

～「基本情報技術者 午後試験対策学習法」～

1. 基本情報技術者：午後試験の出題と形式 ～何が問われるか～

試験時間：2時間 30分

出題形式：マークシートで多肢選択式(複数の選択肢から1個以上を選択)

13問中7問を解答(次表を参照)

合格基準：満点の60%

出題内容：事例問題(応用問題)

問番号	テーマ		配点	解答数・出題数
1～4	ハードウェア	5分野から 4問を出題	各12点	5問選択/7問出題 (12×5＝60点)
	ソフトウェア			
	データベース			
	ネットワーク			
	情報セキュリティ			
5	ソフトウェア設計			
6	マネジメント			
7	ストラテジ			
8	データ構造とアルゴリズム		20点	必須 (20×1＝20点)
9	C		各20点	1問選択/5問出題 (20×1＝20点)
10	COBOL			
11	Java			
12	アセンブラ			
13	表計算			

2. 基本方針を決める ～何をメインにするのか～

上の表に示すように、午後試験の問題は、3つのブロックで構成されている。

第1ブロック：選択問題

第2ブロック：アルゴリズム

第3ブロック：プログラム言語、または表計算

第1ブロックだけで100点満点中「60点」であるから、午後試験の攻略として第1ブロックは重要なブロックといえる。ただし、現実的に考えて、この選択問題5問で満点を取るのは容易ではない。したがって、第2ブロック、第3ブロックの得点率も上げて

おくべきであろう。

ところが、第2ブロック、第3ブロックは配点が20点と高いことからわかるように、難易度も第1ブロックに比べると高い。また、アルゴリズムやプログラム言語といった「アルゴリズム系」の攻略は時間がかかる傾向がある。



本試験までの限られた時間で合格レベルに達するためには、対策に当たって、自分なりの戦略を考えるとよいだろう。つまり、どのブロックを対策の中心に据えるか、という基本方針を決めることである。いくつか例を挙げてみよう。

【選択問題重視パターン】

「第1ブロックで点を稼ぐこと」を目標とした案である。選択する各分野での安定した得点率を磨くことに重きをおく。その分、アルゴリズムやプログラム言語にかかる時間が少なくなるので、第3ブロックではアセンブラ(CASLⅡ)や表計算を選択することで、それを補うとよいだろう。どちらも文法等の知識学習が短時間でできるので、演習時間を多く確保することができる。

または、プログラム言語の学習経験やプログラミング経験などがあり、第3ブロックに関しては、すぐに演習に入れるという方にもお勧めである。

【アルゴリズム系重視パターン】

「第2ブロック、第3ブロックで安定した得点を稼ぐこと」を目標とする案である。ある意味、理想的な攻略であるが、時間的リスクも大きいので、プログラミング経験者などのアルゴリズム系スキルをある程度もった方に向いているだろう。また、現時点までのアルゴリズム学習が順調に進んでいる方なら、狙ってもいいかも知れない。

もちろん、難易度は高めな分野であるうえに、ここで満点が取れても40点(=20点×2)にしかないなので、第1ブロックである程度得点する必要があるのは言うまでもない。

【アルゴリズム系回避パターン】

「第2ブロックのアルゴリズムの比重を少なくする」という案である。どうしてもアルゴリズムが苦手な本試験までに間に合いそうもない、という場合に苦肉の策として提案したい。「第1ブロックの選択問題」と「第3ブロックの表計算」を集中して対策することで、アルゴリズム系をある程度回避しても何とか合格できる正答率を確保するように対策する。ただし、60点/80点の戦いに近くなるので、それなりにリスクは大きいと考えられる。

どの案にも一長一短はあるので、自分の得意分野、学習経験やここまでの学習の進み具合などに応じて考えてみよう。

ただし、覚えておいて欲しい。22 年春の本試験のように、第 1 ブロックが難しかった場合には、第 2 ブロック、第 3 ブロックでの得点も重要になってくる。あるブロックをメインにするといっても、そこだけではなく**トータルでの得点力を挙げて欲しい**。つまり、【選択問題重視パターン】で対策する場合でも、アルゴリズム及びプログラム言語がほとんど 0 点では合格は難しい、ということである(できれば 40 点に対し、15～25 点は取って欲しい)。

3. 選択問題ブロックの攻略

【弱点分野は選ばない!!】

まず、やってはいけないのは、苦手な分野をわざわざ選ぶことである。午前対策の学習のときから感触が悪く、午後の演習もほとんど解けないような分野があったら、選択候補からはずしてしまおう。その分、他分野に時間をかけた方がいい。

【最低でも 6 分野を対策しておく!!】

第 1 ブロックのうち、問 1～4 は 5 分野の中から 4 分野が出題される。

ということは、毎回 1 分野出題されないことになる。自分が選択して対策していた問題が出題されないことも想定し、5 分野ぎりぎりではなく 1 分野多く対策しておこう。

【基礎知識は午前対策で得ているはず!!】

午後対策の知識として、新しいものがあるわけではない。午後問題で扱っているテーマは、ほとんどが午前対策で学習したテーマである。たとえば、最近ではハードウェアの「加算器」や、ストラテジの「SWOT 分析」が出題され、さらにはデータベースの正規化や SQL 文などのように“必須テーマ”となっているものもある。

また、前回 22 年秋本試験では、「論理演算」「基数変換」「シフト演算」などがハードウェアではない分野で使用されたが、これらは午前対策学習で学習する内容であるから、基本的な操作を習得していれば問題なかったはずである。

したがって、午前問題に出てくる事例問題や計算問題は、午後につながる内容として十分に演習しておいて欲しい。

【初出題テーマは問題文に従って処理!!】

実は、午後問題の中には、午前対策でも学習しなかったテーマが出ることもある。ただし、そのテーマに関してはほとんどの受験者も知らない内容であるから、あわてることはない。

それらの問題では、問題文で説明しているとおり処理を進めていけば解答が導けるように作られている。あわてずに問題文をよく読むことが大事である。

対策としては、過去本試験のこの手の演習を積むことで、問題に対処する力、すなわち適応力を磨く練習をしておくべきである。

【定番テーマを中心に、様々なテーマを演習!!】

旧試験から出題実績のある分野などには、定番ともいえるテーマが存在する。午後試験対策では、まず定番テーマを確実なものすることが大切である。それと同時に、なるべく多くのテーマの演習を積むと、より安定した力をつけることができるだろう。では、各分野のポイントを挙げてみよう。

- **ハードウェア**

論理回路、浮動小数点数、磁気ディスクのアクセス時間など必須テーマの出題が目立つ分野である。コンピュータ構成要素の内容や計算などは確実に身につけておこう。また、命令解説や稼働率なども注意しておきたい。

- **ソフトウェア**

OS のタスク管理や記憶管理からの出題が予想されていたが、未だそれらの出題がない。もちろんこれら OS の管理を対策からはずしてはならないが、バックアップや言語プロセッサなどの知識も準備しておこう。

- **データベース**

E-R 図、関係データベースが中心である。特に、E-R 図、正規化、主キー、各種制約などはおさえておきたい。そして、何とんでも「SQL 文」が重要である。SELECT 文については、IN や EXISTS などの副問合せも含めた十分な対策をしておこう。

- **ネットワーク**

ここ 2 回ほど計算問題が含まれており、注意が必要な分野である。定番テーマは、IP アドレス関連(ルーティング、サブネットマスクなど)と考えられるので、TCP/IP を中心に対策しておこう。

- **情報セキュリティ**

暗号化と認証、及びファイアウォールを中心に対策しておこう。SSL や PKI, 認証局などの知識ももっておきたい。

- **ソフトウェア設計**

プロセス流れ図を扱ったファイル処理は新試験でも定番テーマとなっている。過去本試験の演習により、マッチングや集計処理などを対策しておこう。

もう一つ、新試験になって初めて出題された「UML」に注意して欲しい。クラス図、シーケンス図、ステートチャート図などに目を通しておこう。

- **マネジメント**

今のところ、比較的解きやすい問題が多い分野である。問題文の中にヒントが隠れていてそれを見つける長文読解形式の出題がみられる。テクノロジ系中心の方も予備分野として過去 3 回の本試験演習はこなしておくといだろう。

- ストラテジ

戦略の決定に関する分析や計算を中心とした出題が多い。各種分析技法の知識とグラフの読み取り，企業会計に絡んだ計算などを対策しておきたい。

4. アルゴリズムブロックの攻略

【基本アルゴリズムの学習!!】

まず，テキストを用いて「基本アルゴリズム」の学習をしっかりと行うことが重要である。「基本アルゴリズム」には，

最大値(最小値)アルゴリズム

線形探索，2分探索

基本整列アルゴリズム(選択法，交換法，挿入法)

高速整列アルゴリズム(クイックソート，マージソートなど)

文字列照合，文字列置換

などがある。高速整列アルゴリズムなどは，それだけでテーマとなり得るし，線形探索などは別のアルゴリズムの部品として使用される。したがって，これらは必須知識である。

また，これらの学習を通して，

配列処理(添字の初期化，ループの条件，添字の更新など)

添字の特別な表し方

など，アルゴリズムの基礎となる力を身につけよう。

【データ構造の攻略!!】

データ構造には，

スタック，キュー，リスト，木

の4つがる。考え方は決して難しいものではないが，いざプログラム中で実装しようとする時，急に難易度が上がってしまう。

そこで，以下の内容をしっかりと学習しておきたい。

スタック：配列での PUSH，POP の操作。特にスタックポインタの使い方

キュー：配列の循環利用

リスト：リストを先頭からたどるループ。データの挿入・削除の処理手順

木：2分探索木へのデータの追加。ヒープソートの処理手順

【トレース!!】

新試験では，トレースが重要視されている傾向がある。そこで，演習時にも積極的にトレースを行ってみよう。問題で提示された例や，自分で作ったデータ(配列など)

を使って、プログラムの流れを追ってみるのである。

また、トレースにより、

初期値、ループの条件(どういった場合に終了するのか)、

ループ後の変数や配列の状態、添字の更新、具体的な処理手順

などを理解することもできる。

【演習!!演習!!そして解法アプローチ!!】

スキルを上げるために必要なのは、なるべく多くのアルゴリズム(プログラム)を読むことである。問題集はもちろん、過去本試験など多くの演習に取り組むべきである。

ただし、ただ闇雲に解いているだけでは、本試験での限定された時間内に正解を出すのは難しい。そこで、解法アプローチを意識して欲しい。つまり、どのようなアプローチを仕掛ければ、速く正解を導き出せるのか、を常に考えることである。たとえば、

- ・問題文とリンクしてみる。
- ・ブロックに分けて考える
- ・図を描いて、添字や配列の中を確認してみる

などがある。弊社の講座でも、この解法アプローチを組み込んだアルゴリズム演習を用意しているので、ご一考願いたい。

5. プログラム言語の攻略

【文法の学習は穴のないように!!】

プログラム言語の問題には、たまに文法知識で解ける空欄が出題されることもあるので、文法知識は確実なものにしておいて欲しい。また、文法なくしてプログラムは読めないのは当然といってよいだろう。

【アルゴリズムスキルがベース!!】

プログラム言語の問題は、そのプログラム言語で書かれたアルゴリズムの問題を解く問題である。したがって、アルゴリズムスキルが重要となる。

もちろんそういった意味では、プログラム言語の演習は、同時にアルゴリズムスキルの向上をもたらすといってもいいかも知れない。

【特有の処理を身に付けろ!!】

各プログラム言語には、それぞれ得意分野があり、言語特有の処理がある。それらをしっかり理解し、身に付けることが重要である。以下にそれらを挙げてみよう。

C

- ・ポインタを扱えること
- ・再帰的プログラム、関数
- ・画像処理、数学的な問題

Java

- ・ オブジェクト指向(クラス, 継承, 抽象クラス, インタフェースなど)
- ・ コンストラクタを含めた処理の流れが追えること
- ・ ジェネリクス, 拡張 for 文, enum 及び API

COBOL

- ・ 各種ファイル処理
- ・ コントロールブレイク処理, マッチング
- ・ SORT 文, 索引ファイル

CASL II

- ・ ビット操作
- ・ シフト, マスクパターン
- ・ 各種 JUMP 命令

6. 表計算の攻略

【絶対参照・相対参照 !!】

表計算で要注意なので, \$ をつけるべきか否か, すなわち絶対参照と相対参照の見極めである。解答を出すときに, 常に注意する習慣をつけよう。

【各種関数の使用法に慣れること!!】

表計算で使用する各関数に慣れることが大切である。演習の中で培っていくのがいいだろう。また, 問題冊子に掲載されている以外の関数については, 演習問題に登場したら, その場で使用法をしっかり理解しておくようにしよう。

【長文の問題に慣れること!!】

新試験になって 2 回, 3 回と経つうちに, 表計算の問題文が長くなってきている。いわゆる長文問題の様相を呈してきた。問題文からポイントを拾う訓練を, 演習の中でしておこう。また, ワークシートから必要なデータを取り出すことも重要であるから, ワークシートの確認作業を迅速に行えるように練習することも重要と言える。