストで発見されたバグを修正（デバッグ）することではない。そのため，開発者以外のテスト担当者がテストケースの設計やテストを行う。このとき，テスト担当者がバグを発見した場合，検出した「問題（バグ）を記録し，開発者（デバッグ担当者）に修正を依頼する」のが適切な対応である。

問７　【解答ア】

・システムテスト
　：システム全体が機能・性能・操作性を満たしているか検証するテストである。端末からの問合せのレスポンスタイムが目標値に収まることなどを検証する。（正解）
・ソフトウェア結合テスト
　：ソフトウェアユニット及びソフトウェアコンポーネント間のインタフェースを検証する。
・単体テスト
　：各モジュールが仕様書どおり正しく動作するか検証するテストである。
・ホワイトボックステスト
　：プログラムの内部構造（アルゴリズム）に着目して行うテストである。

問８　【解答ウ】

結合テストは，モジュール間のインタフェース（組合せ）を検証するテストである。問題の結合テストに要する時間の求め方は，次のとおりである。

手順１　６個のモジュールのインタフェース数を求める。これは６個から２個を選ぶ組合せの数なので ${}_6C_2$ で求められる。

$$ {}_6C_2 = \frac{6!}{2!(6-2)!} = \frac{6!}{2!4!} = \frac{6\times5\times4\times3\times2\times1}{(2\times1)\times(4\times3\times2\times1)} = 15 $$

手順２　結合テストの所要時間を求める。

結合テストの所要時間＝インタフェース数×１インタフェース当たりのテスト時間
＝15インタフェース×4時間／インタフェース
＝「60」時間