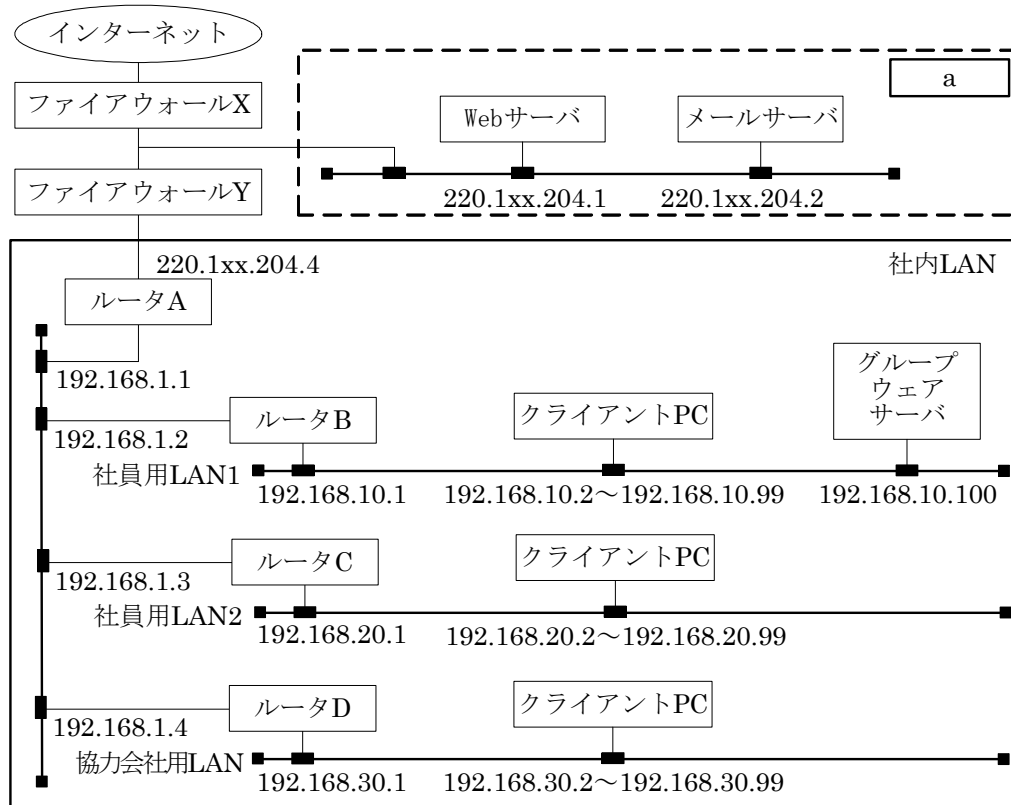


平成19年度 春期 ソフトウェア開発技術者 午後I問題

問1 ファイアウォールによるアクセス制御に関する次の記述を読んで、設問1～3に答えよ。

M社のネットワークは、図のような構成になっている。



注 220.1xx.204.1, 220.1xx.204.2 及び 220.1xx.204.4 の記述中の xx は、ある特定の値を表しているものとする。

図 ネットワーク構成

表1 M社の公開サーバ

公開サーバ名	役割
Webサーバ	自社のWebサイトを公開する。
メールサーバ	外部からの電子メールを受信する。 内部からの電子メールを送信する。 社内LAN上のクライアントPCからの電子メール取込要求に応答する。(社外からの取込みは行えない。)

〔インターネット側アクセス制御〕

M 社は自社のネットワーク上に、インターネットへの公開サーバを設置している。公開サーバの役割は、表 1 に示すとおりである。

M 社の公開サーバは、ファイアウォール X と Y に囲まれた領域に配置されている。このような構成にすることによって、外部からの不正アクセスでファイアウォール X が突破されても、社内 LAN に被害が及ばないようにすることができる。二つのファイアウォールで囲まれた領域は、 と呼ばれる。

ファイアウォール X 及び Y には、パケットフィルタリング型のファイアウォールを用いる。パケットフィルタリング型のファイアウォールは、受信したパケットに含まれる、送信元、送信先それぞれの と ，及び通信の方向を監視し、通信の通過、遮断を制御する。

〔社内 LAN 側アクセス制御〕

社内 LAN の中には、社員用 LAN1，社員用 LAN2 及び協力会社用 LAN が存在する。

社員用 LAN1 及び社員用 LAN2 からは、インターネットの Web へのアクセス、電子メールの送受信、及び社内 LAN 全体へのアクセスを可能にする。協力会社用 LAN からは、インターネットの Web へのアクセスと、グループウェアサーバへのアクセスだけを可能にする。

M 社で導入しているルータには、パケットフィルタリング機能が備わっている。協力会社用 LAN に関するアクセス制限は、 に装備されたパケットフィルタリング機能を用いて実現する。

〔パケットフィルタの設定〕

表 3～5 は、パケットフィルタの設定のうち、 へ向かう通信に関する設定と、協力会社用 LAN に対するアクセス制限に関する設定を抜粋したものである。

表中のポート番号の欄には整数値が、送信元及び送信先の欄には IP アドレスが入る。ここで使用しているポート番号とプロトコルは、表 2 のとおりである。ただし、any は任意のポート番号、anywhere は任意の IP アドレスを意味する。

また、二つ以上の設定が当てはまる場合は、表中の上にかかれた設定内容が優先される。

表 2 ポート番号表

ポート番号	プロトコル
25	smtp
80	http
110	pop3
52000	社内グループウェア

注 Web サーバ、メールサーバ及びグループウェアサーバが提供するサービスには、表 2 のプロトコルだけが使用される。

表3 ファイアウォールXの packets フィルタ設定

通信の方向：インターネット →

送信先ポート番号	状態	送信元 IP アドレス	送信先 IP アドレス
<input type="text" value="e"/>	許可	anywhere	<input type="text" value="f"/>
80	許可	anywhere	<input type="text" value="g"/>
any	禁止	anywhere	anywhere

表4 ファイアウォールYの packets フィルタ設定

通信の方向：社内 LAN →

送信先ポート番号	状態	送信元 IP アドレス	送信先 IP アドレス
<input type="text" value="e"/>	許可	anywhere	<input type="text" value="f"/>
80	許可	anywhere	<input type="text" value="h"/>
<input type="text" value="i"/>	許可	anywhere	<input type="text" value="f"/>
any	禁止	anywhere	anywhere

表5 の packets フィルタ設定

通信の方向：協力会社用 LAN → 社内 LAN

送信先ポート番号	状態	送信元 IP アドレス	送信先 IP アドレス
80	禁止	anywhere	192.168.10.1~192.168.10.254
80	禁止	anywhere	192.168.20.1~192.168.20.254
80	許可	anywhere	<input type="text" value="h"/>
<input type="text" value="j"/>	許可	anywhere	<input type="text" value="k"/>
any	禁止	anywhere	anywhere

[動的 packets フィルタリング機能によるアクセス制御]

社内 LAN からインターネットの Web を参照する場合、ファイアウォール X 及び Y は、社内からインターネットに向かうアクセス要求の packets と、インターネットから社内に向かう応答の packets を通過させる必要がある。

しかし、それらの packets を常に許可するようにしていると、侵入者が外部から を送りつけたときにでも、ファイアウォール X 及び Y は、それを通過させてしまう。それを防ぐため、ファイアウォール X 及び Y では、動的 packets フィルタリング機能を用いてアクセス制御を行う。

動的 packets フィルタリング機能を用いることで、送信したアクセス要求の packets に対応する応答の packets だけを通過させるように制御することができる。

設問1 本文中の ～ に入れる適切な字句を答えよ。

なお、 は図中にある機器を一つ選んで答えよ。

設問2 表2を参照して、表3～5中の ～ に入れる適切な字句を答えよ。

なお、 及び には、設問1の , と同じ字句が入る。

設問3 本文中の に入れる適切な字句を解答群の中から選び、記号で答えよ。

解答群

ア 送信先ポート番号が80, 送信元ポート番号が任意で、アクセス要求に対する応答の packets として作成された不正 packets

イ 送信先ポート番号が80, 送信元ポート番号が任意で、アクセス要求の packets として作成された不正 packets

ウ 送信先ポート番号が任意, 送信元ポート番号が80で、アクセス要求に対する応答の packets として作成された不正 packets

エ 送信先ポート番号が任意, 送信元ポート番号が80で、アクセス要求の packets として作成された不正 packets

問2 オブジェクト指向分析設計に関する次の記述を読んで、設問1～3に答えよ。

E社では、株価の現在値と、変動の表示を定期的に行う、図1に示すようなシステム（以下、株価通知システムという）を開発することになった。株価通知システムには複数の株銘柄が登録されており、証券取引所から定期的に送信される最新の株価に基づき、株価通知システムの株価が株銘柄ごとに更新される。株価通知システムは、株銘柄ごとに表示時点の最新株価（現在値）を数値で、株価の変動の様子（値動き）をグラフで定期的に表示する。

なお、当初は現在値表示と値動き表示の二つの表示機能だけを提供するが、将来的には株価分析用チャート表示など、新たな形式での表示機能の追加が予定されている。

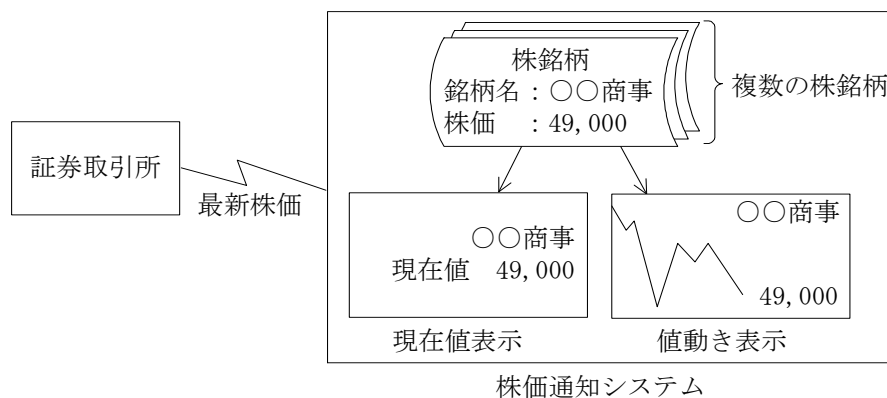


図1 株価通知システムの概要

株価通知システムの開発を担当するF君は、次の設計方針に従い、UML（Unified Modeling Language）のクラス図とシーケンス図を用いたシステムのモデリングを行うことにした。

〔設計方針〕

- ・株価を保持する部分と、現在値や値動きなどの表示を行う表示部は、それぞれを個別のクラスとし、表示部には株価の値を保持しない。
- ・すべての株銘柄は一つ以上の表示部と関連をもち、1回の株価更新で各表示を更新できるようにする。将来の表示機能追加に備え、表示部との関連は任意個可能とする。
- ・現在値表示や値動き表示などの表示機能に関しては、新たな形式の追加が予定されているので、適切な汎化を行い、部品化を図っておく。

F君は、一つの株銘柄に対する複数の表示機能を、株価の管理を行う株銘柄クラス（モデル）、株価の表示を行う株価ビュークラス及びそのサブクラス（ビュー）、株価の更新を制御する株価更新部クラス（コントローラ）の三つに分けて実装することにした。作成したクラス図を図2に示す。

株銘柄は、自身に依存するすべての株価ビューを、依存ビューの一覧としてもつ。株銘柄は、株価の変化に応じて、依存ビューの一覧に含まれるすべての株価ビューに対して表示を促すメッセージを送る。

株価ビューは、依存先の株銘柄を属性としてもつ。表示を促すメッセージを受けた株価ビューは、表示に必要な情報を依存先の株銘柄から取得し、表示を行う。

株銘柄と株価ビューの関連付けは、システム初期化部が関連付けのメッセージを送ることで行う。関連付けに関する処理のシーケンス図を図3に示す。また、定期的な株価更新と表示処理のシーケンス図を図4に示す。

株価通知システムをこのように設計することで、将来新たな形式の表示機能を追加する場合そのクラスを株価ビュークラスの **a** として実装し、そのインスタンスを対応する **b** クラスのインスタンスに依存ビューとして登録するだけでよい。**b** クラスでは、機能追加前と同じく、依存ビューの一覧に含まれるすべての株価ビューに対して表示の更新を促す処理を行うことによって、新たに追加された形式の表示についても適切に表示の更新が行われる。

この方針は、一般に MVC (Model-View-Controller) と呼ばれ、一つのモデルを複数のビューによって表示し、モデルの変化を、それを表示するすべてのビューに反映する必要がある場合に用いられる。これは、モデルとビューの **c** , 及びビューの汎化によって実現される。

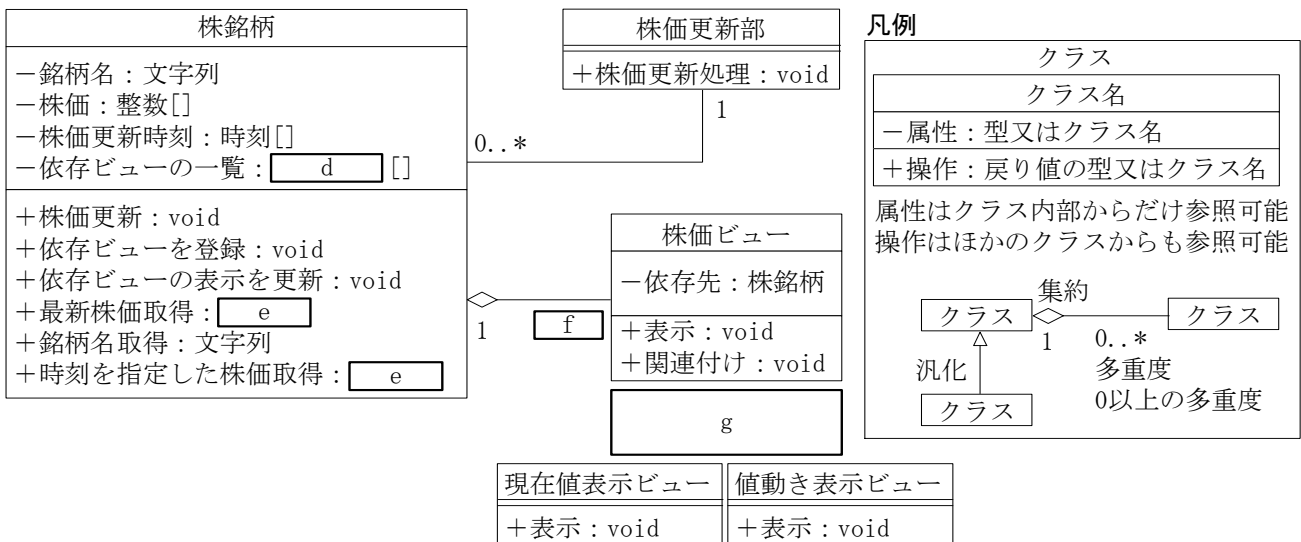


図2 クラス図

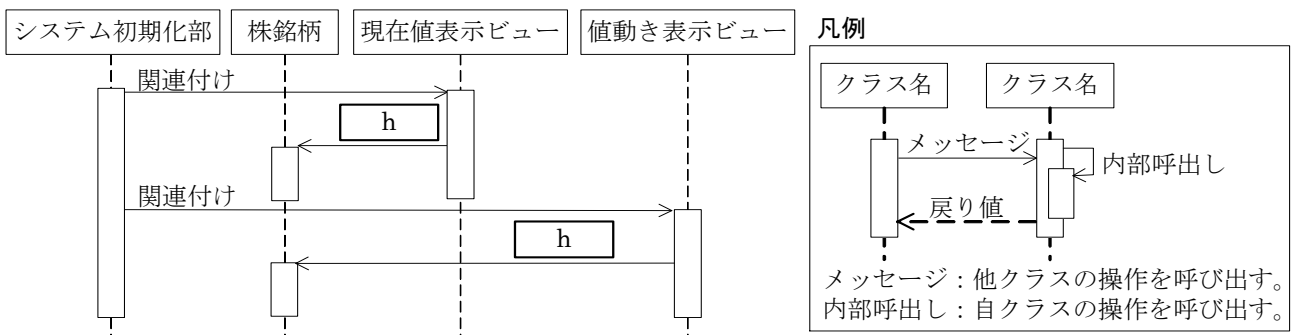


図3 株銘柄と対応する表示部の関連付けに関する処理のシーケンス図

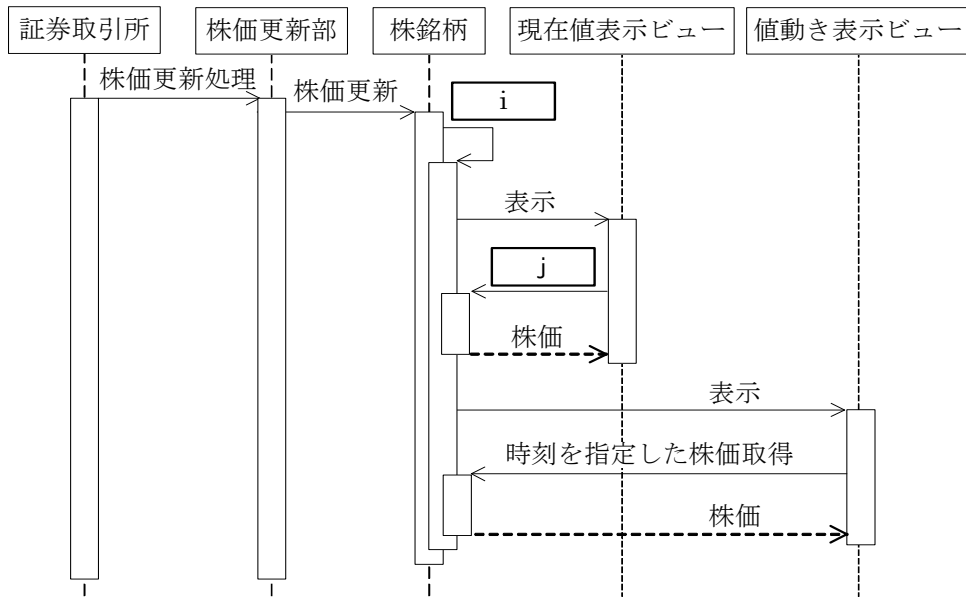


図4 株価更新と表示処理のシーケンス図

設問1 次の(1), (2)について答えよ。

(1) 本文中の **a** , **b** に入れる適切な字句を答えよ。

(2) 本文中の **c** に入れる適切な字句を解答群の中から選び、記号で答えよ。

解答群

- ア 依存関係 イ 親子関係 ウ 包含関係

設問2 次の(1), (2)について答えよ。

(1) 図2中の **d** , **e** に入れる適切な字句を答えよ。

(2) 図2中の **f** に入れる多重度及び **g** に入れるクラス間の関連を表す記号を、それぞれ凡例に倣い答えよ。

設問3 図3及び図4中の **h** ~ **j** に入れる適切な字句を解答群の中から選び、記号で答えよ。

解答群

- ア 依存ビューの表示を更新 イ 依存ビューを登録 ウ 株価更新
 エ 最新株価取得 オ 時刻を指定した株価取得 カ 銘柄名取得

問 3 電子データによる文書の保管に関する次の記述を読んで、設問 1, 2 に答えよ。

A 社では、役員会議議事録や会計報告などの重要な文書を電子データで保管することにした。それらの文書には、偽造や改ざんを防ぐために、署名者（文書の作成者や承認者）による公開鍵暗号方式を利用したデジタル署名を付ける。しかし、それだけでは、 の悪意による後からの文書作成や改変などを防ぐことはできないので、デジタル署名を付けた文書に対して時刻認証局（TSA）のデジタルタイムスタンプを取得し、その日時において文書が存在したことと、それ以降に改変されていないことを保証する。幾つかのデジタルタイムスタンプの方式のうち、ここでは公開鍵暗号基盤（PKI）による方式を採用し、信頼できる日時データをもつ信頼できる第三者機関の TSA を利用する。

なお、デジタル署名とデジタルタイムスタンプの検証に関し、信頼性の向上と運用の簡素化のため、信頼できる認証局（CA）の電子証明書を利用する。デジタル署名やデジタルタイムスタンプの有効期限は、電子証明書の有効期限に依存する。

〔デジタル署名の手順〕

デジタル署名では公開鍵暗号方式を利用するので、あらかじめ署名者の一対の秘密鍵と公開鍵を作成しておく。署名者の秘密鍵は厳重に保管する。

元のデータを で暗号化したものを署名のデータとする。署名の検証時には、その署名のデータを で復号して元のデータと比較し、一致すれば元のデータが偽造や改ざんされていないことが確認できる。 が漏えいしたり暗号が破られたりしない限り、署名が付けられた文書を他人が偽造や改ざんすることは困難である。

ただし、元の文書全体を公開鍵暗号方式で暗号化するのでは処理負荷が過大になるおそれがあるので、実際には元の文書をハッシングで一定の大きさに縮めたデータ（ハッシュ値）を暗号化し、それをデジタル署名のデータとする。ハッシングの性質として、ハッシュ値が になるような別の文書を求めることは困難である。

なお、署名の検証時に使用する の正当性を確認するために電子証明書を利用する。これは を含むデータに によるデジタル署名を付けたものであり、あらかじめ CA から取得しておく。 の正当性を確認するためには電子証明書のほか、 や証明書失効リスト（CRL）が必要である。

デジタル署名の手順の概要を図 1 に示す。

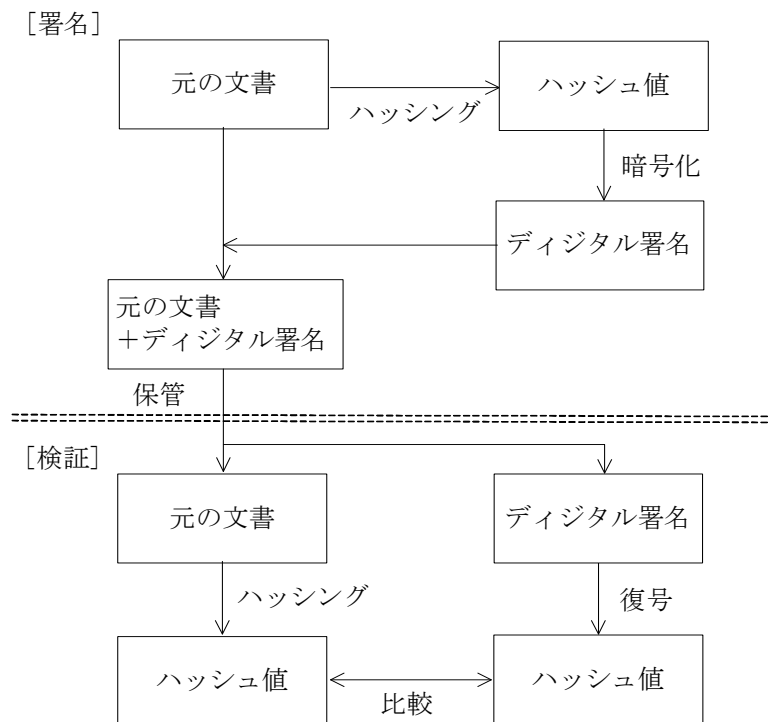


図1 デジタル署名の手順の概要

[デジタルタイムスタンプの手順]

デジタルタイムスタンプを取得するには、A社側で元の文書（デジタル署名を含む）のハッシュ値を求め、TSAに送る。TSAでは、そのハッシュ値と受取時の日時データを合わせたものにTSAのデジタル署名を行ってデジタルタイムスタンプを作成し、受取時の日時データと合わせて返送する。A社側では、元の文書とともに返送されたデータを保管する。

デジタルタイムスタンプの検証は、TSAのデジタル署名を検証することで行うことができる。

なお、TSAでデジタル署名を行うため、TSAでは一対の秘密鍵と公開鍵、及びそれについてのCAが発行する電子証明書を用意しておく。

デジタルタイムスタンプの手順の概要を図2に示す。

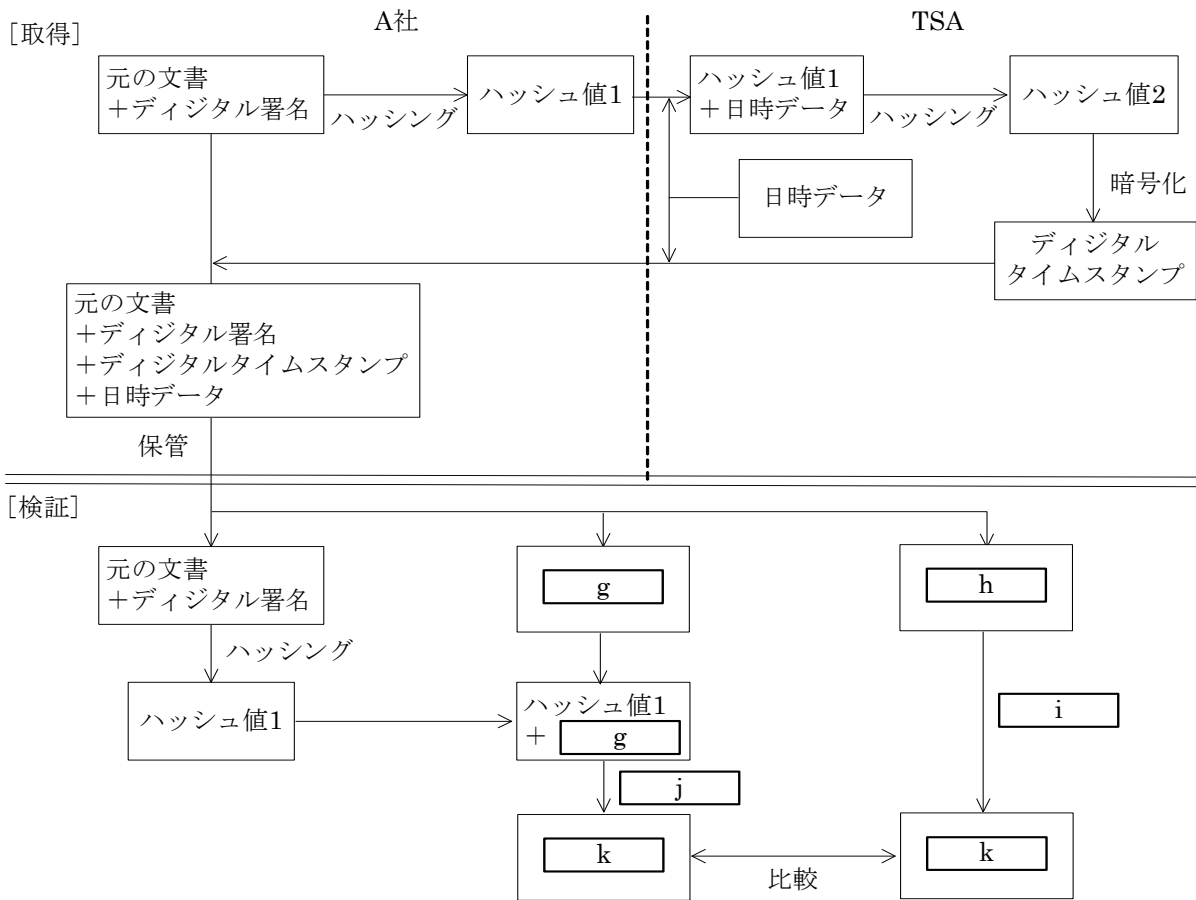


図2 デジタルタイムスタンプの手順の概要

設問1 次の(1), (2)について答えよ。

(1) 本文中の , に入れる適切な字句を答えよ。

(2) 本文中の , , , に入れる適切な鍵を解答群の中から選び、記号で答えよ。

解答群

ア CAの公開鍵

イ CAの秘密鍵

ウ TSAの公開鍵

エ TSAの秘密鍵

オ 署名者の公開鍵

カ 署名者の秘密鍵

設問2 図2中の ~ に入れる適切な字句を答えよ。

なお、図1又は図2中に使用されている字句を用いること。

問4 キャパシティ管理に関する次の記述を読んで、設問1, 2に答えよ。

イベントチケット販売会社であるQ社は、1年前にインターネットチケット予約システムを構築し、運用している。システムは、Webサーバ3台、AP（アプリケーション）サーバ1台、DB（データベース）サーバ1台からなり、顧客のWebブラウザからインターネットを介して利用されている。図にQ社のインターネットチケット予約システム構成を示す。

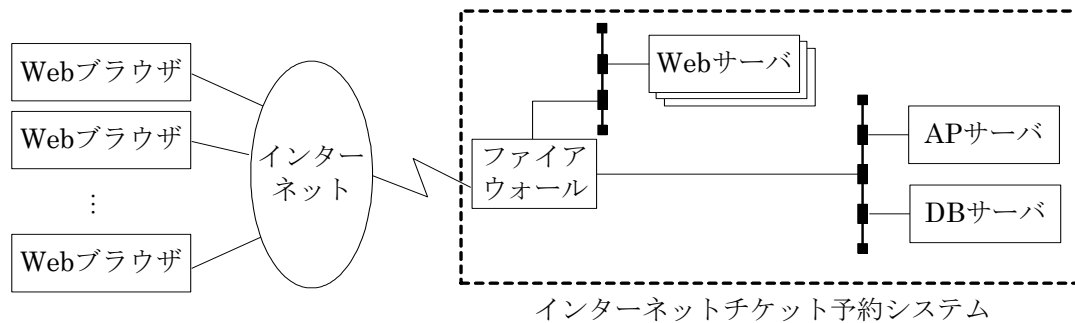


図 Q社のインターネットチケット予約システム構成

システム構築時は、SPEC（標準性能評価法人）やTPC（トランザクション処理性能評議会）などから公表されている テスト結果を参考に、システム構成の検討を行った。システム構成を確定させるまでの手順は、次のとおりである。

- ① 対象システム的前提条件、性能要件などを整理する。
- ② ①からシステム資源の を行う。
- ③ 内容を評価し、最適値を求め、システム構成を確定する。

このうち、APサーバでのトランザクション処理の前提条件及び性能要件は、次のとおりであった。これらからCPUの必要個数を算出した結果、 個を実装した。ここで、APサーバの処理能力は、CPU使用率が60%を超えずに処理できる1秒当たりのトランザクションの件数で表す。その時点で想定される平均トランザクション件数がこの件数を超えなければよい。

[APサーバのトランザクション処理の前提条件及び性能要件]

- ・トランザクション1件を処理するために必要なCPU処理時間：2.4ミリ秒
- ・APサーバの処理能力：1,000件/秒
- ・複数CPU実装時の性能低下は考慮しない。

Q社では、これまでのコンサートや演劇に加え、新たにスポーツ観戦のチケットを取り扱うことが決まり、インターネットチケット予約システムへのアクセスが増加することが予想された。今後18か月の間、APサーバが処理しなければならない平均トランザクション件数は、表1に示すように変化すると予測された。

表1 APサーバへの平均トランザクション件数の予測

	現時点	1か月後	2か月後	3か月後	6か月後	12か月後	18か月後
平均トランザクション件数 (件/秒)	800	880	1,000	1,440	1,890	2,410	2,600

システム資源の利用状況を調査した結果、APサーバがボトルネックとなる可能性があることが判明したので、APサーバのシステム増強策として、表2に示す三つの対策案を考えた。案1は、案2又は案3との併用も可能である。

表1の予測を基に平均トランザクション件数が処理能力を超えないように適時対策を実施していくものとする。対策実施に当たって、コストパフォーマンスを重視する場合は、“コスト/(性能現状比-1)”の値が最も小さい案を採用する。

なお、Webサーバ、DBサーバ、ファイアウォールなど、ほかのシステム資源は十分に余裕があり、増強の必要はないものとする。

表2 APサーバのシステム増強策

対策案	案1	案2	案3
内容	現在のサーバにCPUを追加 ※現在のCPUを含め、最大8CPU まで実装可能	高性能のサーバに リプレース	同型のサーバを複数追加し、 クラスタリング機能で接続
必要時間	2週間	1.5か月 ⁽¹⁾	3か月 ⁽¹⁾
コスト ⁽²⁾	1CPU追加ごとに200万円	1,200万円	1,700万円
性能現状比	$\frac{(\boxed{c} + n)}{\boxed{c}}$ n: 追加CPU数	2.7 ⁽³⁾	① 単独実施の場合: 3.5 ② 案1との併用実施の場合: 案1の性能現状比+2.5

注 (1) 案2及び案3は、案1を併用実施しても必要期間は変わらない。

(2) コストは、ハードウェア、ソフトウェアなどすべてを含んだ初期コストとし、運用コストは考慮しない。

(3) 案2はサーバのリプレースなので、単独実施、案1との併用実施とも性能現状比は同じ。

設問1 本文中の \boxed{a} , \boxed{b} に入れる適切な字句を解答群の中から選び、記号で答えよ。
また、 \boxed{c} に入れる適切な数値を答えよ。

a, bに関する解答群

- | | | |
|----------|------------|-------------|
| ア サイジング | イ スケーラビリティ | ウ ベーシックモデル |
| エ ベンチマーク | オ ペネトレーション | カ ロードバランシング |

設問 2 次の記述中の ～ に入れる適切な字句を答えよ。

なお , , については解答群の中から選び、記号で答えよ。

(1) 3 か月後までには現在の処理能力を超えてしまう可能性がある。3 か月後より後の状況は考えず、まず 3 か月後の状況に対処できる対策を考える。コストに着目した場合、最もコストが低い対策は である。コストパフォーマンスを重視した場合の対策は であり、このときのコストは 万円である。

(2) 3 か月後の状況に対処できるだけでなく、更に 18 か月後までの全期間を通して対処できる対策を考える。コストパフォーマンスを重視した場合の対策は であり、このときのコストは 万円である。

d, e に関する解答群

ア 案 1 (1CPU 追加)	イ 案 1 (2CPU 追加)	ウ 案 1 (3CPU 追加)
エ 案 1 (4CPU 追加)	オ 案 2	カ 案 3

g に関する解答群

ア 案 1	イ 案 1 と案 2 の併用	ウ 案 1 と案 3 の併用
エ 案 2	オ 案 3	

問5 小町算の解を求めるプログラムに関する次の記述を読んで、設問1～4に答えよ。

小町算は、1～9の数字をこの順番に並べて、数字の前に+、-の演算子を挿入し、値が100になる式を作るパズルである。演算子を挿入しなかった箇所は、数字の並びを一つの数値として扱う。例えば、
 $-1+2-3+4+5+6+78+9 (=100)$
 が求める式の一つである。+、-を入れて値が100になる式は、この式以外にも存在する。ここでは、値が100になるすべての式を出力するプログラムについて考える。

すべての式を求めるためには、123456789に+、-の演算子を挿入してできるすべての式を作り出し、その中で値が100になるものだけを出力すればよい。すべての式を作り出すために、次のように考える。
 演算子を挿入できる場所は、各数字の前の9か所である。各位置には、演算子を入れない、-を入れる、+を入れる、の3通りの場合が考えられる。挿入する演算子を表のように整数値に対応付けることにする。

表 演算子と整数値の対応

演算子	整数値
なし	0
-	1
+	2

各数字の前の演算子の種類を0～2の整数値で表記し、数字の1の前の演算子を最上位けたに、数字の9の前の演算子を最下位けたに対応させて順に並べると、3進数として表現できる。このような対応を決めると、一つの3進数に一つの式が対応する。

例えば、先に例として示した式の3進数表現は次のようになる。

$-1+2-3+4+5+6+7 \quad 8+9$
 1 2 1 2 2 2 2 0 2

演算子の種類に対応付けた0～2の数字を順に並べた121222202が、この式の3進数表現となる。

条件に合うすべての式を求めるためには、3進数で000000000～222222222のすべての数を生成し、各3進数に対応する式を計算して、値が100になる式を出力すればよい。

実際のプログラムでは、3進数を配列opで表現し、op[1]～op[9]を3進数の最上位けた～最下位けたにそれぞれ対応付ける。このとき、op[j]はj番目の数字であるjの前に置かれる演算子の種類を表すことになる。この方針に基づいて、図のプログラムを作成した。プログラムで使用される変数は既に別に宣言してあるとする。

```

/* 配列の初期化 */
for( iを1から9まで1ずつ増やす )
    op[i] ← 0
endfor

/* 小町算 */
    
```

```
while( op[1]が3より小さい )      /* 一番外側の while 文の始まり */
  s ← 0                          /* 式の値 */
  n ←     /* 演算する数値 */
  sign ← 1

/* 計算 */
for( iを1から9まで1ずつ増やす )
  if( op[i]が  に等しい )
    n ← 
  else
    s ← 
    n ← i
    if( op[i]が  に等しい ]
      sign ← -1
    else
      sign ← 1
    endif
  endif
endifor
s ← s + sign * n
/* 結果を表示 */
if( sが100に等しい )
  for( iを1から9まで1ずつ増やす )
    if( op[i]が  に等しい )
      “-” を出力
    else
      if( op[i]が  に等しい )
        “+” を出力
      endif
    endif
    i を出力
  endfor
  改行を出力
endif

/* 3進数に1を加える */
i ← 9
op[i] ← 
while( iが1より大きい かつ op[i]が  に等しい )
  op[i] ← 
  i ← i - 1
  op[i] ← 
endwhile
endwhile                          /* 一番外側の while 文の終わり */
```

図 小町算のプログラム

設問 1 3 進数 002002101 に対応する式を示せ。

設問 2 図のプログラム中の ～ に入れる適切な字句又は式を答えよ。

設問 3 図のプログラムに関して次の考察を行った。考察中の ～ に入れる適切な数値を答えよ。

式を構成する整数の中で、最大となる整数について考える。まず、途中に演算子が一切入らない場合は、式を構成する整数自身が最大となり、そのけた数は けたである。一方、途中に + 又は - の演算子が入る場合は、式中の最大の整数のけた数は けた以下であり、式中に現れる整数を加減算しても、演算結果のけた数は けた以下である。

以上のことから、式の演算結果は けたまでの整数であることが保証される。ここで、整数を 32 ビットの 2 進数で取り扱うとすると、表現できる整数の最大値は 10 けたの数なので、演算結果がけたあふれを起こすことはない。

設問 4 図のプログラムを実行すると、実質的に同じ式が出力された。(1)、(2) について答えよ。

(1) どのような式が出力されるかを、20 字以内で簡潔に述べよ。

(2) 一番外側の while 文の条件式を変更することで、この問題を回避できる。条件式をどのように修正すればよいか、20 字以内で答えよ。

問6 家具販売管理システムに関する次の記述を読んで、設問1～4に答えよ。

家具を製造・販売するC社では、3か月前に販売管理システム（以下、本システムという）を稼働させた。本システムは、顧客向けの販売明細などが添付された請求書、社内向けの販売実績レポートなどを提供する機能をもつ。システム稼働から3か月が経過したある日、販売実績レポートに問題が発生した。

[発生した問題とE-R図及びSQL文の確認]

問題が発生した平成19年2月分の販売実績レポートを図1に示す。

家具分類別，顧客企業別 販売実績レポート		
		出力日付：平成19年3月5日
平成19年2月 家具分類別 販売実績		
家具分類番号	家具分類名	販売実績（千円）
S01	収納	6,000
D01	机	4,000
C01	イス	2,600
C02	ソファ	2,000
⋮	⋮	⋮
合計		119,800
平成19年2月 顧客企業別 販売実績		
顧客企業番号	顧客企業名	販売実績（千円）
C0001	〇〇デパート	8,500
C0003	△△百貨店	3,400
C0008	●●商社	2,300
C0004	□□学習塾	1,800
C0010	××ホームセンタ	800
⋮	⋮	⋮
合計		120,000

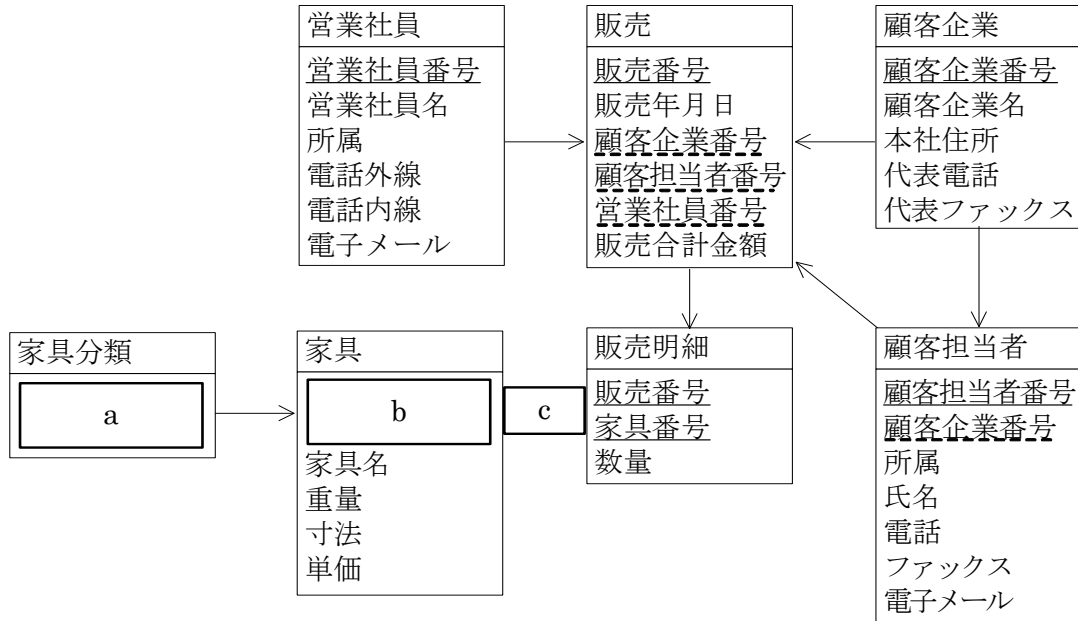
図1 平成19年2月分 家具分類別，顧客企業別 販売実績レポート

家具分類別と顧客企業別の販売実績の合計は同じになるはずだが、家具分類別の方が顧客企業別よりも200千円少ない金額になってしまっている。

この問題の原因を追究することになったDさんは、まずE-R図及び販売実績レポートの出力に使用しているSQL文を確認することにした。

本システムの E-R 図を図 2 に示す。

なお、本システムでは、E-R 図に合わせてエンティティ名をテーブル名（表名）、属性名を列名にして、適切なデータ型でテーブル定義した関係データベースによって、データを管理している。



凡例

- : 1対1
- : 1対多
- ←—— : 多対多

エンティティ名
属性名
属性名
属性名
⋮

注 属性名の実線の下線 _____ は主キー、破線の下線 - - - - - は外部キーを示す。主キーの実線が付いている属性名には、外部キーの破線を付けない。

図 2 本システムの E-R 図

図 1 の販売実績レポートを出力する二つの SQL 文は次のとおりである。ただし，“:指定月開始日”，“:指定月終了日”は、それぞれレポートの出力対象となる販売年月の開始日，終了日を表すホスト変数である。

・家具分類ごとに販売実績を集計する SQL 文 (SQL 文 1)

```

SELECT 家具.家具分類番号, 家具分類.家具分類名,
       d (家具.単価 * 販売明細.数量) AS 販売実績
FROM 販売, 販売明細, 家具, 家具分類
WHERE 販売.販売年月日 e :指定月開始日 AND :指定月終了日
      AND 販売.販売番号 = 販売明細.販売番号
      AND 販売明細.家具番号 = 家具.家具番号
      AND 家具.家具分類番号 = 家具分類.家具分類番号
f 家具.家具分類番号, 家具分類.家具分類名
    
```

示現塾 プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！

開催日、受講料、カリキュラム等、詳しくは、<http://zigen.cosmoconsulting.co.jp> 今すぐアクセス！！

ORDER BY 販売実績 DESC

- 顧客企業ごとに販売実績を集計する SQL 文（SQL 文 2）

SELECT 販売.顧客企業番号, 顧客企業.顧客企業名,

(販売.販売合計金額) AS 販売実績

FROM 販売, 顧客企業

WHERE 販売.販売年月日 :指定月開始日 AND :指定月終了日

AND 販売.顧客企業番号 = 顧客企業.顧客企業番号

販売.顧客企業番号, 顧客企業.顧客企業名

ORDER BY 販売実績 DESC

〔原因の推測と調査の実施〕

D さんは二つの SQL 文に、そこから出力される販売実績に差異を生じさせる要因がないかを検討した。SQL 文 2 は、販売時の実績を集計しているだけであるので、入力ミスがないとしたら、実際の販売金額を出力していると判断できる。SQL 文 1 は、現在の家具の単価から算出された金額を集計しているため、販売時の単価が現在の単価と異なる場合、販売時の実績と異なる金額を出力してしまう可能性が考えられる。その販売時の単価と現在の単価との相違が原因ではないかと推測した。

そこで D さんは家具の単価の設定方法について調査を行ったところ、次の 3 点に整理することができた。

- 家具の単価は四半期に 1 回程度の頻度で不定期に改定される。最後に改定が適用されたのは平成 19 年 3 月 1 日である。
- 営業時間中に単価が改定されることはない。家具テーブルの単価は、改定が適用される前日のオンラインシステムが稼働していない営業時間外に更新される。
- 営業部門から、家具を販売するごとに販売単価を個別に設定できるようにしたいという要望が出ている。しかし、現状では、顧客ごと、又は販売ごとに単価を変更することはできない。

これらの家具の単価の設定方法から、今回の問題は平成 19 年 3 月 1 日に実施された単価改定が原因であることが判明した。そこで、まず、この問題を回避するために、データ構造を見直した次の案を検討することにした。

〔単価改定履歴をもつ案〕

家具ごとに単価改定履歴をもつことができるように、家具エンティティから属性“単価”を削除し、単価管理エンティティを追加する。図 3 は追加した単価管理エンティティと家具エンティティとの関係を示した E-R 図である。

なお、図 3 中の には図 2 中の と同一のものが入る。

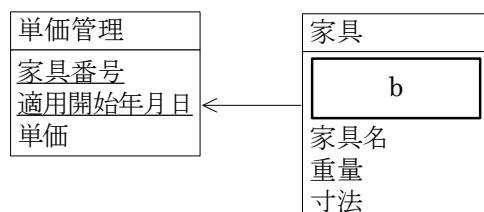


図3 単価管理エンティティを追加したE-R図（一部）

さらに、今回の問題を回避するだけでなく、営業部門から出されている要望を実現するために、次の案も検討した。

〔販売単価を個別に設定可能にする案〕

家具を販売するごとに営業担当者が販売単価を個別に設定できるように、 エンティティに属性 “” を追加する。

設問1 図2のE-R図中の ～ に入れる適切な属性名及びエンティティ間の関連を答え、E-R図を完成させよ。

なお、エンティティ間の関連及び属性名は、図2の凡例に倣って示せ。

設問2 本文中の ～ に入れる適切な字句を答えよ。

設問3 〔単価改定履歴をもつ案〕を採用した際の販売明細を出力するためのSQL文を次に示す。SQL文中の ～ に入れる適切な字句又は式を答えよ。

ただし、販売明細の一覧には、“家具番号、家具名、販売数量、単価、小計”の一覧が、家具番号の昇順で出力される。また、出力対象となる販売の販売番号を表すホスト変数は“:販売番号”とする。

なお、列を表す場合、テーブル名の別名を省略してはならない。

```
SELECT HM.家具番号, KG.家具名, HM.数量, ,  
        AS 小計  
FROM 販売 HA, 販売明細 HM, 家具 KG, 単価管理 TN  
WHERE   
      AND HA.販売番号 = HM.販売番号  
      AND HM.家具番号 = KG.家具番号  
      AND KG.家具番号 = TN.家具番号  
      AND TN.適用開始年月日 =  
      ( SELECT   
        FROM 単価管理 TN2  
        WHERE TN2.家具番号 =   
          AND TN2.適用開始年月 <=  )  
ORDER BY HM.家具番号
```

設問4 〔販売単価を個別に設定可能にする案〕中の , に入れる適切なエンティティ名及び属性名を答えよ。