

問 19 - 4

(H10 春-PE 午後 問 3)

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 ～ 6 に答えよ。

〔問題の背景〕

M ソフトウェア社では、現在、ソフトウェアの受注、開発、売上げなどを管理するため、部ごとに業務管理システムを開発し、運用している。今回、全社的に業務効率向上を図ることを目的として、このような個別業務管理を見直し、併せて、一人 1 台のパソコンとイントラネットの全社導入を機会に業務管理システムを統合することになった。このため、社内の専門家による業務改革検討会を組織し、オブジェクト指向分析・設計方法を利用して新しい業務管理システムの要求仕様を作成することになった。オブジェクト指向分析・設計方法には、OMT 法を使用することにした。なお、この問題に関する OMT 法の表記法は問題の末尾にある説明を参照のこと。

〔業務プロセスのシナリオ分析〕

問題の範囲を明らかにするために、見積依頼から入金に至る業務の流れを分析した。例外的な場合は除き、各部の間で交換される主要な情報をまとめた結果は図 1 に示すとおりである。全社にわたる業務の流れを明らかにするには、見積依頼から入金に至る一連の業務プロセスを分析する必要がある。

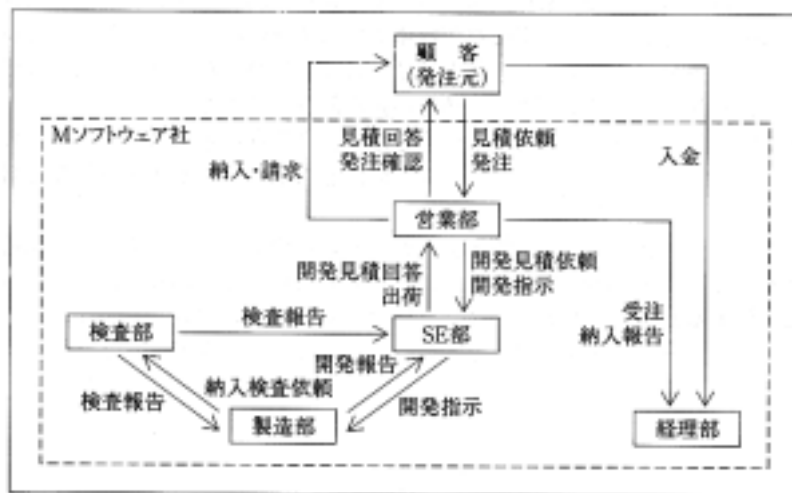


図 1 各部間の情報の流れ

そこで、図 2 に示すユースケースの事象トレース図を作成した。ユースケースは一般にシステムの利用手順を示したものである。OMT 法では、ユースケースをシナリオとして文章で記述したり、事象トレース図として上から下へと進む時間軸に沿って主要なオブジェクト(アクターともいう)間の情報交換として図示したりする。



図 2 ユースケースの事象トレース図 (部分)

図 1 の各部間情報の流れと図 2 の事象トレース図を作成した結果、見積依頼から入金に至る一連の情報交換が業務プロセスを形成していることが分かった。この結果から、業務効率を向上させるには、この業務プロセスに沿って情報の交換や共有を適切に支援する必要があると予想された。そこで、具体的な要求を明らかにするため、各部内の業務のユースケースをシナリオとして記述し、分析した。図 3 に、SE 部における開発見積の詳細シナリオの主要な部分を示す。このシナリオを見ると、情報を紙で渡しているの、SE 部では開発見積依頼書と仕様書を営業部から受け取った後、その内容を SE 部の見積データベースに登録するといった非効率的な作業を行っていることが分かる。これらの作業は、見積データベースを部間で共有することによって改善が期待できる。

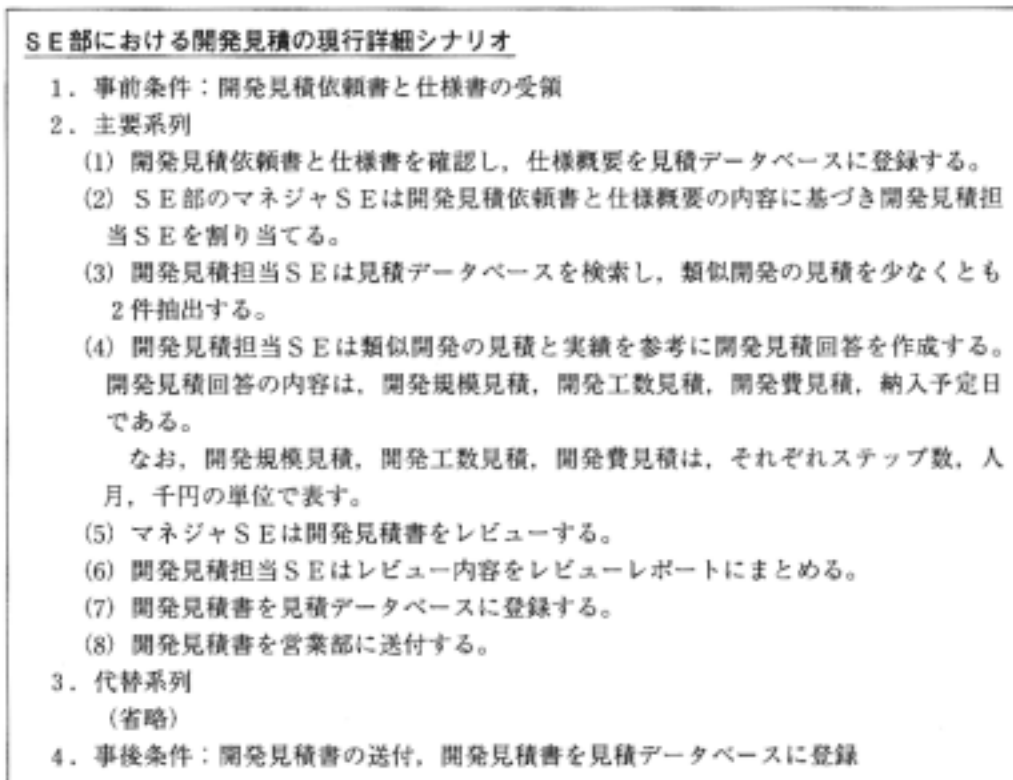


図 3 SE 部における開発見積の詳細シナリオ

〔オブジェクト分析〕

一方、各部で現在管理しているデータベースを分類した。一つのデータベースをオブジェクトとして列挙した結果は、図 4 に示すとおりである。

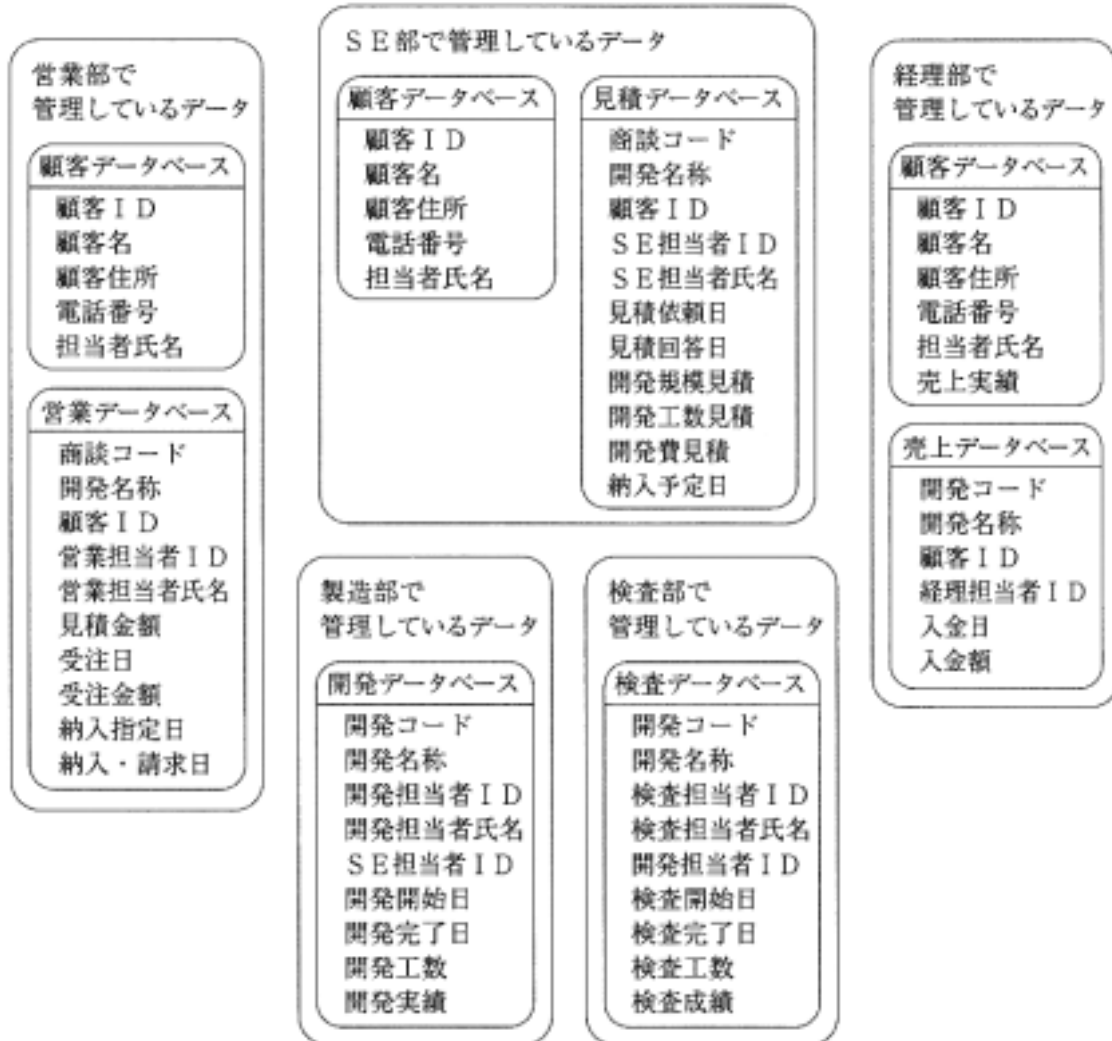


図 4 各部で管理しているデータ

これらの管理データには重複があり、部ごとにデータの再入力を行っている。また、データの更新漏れなどの問題も発生している。そこで、現行のデータの移行も考慮しながら、これらのデータを分析、整理することにした。これによって、対象システムを理解し、オブジェクト抽出の手掛かりが得られると判断した。まず、図 4 の各データを見ると、各部でデータが重複していることが分かる。例えば、営業部、SE部、経理部で顧客データが重複しており、このデータは一つにまとめることができる。このように各データは、次のようにグループ化することができる。

- ・ 全社にかかわるデータ：顧客データベース、見積データベース
- ・ SE部、製造部、検査部にかかわるデータ：開発データベース、検査データベース

- ・ 営業部, 経理部にかかわるデータ : 営業データベース, 売上データベース

これらのデータベースには, 各部の業務の主体である営業や開発の管理データと業務の成果物として作成される見積やソフトウェアに関するデータがある。これらのデータの中で, 各部間で共有する必要があるのは成果物のデータである。したがって, 業務に関するデータと成果物のデータを分けて, 独立したクラスにすることにした。このような分析から, クラスとその関連, 多重度を含むクラス図(OMT 法ではオブジェクト図という)の第 1 案として, 図 5 の初期クラス図を得た。多重度は関連するクラスのオブジェクトの数の対応を表す。

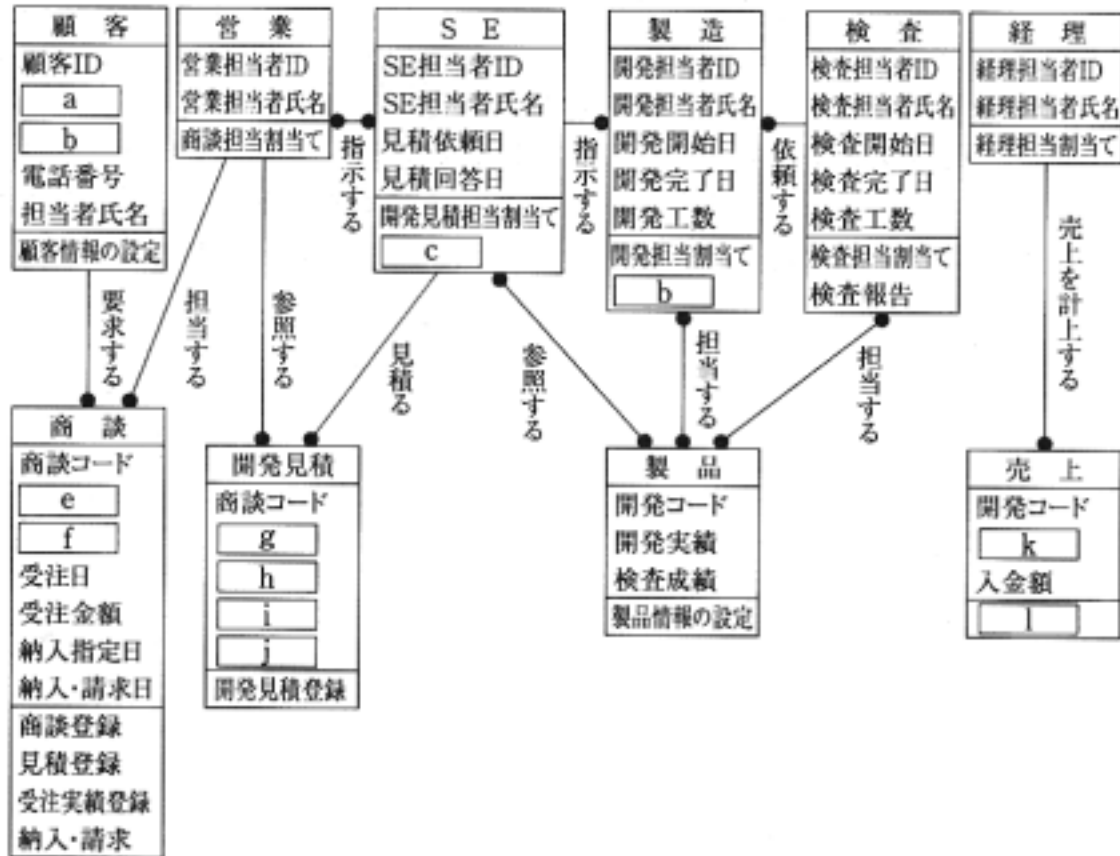


図 5 多重度と関連を考慮した初期クラス図 (オブジェクト図)

〔クラス構造の洗練〕

図 5 のクラスを見ると, 同一の属性をもつクラス群がある。このような同一の属性は, より抽象的な概念や, 業務や組織などの枠組みを定義する上位概念としてまとめることができる。そのような整理を行って, 複数のクラスに分散している共通の属性を一つのクラスにまとめると, システムの構造の見通しが良くなるとともに, 設計時の考慮範囲や設計変更時の変更範囲が局所化されるので, モデルが洗練できる。この方針に基づいて図 5 を見ると, 図のクラスは, 上段と下段の二つのグループに分けることができそうである。すなわち, 図の上段にあるクラス群は, 業務プロセスに沿って各部で業務を行う人のモデルと解釈でき, 下段のクラス群は, 業務プロセスの入出力となるプロダクトのモデルと解釈できる。



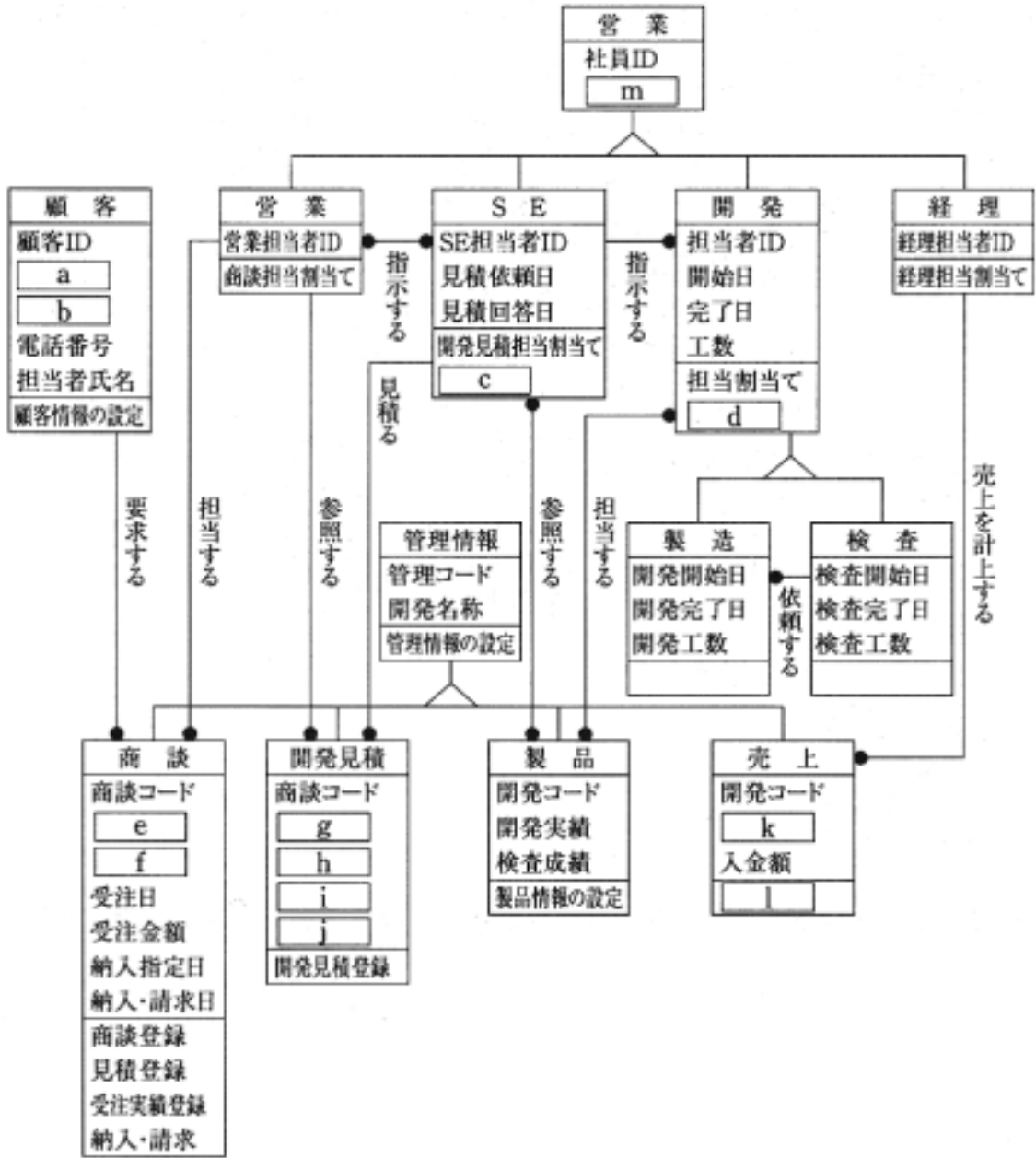


図7 洗練されたクラス図 (オブジェクト図)

〔インタフェース仕様の定義〕

分析結果の成果物として、図7のクラス図に示した各クラスについて、そのインタフェースを図8のインタフェース仕様書としてまとめることにした。

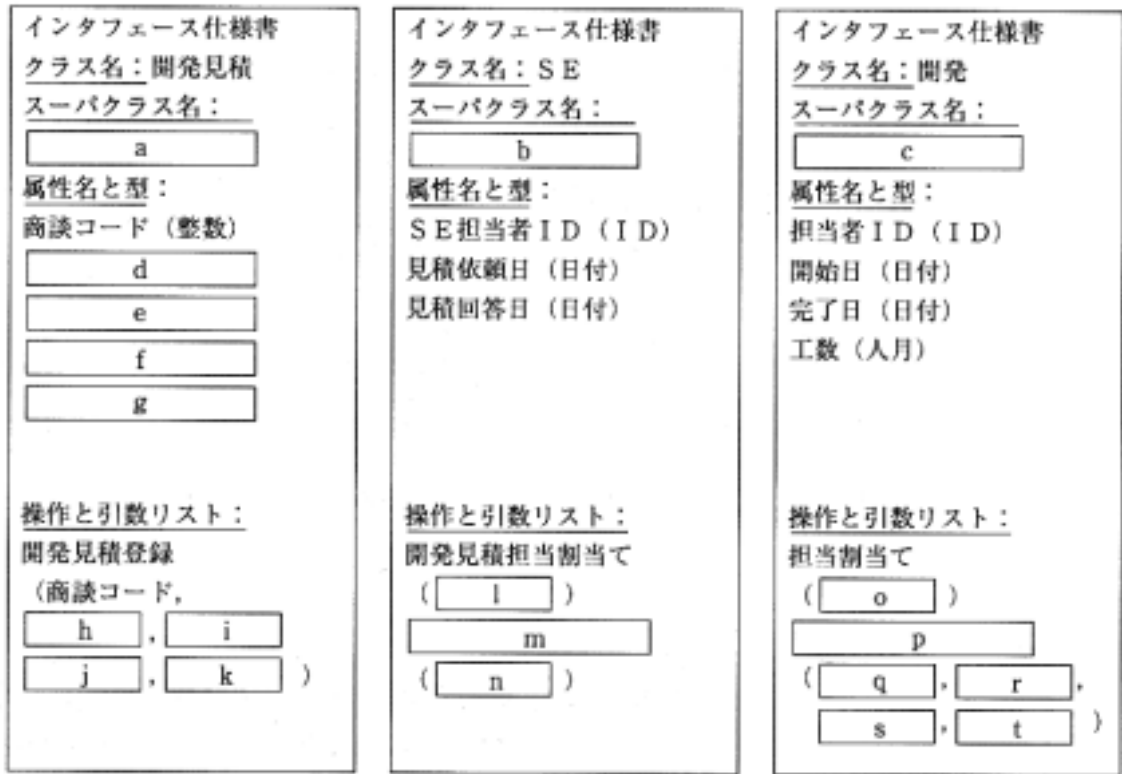


図 8 インタフェース仕様書

〔永続オブジェクト(データベース)の設計〕

図 7 のクラス図に示す製品クラスを、永続オブジェクトとしてデータベースで管理したい。 現行の関係データベースで管理する場合、製品クラスと開発クラスとの関係のような多対多の関係は、一般にレコードが冗長になる。 構造的な整合性と効率を考慮すると、多対多の関係にあるデータを、間に適切なデータを設けることによって、1対多の関係に分解しておく方が良くと判断した。 このような多対多の関係を分解するために、図 9 に示すように、中間に新たなクラスを導入した。

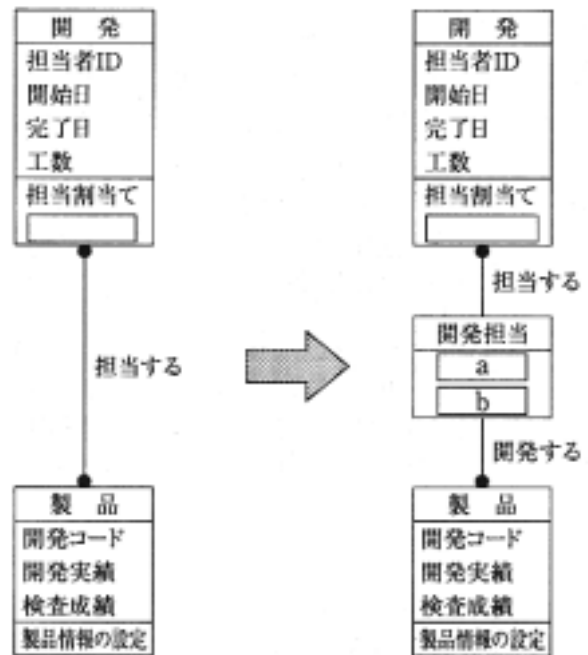


図 9 データベース設計

〔アーキテクチャ設計〕

図7のクラス図によってモデル化された業務管理システムを全社イントラネット上で実現するためには、クライアントサーバアーキテクチャをとるのが良いと判断した。このため、図のクラス群をクライアントとサーバに割り当て、ソフトウェアアーキテクチャの設計を行いたい。さらに、サーバは、各部のサーバと全社サーバを組み合わせ、データの適切な共有を図りたい。図7のクラス図を観察した結果、各部のサーバと全社サーバに配置するのに適した二つのサブシステムにクラス群を分割できることが分かった。

設問1 図2に示すユースケースの事象トレース図は一部だけ記述してある。記述にならなくて、破線の枠内に残りの事象トレースを補い、図を完成させよ。

設問2 図5の多重度と関連を考慮した初期クラス図の [a] ~ [l] に入れる適切な属性名又は操作(メソッド)名を答えよ。操作名は図1を参考に適切な名前を付けよ。

設問3 図7の洗練されたクラス図の [m] に入れる適切な属性名を答えよ。

設問4 図8のインタフェース仕様書の [a] ~ [t] に入れる適切な字句を答えよ。

ただし、属性名の順序や操作の引数の順序は問わない。

設問5 図9の [a] , [b] に入れる適切な属性名を答えよ。

設問6 図7のクラスを各部のサーバと全社サーバの二つのサブシステムに分割して配置する場合、どのように分割すべきか。各部のサーバと全社サーバに置く主たるクラスのクラス名をそれぞれ一つずつ挙げよ。さらに、そのように分割した理由を100字以内で述べよ。

OMT 法の表記法の説明

1. クラス



2. 関係





《解答》

[設問 1]



[設問 2]

- (a)顧客名      (b)顧客住所      (c)開発見積回答の作成      (d)開発報告
  - (e)開発名称      (f)見積金額      (g)開発規模見積      (h)開発工数見積
  - (i)開発費見積      (j)納入予定日      (k)入金日      (l)売上計上
- (a と b, e と f, g ~ j は順不同)

[設問 3]      (m)社員氏名

[設問 4]

- (a)管理情報      (b)社員      (c)社員
- (d)開発規模見積(ステップ)      (e)開発工数見積(人月)      (f)開発費見積(千円)
- (g)納入予定日(日付)      (h)開発規模見積      (i)開発工数見積
- (j)開発費見積      (k)納入予定日      (l)SE 担当者 ID
- (m)開発見積回答の作成      (n)商談コード      (o)担当者 ID
- (p)開発報告      (q)開発コード      (r)開始日
- (s)完了日      (t)工数      (e ~ g, h ~ k, q ~ t は順不同)

[設問 5]      (a)担当者 ID      (b)開発コード

[設問 6]

部内サーバ：営業

全社サーバ：商談

理由：各クラスを部間の共有有無によって分類し、営業クラスのように部内だけで使用する管理データに関するクラスは部内サーバに、商談のように全社で共有するプロダクトに関するクラスは全社サーバに配置する。

(解説)

[設問 1]

ユースケースとは、業務の流れの典型的なシナリオのことで、問題文中にあるように文章で記述したり、図 2 のような事象トレース図によって図示したりする。 図 2 では発注元が

ら営業部への見積依頼をトリガとして、営業部からは、SE 部に対して開発見積依頼が行われることなどが読み取れる。分析対象の業務管理システムについての内容は、図 1 の各部間の情報の流れとして整理されているだけなので、この内容をもとに、ユースケースの事象トレース図を作成することになる。図 1 では、単純に部門間の情報の流れだけが記述されているだけであるので、事象トレース図を作成するためには、各情報の流れを事象としてとらえ、各事象間の関連を時系列(トリガーアクション)といった観点で整理する必要がある。情報の名称から、常識的な範囲で時系列に整理すると以下のようになる。

見積依頼(顧客 営業部)  
 開発見積依頼(営業部 SE 部) (SE 部で開発の見積りを行う)  
 見積回答(営業部 顧客) (顧客は、見積内容を検討する)  
 発注(顧客 営業部)  
 発注確認(営業部 顧客)  
 開発指示(営業部 SE 部), 受注(営業部 経理部)  
 開発指示(SE 部 製造部) (製造部では、指示に従い開発する)  
 納入検査依頼(製造部 検査部) (検査部では、製品を検査する)  
 検査報告(検査部 製造部, 検査部 SE 部)  
 開発報告 製造部 SE 部)  
 出荷(SE 部 営業部)  
 納入・請求(営業部 顧客), 納入報告(営業部 経理部)  
 入金(顧客 経理部)

上記の内容の ~ の内容が波線の枠内に対応するので、これを事象トレース図としてまとめると、解答のようになる。

#### [設問 2]

オブジェクトとは、属性と操作(メソッド)を一体化したものであり、オブジェクト指向分析では、このオブジェクトの抽出が一番難しいとされている。幸い、図 5 には、すべてのオブジェクトが記述されており、属性やメソッドの空欄を埋めればよい。解答にあたっては、属性部分は、図 4「各部で管理しているデータ」が、メソッドの部分については、図 3「SE 部における開発見積の詳細シナリオ」がもとになる。また、各オブジェクトの空欄以外の内容を参考に名称などを決定する。

- (a), (b) : 顧客に関する属性であるので、図 4 の顧客データベースをもとにして、顧客名、顧客住所となることが分かる。
- (c) : メソッドの部分であるので、図 4 の内容をもとに解答する。SE 部の操作として、(c) 以外に「開発見積担当割当て」が記述されている。この他にメソッドに相当するものとしては、「見積データベースへの登録」、「類似開発の見積の抽出」、「開発見積回答の作成」などがあるが、「見積データベースへの登録」はクラス「開発見積」のメソッドであり「見積の抽出」などのアクションは、「開発見積回答の作成」のためのものであるので、「開発見積回答の作成」が解答となる。
- (d) : 「製造」のメソッドであるが、「検査」の内容がすべて記述されているので、この内容を参考に考えれば、「開発報告」であることが分かる。

- (e),(f) : 「商談」の属性であるので、「商談データベース」の内容が基になる。「商談データベース」の項目で、「商談」に記述されていないのは、開発名称,顧客 ID,営業担当 ID,営業担当者氏名,見積金額であるが,その商談の相手である顧客(顧客 ID)は、「顧客」と「商談」の関連である「要求する」によって表現されることになる。また、担当の営業に関する属性である営業担当 ID,営業担当者氏名については、「営業」と「商談」の関連である「担当する」によって表現される。したがって,開発名称,見積金額が解答となる。
- (g) ~ (j) : 「開発見積」の属性であるから、「見積データベース」がもととなる。上記(e, f)と同じ理由により,開発規模見積,開発工数見積,開発費見積,納入予定日が解答となる。
- (k) : 「売上」の属性であるから、「売上データベース」をもとにして,入金日であることが分かる。
- (l) : 「売上」の属性は、開発コード,入金日,入金額である。また,「経理」との関連が「売上を計上する」というものであるから,売上計上が解答となる。

[設問 3]

空欄 m が含まれる「社員」は、「営業」、「SE」、「開発」、「経理」のスーパークラスであり,サブクラス共通の属性とメソッドが定義されることになる。図 6 と図 7 を比較すると、営業担当者氏名, SE 担当者氏名,開発(検査)担当者氏名,経理担当者氏名といった属性が,各サブクラスからなくなっていることが分かる。営業, SE などを抽象化すると社員ということになり,各氏名は,役職(部門名)を抽象化して社員氏名ということになる。

[設問 4]

- (a) 図 7 によれば、「開発見積」のスーパークラスは「管理情報」管理情報が解答となる。
- (b), (c) 同様に、「SE」および「開発」のスーパークラスは「社員」である。したがって,(b),(c)ともに社員が解答となる。
- (d) ~ (g) 「開発見積」の属性であるから,設問 2 の(g) ~ (j)と同様に、開発規模見積,開発工数見積,開発費見積,納入予定日である。また,属性名と型とあるので,図 3 と空欄以外の記述内容を参考とすれば,開発規模見積(ステップ),開発工数見積(人月),開発費見積(千円),納入予定日(日付)が妥当であろう。
- (h) ~ (k) 「開発見積」の操作である開発見積登録の引数である。開発見積の登録とは、開発見積の属性に値を設定することであり,記述されている商談コード以外の開発規模見積,開発工数見積,開発費見積,納入予定日が解答となる。
- (l) 「SE」の操作である開発見積担当者割当てとは,属性である SE 担当者 ID に値を設定することであり,引数は SE 担当者 ID である。
- (m),(n) 「SE」のもう一つの操作は,設問 2 の(c)と同様に,開発見積回答の作成(m)である。開発見積回答の作成とは,営業部へ回答する開発規模などの値を算出することであり,その結果を開発見積として登録することになる。登録は「開発見積」の操作である開発見積登録によって行われるが,登録対象の開発のキーとなる商談コードが引数として必要となる。
- (o) 担当割当ての引数は,(l)と同様に担当者 ID である。

- (p) 「開発」のもう一つの操作は、設問 2 の(d)と同様に、開発報告である。
- (q) ~ (t) 開発報告とは、「開発」の属性である開始日、完了日、工数の各値の報告(引渡し)と、製品に対する実績(成績)の報告となる。製品の実績(成績)は、「製品」の属性であるため、そのキーとなる開発コードも引数として必要となる。

[設問 5]

多対多の関係にあるデータ(エンティティ)を、1対多の関係に分解するためには、エンティティ間の関係に属性を持たせて新たなエンティティとして定義する。このエンティティを連関エンティティと呼ぶ。したがって、空欄には、担当者 ID と開発コードが入ることになる。

[設問 6]

各部のサーバと全社サーバへの配置であるから、各クラスの使用範囲(部間共有の有無)によって分割されることになる。問題文の〔オブジェクト分析〕によれば、顧客データベースと見積データベースを全社にかかわるデータ、開発データベースと検査データベースを SE 部・製造部・検査部にかかわるデータ、営業データベースと売上データベースを営業部・経理部にかかわるデータと分類している。さらに、これらのデータのうち、業務の管理データは部内のデータで、業務の成果物として作成される見積りやソフトウェアに関するデータは、各部間で共有するデータであるとしている。また、〔クラス構造の洗練〕図 6 および図 7 の上段のクラス群は、業務プロセスに沿って業務を行う人のモデルであり、下段のクラス群は業務プロセスの入出力となるプロダクトのモデルであるとしている。さらに、プロダクトに関するクラス群は、各部で共有したい情報であるため、これらのクラス群を一貫して管理できるように、管理情報というスーパークラスを導入し、各部で共有できるように業務管理システムを構築することが望ましいと考えたとある。これらの記述をヒントとして、部内だけで使用するクラスと、部間で共有するクラスに分類すれば、営業、SE、製造、経理などのクラスが部門内で、商談、開発見積、製品、売上などのクラスが部間の共用クラスとなる。したがって、それぞれからクラスの例を挙げ、その理由としては、解答のように記述すればよい。

OMT 法(Object Modeling Technique) : ランボーが GE の同僚と開発した。