

## テクニカルエンジニア（データベース） 受講生アドバイス

（直前でできること）

4月18日（日）の試験本番まで、残りわずかとなりました。

今回は「直前でできること」と題して、午後試験対策として本番前に必ず準備・確認しておいていただきたい事項のアドバイス、ならびに午前試験に出題される重要キーワードのおさらい解説をお届けします。

試験まであと残りわずか…。受講生の皆さんは、午後（記述式試験）を中心に対策講座を通して着実に実力アップしたことでしょう。

今回は全体的に直前に実施できるポイント解説を記載しました。試験まで残り時間を計画的に活用し、不得意分野の克服に取り組んでください。

データベース試験問題におけるポイントは以下です。試験では、これらを的確に定義できるかどうか、これらを適切に実現できるかが問われています。

### 午前のポイント

コンピュータシステム、システム開発と運用、データベース技術、セキュリティと標準化

### 午後 のポイント

正規化、SQL、トランザクション管理・同時実行制御（排他制御）、障害回復、リポジットリ、データベース設計、スペース量の計算、チューニング、分散データベース

### 午後 のポイント

コード設計、データベースの要件定義、概念データモデルの作成、論理データモデルの作成、  
物理データベースの設計、データ移行、データベースシステムの運用管理

本試験では、これを問題事例に適用させ、その事例に特化した具体的な解を記述する必要があります。その糸口を得るために、いわゆる一般論・常識・定石としてこれらの要件を整理しておきましょう。

## 1 2週間前にできること

### 1.1 最後の仕上げに午前対策

最後の総仕上げとして、午前問題を一通り解いて知識をリフレッシュするのも効果的です。

IT 共通分野などは、とにかく試験当日に記憶にとどまっていればよいのです。直前におさらいして、午前問題をクリアしましょう。

**<巻末の重要キーワードは必ず暗記してください>**

### 1.2 本番に備えて最後の調整

本試験日に知力、体力、精神力のすべてがピークに達するように調整してください。友人や、家族の方の協力も使えるものはすべて使うべきです。前日は早めに床につきましょう。

試験会場に持参するものは前日までに用意して念入りにチェックしましょう。また、会場までの交通、駅からの地図を今一度確認してください。

## 2 試験当日

### 2.1 合格点狙いでよい

はっきり申し上げます。所詮はペーパー試験と割り切ってください。午前で6割、午後で6割、午後で6割、これで合格です。100点を目指すのではなく、合格点狙いでゆきましょう。よく分からない小問の一つ、二つは早々と切り上げて部分点狙いでゆく位の大胆さが必要です。そういう問題に出くわしたならば、せめてキーワードの一つ吐き出して一矢報いる位の気構えで臨みましょう。空欄だけは避けましょう。解いているうちに答えが見つかることもありますから、気持ちの上で負けてはダメです。

制限時間内に合格ラインを超えるだけの解答をアウトプットすることを、試験当日の目標にしてください。徹頭徹尾、合格にこだわりましょう。

### 2.2 最後まで諦めない

試験に合格する人は「最後まで諦めないで全力を尽くす人」なのです。諦めた人から合格者は出ないのです。「試験終了です」と言われて答案を試験官に渡す瞬間まで、答案を書く権利はあなたにあるのです。最後の1秒まで諦めないでください。

<午後 / 試験テクニック>

### 2.3 午後

時間との勝負になります。限られた制限時間で、解答欄にキーワードを記述することで、得点を稼いでゆくことができます。それを首尾よく行うために効果的な戦略は、設問から読むというやり方です。

問題の選定には、時間をかけません。4問の中から1問を捨てる感覚です。自分は何が得意か不得手かを知っておくと選びやすいでしょう。しかし、苦手そうな分野であっても本文中にヒントがありますので、午前分野の知識が十分に備わっていれば何とかできます。

次の要領で解いてゆきます。

- (1) 設問を読んで出題テーマ、出題者の意図を汲み取る。
- (2) 事例に特化した設問 = 設問中に具体的なテーブル名とか属性名などが書いてある  
= 本文の中に手がかりがある
- (3) 一般論で答える設問 = 設問を見て大体答えが浮かぶ
- (4) 本文を読みながら、手がかりを探してゆく。
- (5) 出題テーマと設問の趣旨から、採点者が尋ねているキーワードを連想する。
  - 質問の意図 - Whyか? Whatか? Howか? - を読み違えないこと。
  - いざというときは、設問の文章をオウム返しに書いてみる。そこに技術用語を加えて脚色する。
  - とにかく空欄だけは避ける。
- (6) 出てきた答えに肉付けする。
  - 「理由」「から」、「問題点」「である点」、「機能」「する機能」

### 2.4 午後

問題の選定は、10分以内にしましょう。問題指示文、見出し、図表、設問の最初の指示文などを頼りに全体像を把握します。まったく歯が立ちそうにないもの、バックグラウンドとなる技術キーワードが浮かばないものが幾つありますか? そういった設問の少ない問題を選ぶのがよいです。事例を解析すると本文中に手掛りが見つかるケースが殆どです。それを見つけれられるかどうかは、技術のバックグラウンドがあるかどうかで左右されます。ですから、全部の設問に対して何とか解答が書けそうな方を選ぶのがよいです。

いかに短時間で要件定義を頭の中に叩き込んで、解答に反映させるかが勝負の分かれ目となります。

ポイントは次の通りです。

- ( 1 ) 問題の全体像を把握する。
- ( 2 ) 段落毎に本文を読み、要点をマークする。
- ( 3 ) 段落毎に要点をまとめ、ポイント同士をリンクで結ぶ。
- ( 4 ) 設問全部に目を通し、戦略を立てる。
- ( 5 ) 問題を解く                    論述問題は、問題事例に特化した解を書くこと。

以下、解説します。

#### ( 1 ) 問題の全体像を把握する

多くの場合、本文に見出し（山括弧で囲まれているもの）や副見出しがありますので、これに注目します。さらに、図や表も手掛りに全体を眺めます。まず始めにすべきことは、これから取り組むべき問題の全体像を把握することです。

#### ( 2 ) 段落毎に本文を読み、要点をマークする。

次に、段落毎に本文を読みます。読み進めていくうちに、気になる箇所が幾つかあるはずですが、

- ・ 要件定義
- ・ 書類やデータの流れ
- ・ 業務の流れ（データのフローに変換）
- ・ 例外的な事象
- ・ 顧客や上司から指摘された問題点や改善提案、現行システムでは対応できないような事態
- ・ 将来に問題発生が予見されるような不適切な処理、不自然な記述、極端な物言い

これら気になる箇所を、下線などを引いてマークしておきましょう。人によって、マークする箇所は異なる事があります。人それぞれ、経験やスキルが異なるからです。例えば、信頼性設計などを数多く手懸けてきた人にとっては「当たり前」であっても、異なる背景を持つ方にとっては「気になる箇所」と映るかもしれません。ですから、マークすべきポイントに唯一解はありません。「気になるな」と思ったらそこをマークすれば良いのです。

とはいえ、慣れないと全部が全部気になってしまいマークした箇所がたくさん出てしまい、かえって要点がぼやけてしまう事もあります。ある程度、過去問題を解いて感触をつかむ必要があるでしょう。

このようにマークする利点は一体何でしょうか？午後 は多くの文章を読んで解析する必要があります。設問の答えに結びつく要点が、たくさんの情報量の中に埋もれてしまいがちです。しかも、制限時間は 2 時間と大変短い中で解答する必要がありますから、ひとたび取りこぼすと後でピックアップするのが難しくなります。

ですから、指定された時間内で効率的にポイントを洗い出すため、気になる箇所をマークしておき、目立たせておくのです。

本文を読んで行くうちに、ポイントがリストアップされている箇所に出くわす事があります。このような箇所に丸数字や英文字などで番号を付けておきましょう。

また、図や表にも注目します。図表は、一目瞭然に情報を伝えます。また、詳細仕様、例外事項、制限事項などが密かに記述されている事が多いので、要チェックです。しかし、試験時間が限られていますから、最初に本文に目を通す段階では、図表の全ての文章や脚注に目を通す必要はありません。実際に設問を解く段階でもよいでしょう。

( 3 ) 段落毎に要点をまとめ、ポイント同士をリンクで結ぶ。

こうして、一通り段落を読み終わったら、その段階で一旦手を止めて、頭の中で要点を思い起こしてください。要点を整理してください。疑問点も出てくることと思いますが、それが後の段落で掘り起こされたり、設問で取り上げられたりするかもしれません。

必要に応じて、要点や疑問点を余白に書きとめましょう。

次に、ポイント同士をリンクで結びましょう。設問の答えに結びつく要点が、親切に一箇所にまとめられているとは期待できません。おそらく、最初の方の段落ではシステムの概要が載せられていて、きっと簡単な説明が収められていることでしょう。後の段落で所々に関連事項を忍ばせてある事でしょう。それらを確実に拾い出し、総合的に分析し、真のポイントを浮かび上がらせ、適切な設計を行う力が試される事でしょう。

そこで、大切になってくるのが、リンク付けです。相互リンクを貼っておくことで、本文全体に散在している情報をいもづる式に寄せ集めて効率的に漏れなく収集することができます。その上で適切な分析を行う事ができます。“素朴な疑問”や“ひらめき”など何か気がついた点があれば、どんどん書き込んでおくのが良いのです。もしかするとそれは解

答への重要な糸口であるかもしれませんが。せっかくここで思い付いたのに、本文の情報量の洪水に埋もれてしまってやがて忘れてしまったら、本当にもったいないことです。

(4) 設問全部に目を通し、戦略を立てる。

本文を読み終わったら、いきなり設問1の(1)から解くのではなく、設問全体に目を通しましょう。これから取り組むべき本当の相手は「設問」です。その真の姿を知り、戦略を立てて取り組む事により、高得点を狙う事ができるのです。今は頭の中に全体像が入っています。設問を読んで何かの手掛りやアイデアがひらめくかもしれません。それを余白に書きましょう。これから問題を1つ1つ解いてゆきますと、深みにはまってしまい、今せっかく思いついたことを忘れてしまうかもしれないからです。

それぞれの設問は独立性が高く、大抵1つか2つの見出しと関連があります。この段階でもリンク付け作業を行ってください。1つの設問の中で小問題は結合度が高いです。流れに沿って解くのもよいですし、一緒に考えてみたり、あえて順番を逆にしてみたりと、戦略を練ってみてください。

戦略を立てる事には、時間配分も含まれています。この設問はだいたいこれ位までに解いておく、という目標を書いておきましょう。一つの空欄の解を導くのに深みにはまってしまい、後の方の問題を解く余裕が無かった、という事が無いようにしましょう。どこに時間をかけ、どこを切り捨てても良いのか、その優先順位を判断する際に大切なのは、設問全体の戦略です。後の設問に結びつくようなポイントは是非とも時間をかけて解くようにしましょう。そのためには、設問を解く順番を変えてみるのもよいです。

(5) 問題を解く                      論述問題は、問題事例に特化した解を書くこと。

いよいよ、問題を解きます。事例解析と銘打っているわけですから、問題事例に特化した解が求められています。論述問題は次の要領で解くと良いでしょう。

- ・ 糸口を探るため、まずは一般論で考える。ひねった問題は出ない。素直に考える。
- ・ 次に、問題事例の中から関連するキーワード、キーフレーズを抽出する。どうしても見つからなければ一般論で考えるしかないが、これまでの傾向からして「半分以上の設問には本文にヒントが必ず用意されているはずである」と心得よ。
- ・ それらを組み合わせて下書きする。本文を引用することで、事例の特殊性、具体性が採点者にアピールできる。また、時間短縮にもつながる。

- ・ 問題全体、設問内で首尾一貫する。例えば、最初の小問で事例特有の問題点を指摘し、次の小問で改善案を述べる場合、その問題点と改善案が論理的にも技術的にも整合の取れたものでなければならない。
- ・ 制限文字ぎりぎりまで記述する。下書きは制限文字数より多めに書き、それを添削してゆき、洗練された訴求力のある文章を作り上げてゆく。しかし時間が限られているので、採点者に意味が通じればよい。
- ・ 誤字脱字は減点の対象である。特殊な用語は避ける。(本文中で用いられているものはよい)

事例解析の場合、解は唯一ではありません。「採点者よ。私の解を見よ！」という位の気迫で最後まで臨んでください。採点者を納得させた人が午後 の合格者です。すなわち、本試験の合格者なのです。

### 午前試験 重要キーワードのおさらい

過去の本試験午前問題にて繰返し出題されている，IT共通知識分野の重要キーワードのおさらい解説です。さっと斜め読みをしてみて，もし理解が曖昧なものがあれば，今のうちにテキスト等をあたって確認しておきましょう。

## コンピュータシステム

### 1 ハードウェア

#### 1.1 情報素子

##### メモリ素子の特徴(半導体)

SRAM: フリップフロップ回路で構成, 高速, キャッシュメモリに使用

DRAM: リフレッシュが必要, 遅い, 主記憶に使用

SDRAM: DRAMの改良版, 高速

##### メモリ素子の特徴(集積回路)

CMOS: 集積度を高められる

バイポーラ: 動作速度がはやい

#### 1.2 プロセッサアーキテクチャ

##### 高速化方式の特徴

パイプライン: 異なる命令ステージの並行処理

スーパースカラ: 同じ命令ステージでも並行処理可能

VLIW: 長い機械語命令(関連する命令をあらかじめまとめたもの)による並列処理

#### 1.3 メモリアーキテクチャ

##### 主記憶のアクセス時間短縮

メモリインタリーブ: 主記憶を複数バンクに分割し並行アクセスすることによる高速化手法

キャッシュメモリ: (平均読み取り時間の計算は頻出)

<p>平均読み取り時間          = キャッシュのアクセス時間 × ヒット率          + 主記憶のアクセス時間 × (1 - ヒット率)</p>
---

#### 1.4 補助記憶

##### RAIDの種類と特徴

RAID0: ディスク2台へのストライピング/  
冗長性なし, 狙いは性能向上

RAID1: ディスク2台へのミラーリング(同一データ書込)

RAID0+1: 性能と可用性の両立, 但し高価  
RAID2, 3, 4: 冗長符号用の専用ディスクを使用, この冗長化ディスクにアクセスが集中し性能上のボトルネックに/ディスクは最低3台必要

RAID5: 冗長符号(パリティ)用のディスクを特定せず, データディスクに分散格納することでボトルネックを解消/ディスクは最低3台必要

#### 1.5 入出力アーキテクチャと装置

##### パソコンの入出力インタフェース

SCSI: HDD, CD-ROM, スキャナ等の接続用

IDE(ATA): HDD 接続用規格の古典

EIDE(ATAPI): IDE の拡張版, HDD の容量制限緩和や CD-ROM サポート等が盛り込まれた

セントロニクス: もともとはプリンタ接続用だったが, 双方向通信も OK

GPIB: 計測機器接続用

USB: 1.5/12Mbps の汎用シリアルインタフェース/キーボード, マウス, モデム等さまざまな周辺装置が接続可能

IEEE1394: 100Mbps ~ の高速シリアルインタフェース

IrDA: 赤外線通信, 各種情報端末等のワイヤレス接続

#### 1.6 コンピュータの種類とアーキテクチャの特徴

##### スーパーコンピュータ

スーパーコンピュータはベクトルプロセッサやマイクロプロセッサを複数結合させて高速化をはかっている

## 2 基本ソフトウェア



## 2.1 オペレーティングシステム

### プロセス(タスク)とスレッド

プロセス(タスク): CPU資源, メモリ資源の割り当て単位/実行可能状態, 実行状態, 待ち状態の三つの状態間を遷移

スレッド: プロセス(タスク)内に複数存在/メモリ資源(アドレス空間)は共有, CPU資源(スタック, プログラムカウンタ, レジスタ)は別々に割り当てられ並行処理

### 割り込み

内部割り込み/外部割り込みの区別: プログラムの内側(=プログラムによって発生する割り込み)か, 外側(=ハードウェアによって発生する割り込み)かの違いによる分類

内部割り込み: SVC割り込み(システムコール), プログラム割り込み

外部割り込み: 入出力割り込み, タイマ割り込み, マシンチェック割り込み, コンソール割り込み

### 仮想記憶

仮想記憶の分類:

セグメンテーション(方式): アドレス空間の分割を可変長の「セグメント」単位に行う方式

ページング(方式): アドレス空間の分割を固定長の「ページ」単位に行う方式/主記憶装置から補助記憶装置への主記憶内容の退避(ページアウト)や呼び戻し(ページイン)

関連用語:

スラッシング: 実記憶容量が充分でない場合, プログラム多重度の高まりにより主記憶装置~補助記憶装置間のページ入替の頻発

スワッピング: 主記憶装置から補助記憶装置への主記憶内容の退避(スワップアウト)や呼び戻し(スワップイン)

ページフォールト: ページング方式で, 必要なペ

ージが主記憶に存在しない場合に発生する割り込み

## 3 システム

### 3.1 システムの構成技術

#### 信頼性向上のためのシステム構成

オンラインシステムシステム構成:

デュアルシステム: システムを完全二重化, 結果の照合/原子力プラントのオンラインシステム等

デュプレックスシステム: 故障時に主系(現用系)から副系(待機系, 予備系)に切り替え/平時は主系でオンライン処理, 副系でバッチ処理

現用系 - 予備系への切り替えの仕方:

ホットスタンバイ: 現用系と同じシステムを予備系でも起動して待機

ウォームスタンバイ: 予備系ではOSだけ起動して待機

コールドスタンバイ: 予備系の電源を切って待機

### 3.2 システムの性能

#### M/M/1の待ち行列モデル(頻出)

待ち行列〔概要〕

(分野: コンピュータシステム システムの構成と方式)

待ち行列理論では, その構成要素を次の3つに分けて考えます。

- ・集団: やがて窓口に来てくる, サービスを受けようとする集団
- ・待ち行列: 窓口が処理中でふさがったときに見える並び(窓口が空き次第, 順番にサービスを受ける)
- ・窓口: 実際にサービスを提供するところ

待ち行列の基本的な考え方は, 「母集団自体はき

わめて大きいですが、ある時点に着目してとらえると、実際にサービスを受けに来ている人はそのごく一部である」という事実に基づいて、「経済的・統計的な観点から窓口の数を決めよう」というものです。

母集団内の個々の要素を「トランザクション」と呼びます。銀行などの窓口の場合は待っているお客一人一人、コンピュータ処理の場合はCPUやI/Oなどのリクエストがこれに該当します（ついでに言えば、コンピュータ処理の場合、待ち行列は典型的には「入出力キュー」が該当します）。

待ち行列の長短は、サービスを受けに来る人の到着頻度とか、窓口でのサービスのさばき具合、あるいは窓口の数等によって左右されます。これらについて、A E・P M・S A 午前共通問題では、次の前提条件のもと出題されます。

- ・一定時間内の到着トランザクション数は全くランダム（これをポアソン到着と呼ぶ）
- ・あるトランザクションの処理時間の統計をとると指数分布のカーブを描く（この場合、一定時間内に処理できるトランザクション数に着目すると全くランダム）
- ・サービス窓口の数は一つ

このような、待ち行列を左右する条件を表記するための記法が「ケンドールの記法」です。それを用いて上記の前提条件を表したものが、問題文中に示される「M / M / 1 型待ち行列」という表現です。

〔待ち行列に関する公式〕

参考書にはいろいろな公式が掲載されていますが、以下の公式とそれぞれの意味を理解していれば

ば充分です。

(1) 平均到着率： $\lambda$ ，平均到着間隔： $T_a$

ある一定時間あたりに到着するトランザクション数が、平均到着率（ $\lambda$ ）です。「単位時間あたりのリクエスト数」とか「単位時間あたりの平均到着数」などと表現するテキストもあります。

同じ事柄を違った観点で表現したものが、平均到着間隔（ $T_a$ ）です。あるトランザクションが到着してから、次のトランザクションが到着するまでの時間間隔の平均です。

と  $T_a$  には次の関係があります。

$$\lambda = 1/T_a$$

(2) 平均サービス率： $\mu$ ，平均サービス時間（平均処理時間）： $T_s$

ある一定時間あたりにサービス可能なトランザクション数が、平均サービス率（ $\mu$ ）です。「単位時間あたりに処理可能なリクエスト数」とか「単位時間あたりの平均処理数」などと表現するテキストもあります。

同じ事柄を違った観点で表現したものが、平均サービス時間（ $T_s$ ）です。あるトランザクションが窓口に入って処理開始してからそのトランザクションが処理を終了し窓口を出る（それと同時に、次のトランザクションが窓口に入って処理開始する）までの時間間隔、すなわち「トランザクション1件あたりの処理時間」、その平均です。

$\mu$  と  $T_s$  には次の関係があります。

$$\mu = 1/T_s$$

(3) 窓口利用率 (あるいは単に利用率):  $\rho$

ある一定時間あたりに窓口が利用されている確率が、窓口利用率 ( ) です。次の式で表せます。

$$\rho = L/\mu = Ts/Ta$$

「窓口のキャパシティを越えた数のトランザクションが到着」、つまり「一定時間あたりに到着するトランザクション数 ( ) が、一定時間あたりにサービス可能なトランザクション数 ( $\mu$ ) を越えた状態 (  $> 1$  )」になると、待ち行列が際限なく大きくなるのがわかります。

(4) 処理中も含めた待ち行列中の平均トランザクション数:  $L$

次の式で定義されます。

$$L = \rho / (1 - \rho)$$

なお、これに類似のものとして、「処理中を含めない待ち行列中の平均トランザクション数 (= 純然たる待ち行列の長さ):  $L_q$ 」なる概念もあります。こちらは試験対策としては不要と思われます。

(5) 平均待ち時間

到着してからサービスを受けるまでの待ち時間、

言い換えれば順番待ちの時間です。自分の前にいるトランザクション数 ( 窓口内にあるものも含む ) に平均処理時間を乗ずれば求められますね。

$$Wq = L \times Ts$$

(6) 平均応答時間

到着してから処理が完了するまでの総待ち時間です。順番待ちの時間に自身の処理時間を加味すれば求められますね。

$$W = Wq + Ts$$

マスターすべき公式はこれだけです。確認の意味では具体的問題にチャレンジしてみる必要があるのでしよう

### 3.3 システムの信頼性・経済性

#### システム信頼性設計の考え方

フェールセーフ: 障害が発生した場合、その影響が安全サイドに働くように設計すること (例: 交通管制システムの障害時、混乱を回避するためにすべての交通を停止する指示が出るよう設計)

フェールソフト: 障害が発生しても、それによる処理能力の低下はあるものの、機能停止せずに維持できるように設計すること (例: デュアルシステム)

フォールバック (縮退): フェールソフトシステムで、障害が発生した装置を切り離して運転している状態

フルブールフ: 意図しない使われ方をしても故障しないこと

## 4 システム応用

### 4.1 データベースの応用

#### データウェアハウス

データウェアハウス：膨大な生データの集まり / ここからOLAPツールやデータマイニングツール等で意思決定用の情報を抽出

OLAP：対話型データ分析

データマイニング：大量データから規則性・法則性を見つける

データマート：多角的データ分析をサポートするための目的別データベース

## システム開発と運用

### 1 システムの開発

#### 1.1 言語

##### プログラムの特徴

リエントラント：再入可能 / データ域を別々に用意することで、複数プログラムからの同時呼び出し可能

(シリアリ)リユーザブル：(逐次)再使用可能 / 自身でデータ域の初期設定をすることで、再ロードせずに何度も使用可能

リロケータブル：再配置可能 / 主記憶の任意のアドレスに配置して実行可能

リカーシブ：再帰 / 自分の中から自分を呼び出し可能

#### 1.2 開発手法

##### ソフトウェア開発プロセスモデル

ウォーターフォールモデル：開発を上流から下流への一方向に進める

スパイラルモデル：開発プロセスを繰返し、各繰返しにおいてコストや品質などの評価を行い、リスクを最小にするプロセスをとる

インクリメンタル(成長型)プロセスモデル：ウ

ォータフォールモデルを繰返し、機能を段階的に提供

プロトタイプモデル：開発初期段階での試作品による確認により、後工程での手戻りを防ぐ

RAD：“早く、安く、高品質”のシステム開発を目的とした短期システム開発手法

ラウンドトリップ：オブジェクト指向開発において、分析・設計～プログラミングを行き来しながらトライアンドエラーで完成させていく方法

##### プロセス成熟度モデル(CMM)

ソフトウェア生産部門におけるプロセス(作業)水準

レベル0：プロセスが未定義

レベル1：初期…勘に頼っている

レベル2：反復可能…統計値等が活かされている

レベル3：定義…プロセスが統合的な組織運営によって実施されている

レベル4：管理…プロセス目標を念頭においた制御・改善の仕組みがある

レベル5：最適化：ビジネスニーズを念頭においた改善の仕組みがある

#### 1.3 要求分析・設計手法

##### データ中心アプローチ

プロセス(機能)中心設計：データの流れ・機能に着目した、従来型の設計

データ中心設計：データ構造をベースにプロセス設計を行う手法

オブジェクト指向設計：データとプロセスを一体化(カプセル化) / データ中心設計の発展形

#### 1.4 プログラミングの手法

##### オブジェクト指向の概念

インヘリタンス(継承)：サブクラス(特化)～ス

ーパクラス（汎化）の関係〔例：乗用車，バス，トラック 自動車〕 / サブクラスはスーパークラスの属性を引き継ぐ

集約：構成部品と製品の関係〔例：駆動装置，車体，車輪 自動車〕

ポリモルフィズム（多様性）：同一メッセージを送ってもオブジェクトごとに動作が異なること

カプセル化：データと操作（メソッド）をオブジェクトとして一体化

#### 1.5 テスト・レビューの手法

##### レビュー技法

インスペクション：組織的・制度的なレビュー，モデレータ（進行役）中心に進行

ウォークスルー：自主的レビュー，作成者中心

ラウンドロビンレビュー：レビュー参加者が持ち回りでレビューのリーダーを担当する方法

#### 1.6 開発管理

##### システム開発工数見積り

ファンクションポイント法：機能数（ビジネスアプリケーション開発では画面数や帳票数などがその目安になる）から開発規模を算出

COCOMO：過去の生産性データを参考に開発規模（行数）から開発人月を推定

## 2 システムの運用と保守

### 2.1 システムの運用

#### 非常用電源装置

UPS：Uninterruptible Power Supply；電源の瞬断や短時間の停電用

自家発電装置：長時間の停電用 / CVCF と組み合わせて使用

CVCF：Constant Voltage Constant Frequency；自家発電装置の出力（電圧・周波数の変動がある）を安定させるために使用

AVR：Automatic Voltage Regulator；自動定電圧装置 / 電圧変動が小さい場合に有効

## セキュリティ

### 1 セキュリティ対策

#### 1.1 機密保護・改ざん防止対策

##### 公開かぎ暗号方式

（分野：セキュリティと標準化 セキュリティ）

公開かぎ暗号方式以前から広く用いられていたのが、送信側と受信側のみが知っている、かつ両者だけの秘密のかぎを用いる「秘密かぎ暗号方式」です。この「秘密かぎ暗号方式」は、かつて「慣用暗号方式」と呼ばれていましたし、さらに送信側と受信側で共通のかぎを用いるため「共通かぎ暗号方式」と呼ばれることもあります。

さて、情報の秘匿のために暗号化する、という機能そのものに即して考えれば、秘密かぎ暗号方式でも問題はありません。しかし、情報通信の世界で暗号を使用するにあたり、本文の暗号化を行う前の問題として、「秘密であるべきかぎを、いかにして秘密裡に相手に送り届けるか」という矛盾が生じました。この答えが、公開かぎ暗号方式です。

公開かぎ暗号方式は、非可逆関数を用いています。私達は中学・高校などで「逆関数」を学びました。たとえば  $y=2x-4$  に対する  $y=x/2+2$  とか、あるいは  $y=5 \times x^2$  に対する  $y=\sqrt{x/5}$  の類です。逆関数は元の関数よりも複雑という印象がありますが、公開かぎ暗号方式は、この逆関数が複雑どころかそれを見出すことが（事実上）不可能、という特殊な数式（と数学の体系・理論）を用いて実現されています。

この特殊な数式に関する2つ(一対)のパラメータを、鍵として用意します。この一対の鍵それぞれは、元々はどちらがどちら(どちらが暗号用でどちらが復号用)といった違いはありませんが、面白いことに一方のかぎを用いて暗号化した場合、他方のかぎを用いないと復号できない(暗号化した際に用いたかぎでは復号できない)、という性質があります。

この一対のかぎのうち一方を「秘密裡に相手に送り届けることができない」情報通信などを用いて相手に送り届けます。この過程で第三者が盗聴する危険性も当然ありますが全くおかまいなしです。この送り届けたほうのかぎを「公開かぎ」と呼びます。そしてもう一方の手元の鍵は、自分だけが知るものです。これを「秘密かぎ」と呼びます。

ここでポイントとなるのは、次の2点です。

- ・ 情報の秘匿を目的として公開かぎ暗号方式を用いる場合、一対の鍵を用意するのは、受信者の側である
- ・ 秘匿すべき情報のやりとりに先立って、あらかじめ受信者から送信者に対し、鍵(受信者の公開鍵)の受渡しを行う

つまり、「自分に秘密裡に情報を送り届けて欲しい」と思う側(受信者)が一対の鍵を用意し、そのうちの片方の鍵をあらかじめ相手(送信者)に送り届けておくわけです。そして、受信者から送信者に対するかぎのやりとりが済ませてしまった後、実際の秘密情報のやりとりを行うわけですが、情報の送信者は「受信者の公開かぎ」を用いて情報を暗号化して送信、その受信者は、「受信者(自身)の秘密かぎ」を用いて暗号化された情報を復号します。

さて、ここで悪意の第三者に「受信者の公開かぎ」と「暗号化して送信した情報」の両方を入手されたとします。しかし送信者は第三者が入手したのと同じ「受信者の公開かぎ」によって暗号化していますから、悪意の第三者は情報を復号することができないのです!

ところで、一方の鍵を公開しても危険がないのは、前述した不可逆関数の特質によります。たとえばわれわれは、元の関数(たとえば「二項を乗ずる」という計算式)と、二項のうちの一項(たとえば5)、ならびにその答え(たとえば10)がわかれば、二項のうち残りの一項がいくつ(この場合2)であるかがわかります。それは、「二項を乗ずる」の「逆関数」である「積を一項で割る」によって求めることになります。ところが、公開かぎ暗号方式の世界では、この逆関数を見出すことができません。それゆえ、たとえ元の関数、関数で用いる一項、そしてその答えのうち、その一部を公開した(あるいはたとえそのすべてが既知になった)としても、秘密である残りの一項は決して求められないのです。それは逆関数を見出すことができないからです。これが公開かぎ暗号方式のからくり(のイメージ)です。

このような「かぎ配送にかかる問題点」をクリアした公開かぎ暗号方式の欠点は、計算の複雑さ(=計算量の多さ、すなわち遅さ)です。それゆえ、現実の情報通信では、そのすべてを公開かぎ暗号方式で行うのではなく、秘密かぎ暗号方式(=公開かぎ暗号方式より高速)と適材適所で使い分けしています。その具体例が「SSL」や「PGP」などのセキュリティプロトコルです。

ここまで理解できたなら、できれば上述のセキュリティプロトコル、そして電子署名や電子認証

の概要について確認しておきましょう。そこまでやっておけば、暗号化の仕組みについては完璧です。

### 公開かぎ暗号方式の応用

#### (1)機密保護

〔送信者〕本文を受信者の公開かぎで暗号化

〔受信者〕受信者の秘密かぎで復号

#### (2)送信者の認証（電子署名）:

〔送信者〕署名を送信者の秘密かぎで暗号化

〔受信者〕送信者の公開かぎで復号（復号できれば送信者は秘密かぎの持ち主つまり本人であることがわかる）

#### (3)送信者の認証 + 改ざん有無の確認

〔送信者〕本文からハッシュ関数を使ってメッセージダイジェストを作成し、これを送信者の秘密かぎで暗号化 / 本文は別途何らかの方法で暗号化する等して送信

〔受信者〕何らかの方法で暗号化する等して送信された本文からハッシュ関数を使ってメッセージダイジェストを作成 / これと送信者の公開かぎで復号することで取り出したメッセージダイジェストと比較（一致していれば改ざんがないことがわかる）

## 2 リスク管理

### 2.1 リスク分析

#### リスク管理

リスク管理の基本的考え方：一定のコストでリスク発生の場合の損害を最小にするためのプロセス  
リスク管理の方法：移転（契約等を通して他者に転嫁）、保有（準備金積み立てや安全度を見込むことで受容）、分離等

## 3 ガイドラインと関連法規

### 3.1 セキュリティに関するガイドライン

## コンピュータウイルスが持つ機能

コンピュータウイルス対策基準による定義：

自己伝染機能（他のプログラムやシステムに自身をコピーする）

潜伏機能（発病まで症状を出さない）

発病機能（破壊や利用者の意図しない動作を行う）

## 標準化

### 1 開発と取引の標準化

#### 1.1 開発プロセス，取引プロセスの標準化

ISO 9000シリーズ

ISO 9001 ~ 9003：品質保証モデル / この中でソフトウェア開発・供給・保守に適用されるものはISO 9001

ISO 9000 - 3：ISO 9001をソフトウェア開発・供給・保守に適用するための指針

ISO 9004：ISO 9001を取得した企業が継続的改善を目指す際の指針（ISO 9001等と対をなす関係）

### 2 情報システム基盤の標準化

#### 2.1 ソフトウェアの標準化

CORBA

CORBA：機能単位に分割したアプリケーション（オブジェクト）を分散システムで実行・通信・管理する技術仕様

ORB：CORBAのソフトウェア及び機能

OMG：CORBAを制定した標準化団体

### 3 データの標準化

#### 3.1 ファイル形式の標準化

#### 文書の構造化記述言語

SGML：さまざまな文書構造の記述について標準化した国際規格 / 機能が豊富すぎて実装が困難

XML：いわば“実装可能なSGML” / Web対応と拡張性を考慮した、SGMLのサブセット

HTML：これもSGMLのサブセット/Webの記述用として広く普及(この文書もHTMLで記述されている)

#### 4 標準化組織

##### 4.1 標準化組織

###### 標準化団体

ANSI：米国規格協会

ISO：国際標準化機構/工業・技術に関する国際規格の統一と国家間調整

ITU-T：国際電信電話諮問委員会/電気通信の標準化に関する勧告を行う国連機関

IEEE：米国電気電子工学会/LAN等にインタフェース規格制定に尽力

標準化は「キーワードのオンパレード」です。前述の項目だけでなく、IT共通知識体系を参照し、漏れのないように確認しておきましょう。

## データベース技術

最低限覚える必要がある用語ですので、言葉の内容だけではなく関連性を理解してください。

### 関係データベース

#### 1. 選択

表から条件を満たす組だけを抽出して、新しい表を作成する操作

#### 2. 射影

表から必要なフィールドだけを取り出して、新しい表を作成する操作

#### 3. 結合

特定のフィールドの値をもとに、複数の表のレコードを結びつけて、新しい表を作成する操作

### データの正規化

受注伝票や社員名簿などを例にとって具体的に

### 実施してみてください

関係データベースにおいて、表から集合や配列などを取り除き、データの整合性を保てるような表にすることを正規化するという。

#### 1. 非正規形

表や伝票など、ひとつのコードにおいて

#### 2. 第1正規化

繰り返し項目をそれぞれ別の行に分離する

#### 1. 第2正規化, 第3正規化

別表に持つべきものの分離。

第2正規化は、主キーの一部だけに依存する列の分離する。第3正規化は、主キー以外の列に依存する列の分離する

### SQL (代表的な命令)

関係データベースは行と列からなる2次元の表であるが以下の命令で操作を行うことができる

#### 1. SELECT データの照会

#### 2. INSERT 挿入

#### 3. DELETE 削除

#### 4. UPDATE 更新

2次元の表からレコードを抽出するには、次の命令を用いる

**SELECT 項目名 FROM 表名 WHERE 条件**

項目名：抽出したい項目の名前(2個以上も可)

表名：抽出のもととなる表の名前(2個以上も可)

条件：抽出すべきレコードの選択条件(ANDやORを用いてよい)



試験当日まであとひと頑張り，最後の追い込みを。そして試験当日は途中退場せず，試験時間満了まで全力を尽くしましょう。

**あなたなら必ず合格できます！ 合格を心から成功を祈っています！！**