問5　共通ライブラリのオブジェクト指向設計に関する次の記述を読んで，設問 1，2 に答えよ。

システムインテグレータの T 社は，自社で開発するソフトウェアの品質の安定化，開発の生産性の向上などを目的として，オブジェクト指向を用いた共通ライブラリの設計に取り組んでいる。

これまで T 社では，システム開発において社員検索機能を数多く開発してきた。この機能は，組織情報から特定の社員を探すために使用される。図 1 に示すように，左側に階層表示されている組織の一つを選択すると，右側にその組織に属する社員及び下位の組織の情報が表示されるので，目的とする社員が見つかるまで下位の組織を選択して検索する。図 1 は，左側で金融システム部を選択し，右側に金融システム部に属する社員及び下位の組織を表示している例である。

T 社では，この組織の階層構造情報（以下，組織階層という）を，デスクトップアプリケーションソフトウェア，Web ページなどのユーザインタフェースに出力するシステムを開発してきた。

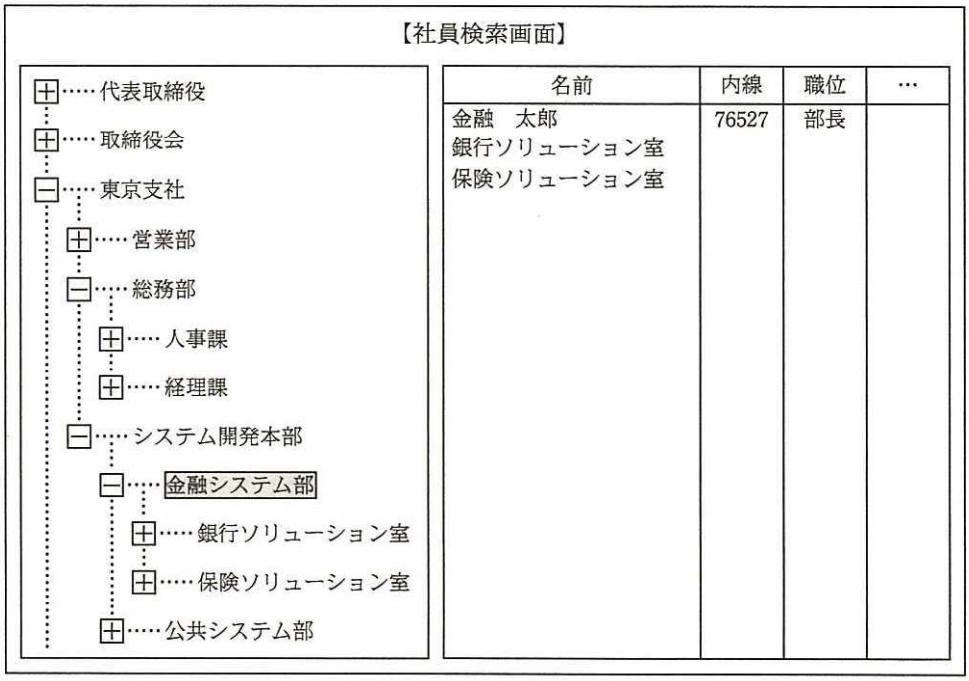


図 1　社員検索機能の表示例

組織階層の扱いは，どの開発プロジェクトにおいても類似するものが多い。そこで，組織階層を扱う機能をユーザインタフェースから切り離して，様々な開発プロジェクトで再利用できるように，共通ライブラリとして設計することにした。共通ライブラリは，オブジェクト指向で設計し，UML のクラス図で表現する。ここで，共通ライブラリの設計に当たり，共通ライブラリを利用する側のソフトウェアを一律にクライアントと呼ぶことにする。

まず，分析のためのクラス図を作成した。

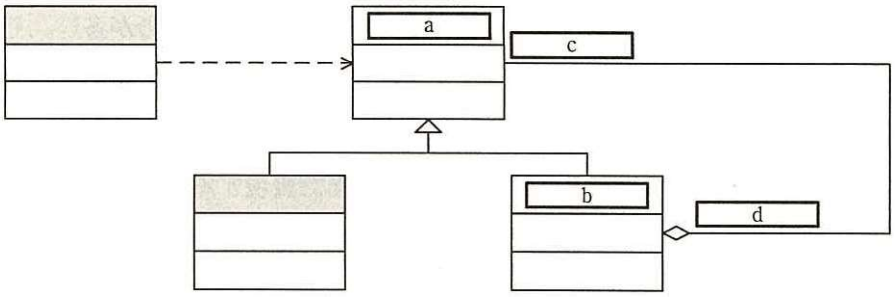
〔分析のためのクラス図の説明〕

組織階層において，ある組織が親の組織（以下，親組織という）をもつ場合，親組織は必ず一つであり，子の組織（以下，子組織という）をもつ場合，子組織の数に制限はない。また，親組織をもたない組織，子組織をもたない組織，親組織も子組織ももたない組織もある。社員は必ず一つの組織だけに属する。

(1) 組織階層は木構造で管理する。

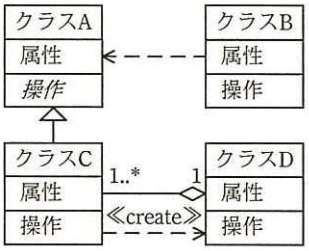
(2) クライアントが組織階層を参照する際，組織と社員を共通に扱えるようにするために，どちらも組織エントリとして管理する。

図 2 は，分析のためのクラス図である。



注記1　網掛けの部分は表示していない。
注記2　クラスに定義する属性，操作は記載していない。

（凡例）



長方形はクラスを表す。長方形の上段にはクラス名，中段には属性名の並び，下段には操作名の並びを記述する。属性名及び操作名は省略できる。

クラス名に≪interface≫が記載されている場合は，それがインタフェースであることを表す。

斜体の操作名は，それが抽象操作であることを表す。

クラス間を結ぶ線は関連を表す。

関連の端には，クラスのインスタンス間の多重度の範囲をx..yで表す。この凡例では，クラスDの1個のインスタンスが，クラスCの1個以上のインスタンスと関連することを示している。

関連の端の“◇”は集約を表し，凡例ではクラスCはクラスDの部分であることを示している。

“—▷”は汎化関係を表し，凡例では上位の一般的なクラスAと下位のより特殊的なクラスCとの間の分類関係を示している。

“--▷”は，インタフェースの実装を表す。

“-->”はクラス間の依存関係を表し，凡例では，クラスBがクラスAに依存することを示している。

依存関係に≪create≫が記載されている場合は，インスタンスを生成することを表し，凡例ではクラスCがクラスDのインスタンスを生成することを示している。

図2　分析のためのクラス図

次に，組織エントリを管理するクラスの，設計のためのクラス図を作成した。

〔設計のためのクラス図の説明〕

組織は自身に属する組織エントリのリスト（以下，組織エントリリストという）を，属性として保持する。

この組織エントリリストを設計するためのクラス図を図3に，組織エントリリストに定義する属性と操作の説明を表1に示す。以下，操作をメソッドという。

(1)　組織エントリリストは，組織エントリをサイズが可変の配列で管理する。

(2)　クライアントは，組織エントリリストに表1のメソッドでアクセス（組織エントリの追加や削除，取出し，走査など）する。

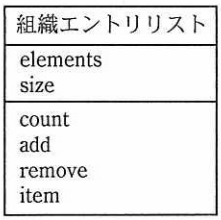


図３　組織エントリリストを設計するためのクラス図

表１　組織エントリリストに定義する属性とメソッドの説明

名前	説明
elements	組織エントリを格納するための配列。配列における要素の位置は要素番号で指定する。
size	elements の大きさ（配列の要素数）を表す。
count	組織エントリリストに登録されている組織エントリの数を返す。
add	組織エントリと要素番号を受け取り，受け取った要素番号の位置に組織エントリを挿入する。要素番号は省略可能であり，省略された場合は，組織エントリリストの最後に組織エントリを追加する。
remove	要素番号を受け取り，その要素番号の組織エントリを組織エントリリストから削除する。
item	要素番号を受け取り，その要素番号にある組織エントリを返す。

設問１　図 2 の分析のためのクラス図の中の［　　］に入れる正しい答えを，解答群の中から選べ。

a，bに関する解答群

ア　親組織　　イ　クライアント　　ウ　子組織　　エ　社員

オ　組織　　カ　組織エントリ　　キ　組織階層

c，dに関する解答群

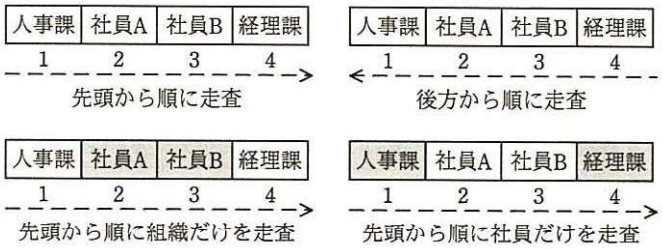
ア　0　　イ　0..1　　ウ　0..*　　エ　1　　オ　1..*

設問２　次の記述中の［　　］に入れる適切な答えを，解答群の中から選べ。

共通ライブラリの社内レビューを実施したところ，組織エントリリストのクラス設計に対して次の指摘が挙がった。

〔指摘の内容〕

組織エントリリストから組織エントリを取り出すとき，クライアントが要素番号を指定する必要がある。全ての組織エントリを取り出すときや，特定条件を満たす組織エントリだけを取り出したいときなど，クライアント側の実装依存が大きくなる。図4にクライアントの走査の例を示す。



注記　網掛けの部分は読み飛ばすことを意味している。

図4　クライアントの走査の例

共通ライブラリは，組織エントリリストの内部構造をクライアントに意識させず，組織エントリリストの走査を拡張できるように改善すべきである。

この指摘を受け，組織エントリリストのクラス設計を次の設計方針に従って見直すことにした。

〔設計方針〕

(1)　組織エントリリストは，より汎用的にするために，任意のオブジェクトを管理できるようにする。

(2)　リストを走査する処理は，別のクラス（以下，イテレータという）に与える。リストでは，このイテレータを生成して［　c　］をイテレータに登録することでイテレータがリストを走査できるようにする。

(3)　共通ライブラリで提供するイテレータは，先頭から順に全ての要素を走査するメソッドを提供する。

(4)　アクセス方法の統一を目的として，必要なメソッドをインタフェースとして定義する。クライアントは，必ずこのインタフェースを用いてリストを走査することとする。

(5)　クライアントは，共通ライブラリで提供するメソッドと異なる走査をした

い場合は，［ f ］する。

図5は見直し後の設計のためのクラス図，表2は図5のクラス図の説明である。

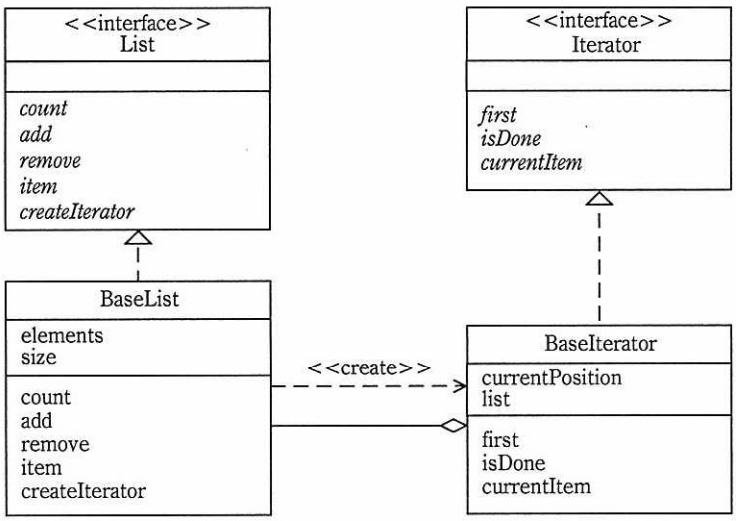


図5　見直し後の設計のためのクラス図

表2　図5のクラス図の説明

名前	説明
List	元の組織エントリリストに定義していたメソッド及びイテレータを生成するためのメソッド createIterator を定義する。
BaseList	元の組織エントリリストである。リストを表現するクラスに共通となる属性とメソッドを提供する。 createIterator は，リストを走査するイテレータを生成して，クライアントに返す。
Iterator	リストの要素へのアクセスや，リストを走査するためのメソッド first，isDone，currentItem を定義する。
BaseIterator	リストの先頭から順に全ての要素を走査する機能を提供する。走査対象のリスト list 及び走査中の位置 currentPosition を保持し，クライアントからの要求に応じて適切な要素を返す。 first は currentPosition を先頭に移動する。isDone はまだ走査対象が存在するかどうかをクライアントに返す。currentItem は currentPosition の位置にある要素をクライアントに返した後，currentPosition を走査方向に一つ移動する。

eに関する解答群

ア　クライアント　　イ　属性　　ウ　メソッド　　エ　リスト自身

fに関する解答群

ア　BaseList と BaseIterator を継承して処理をオーバライド

イ　BaseList と BaseIterator を継承してメソッドを追加

ウ　BaseList と BaseIterator を実装して処理をオーバライド

エ　BaseList と BaseIterator を実装してメソッドを追加

オ　List と Iterator を継承して処理をオーバライド

カ　List と Iterator を継承してメソッドを追加

キ　List と Iterator を実装して処理をオーバライド

ク　List と Iterator を実装してメソッドを追加