

社員メンテナンス（CUI_EM_OOP）

詳細設計書 — Day1～Day3

Java CLIアプリケーションの実装仕様・処理設計・テスト観点

対象 Main / EmployeeData / BaseData / DepartmentData 更新 2026-10-14

目次

- | | |
|----------------------|-----------------|
| 1. 概要 | 7. フローチャート（4種類） |
| 2. 前提知識 | 8. 疑似コード（言語非依存） |
| 3. 全体構成 | 9. テストケース一覧 |
| 4. 処理仕様（各クラスと主要メソッド） | 10. ユースケース図 |
| 5. 入力チェック仕様 | 11. 画面入出力例 |
| 6. データ項目定義 | |

1. 概要

本書は3日間のミッション課題を対象に、社員データの生成・一覧表示・ID検索、コンストラクタとカプセル化、継承・抽象クラス・ポリモーフィズム、部署名を用いた社員抽出までを定義する。社員5件、部署6件はプログラム内で生成し、配列で管理する。

対象は研修段階のCLIアプリケーション。コレクション、例外処理、ファイル読み書きはDay1～Day3の実装範囲外。Day3の部署名キーボード入力は任意の拡張機能として扱う。

2. 前提知識

Day	利用する主要要素	適用箇所
Day1	クラス・インスタンス・フィールド・メソッド、配列、for/while、Scanner	社員データの作成・一覧・ID検索ループ
Day2	private、getter/setter、コンストラクタ、@Override、static	データ保護、入力チェック、表示形式、生成件数
Day3	extends、abstract、親クラス型配列、instanceof、キャスト	共通型での管理、種別集計、部署名からの抽出

基礎確認ゾーンの Book・Item・Student 等は学習用であり、本体の社員検索処理には組み込まない。Day3 の基礎確認は独立した新規クラスで行う。

3. 全体構成

クラス	責務
Main	起動、社員・部署データの初期化、一覧、ID検索ループ、Day3挑戦処理の実行。
BaseData	社員と部署に共通の抽象親クラス。toString、toCSV、getPrimaryKeyValue を宣言。
EmployeeData	社員1名の6項目を保持。getter/setter、出力形式、主キー取得、静的生成件数を担当。
DepartmentData	部署ID・部署名を保持。BaseData の抽象メソッドを実装。
学習用クラス群	Day1・Day2のBook/Item/Student、Day3のShape等。社員メンテナンス本体とは独立。

関連：EmployeeData extends BaseData、DepartmentData extends BaseData。Main は BaseData[11] に両子クラスのインスタンスを格納する。

4. 処理仕様（各クラスと主要メソッド）

4.1 Main

- 社員5件を生成して employees 配列に格納し、必要に応じて一覧表示する。
- Day3は BaseData[] allData = new BaseData[11] に社員5件と部署6件を格納する。
- System.out.println(各要素) により各クラスの toString() を利用して一覧表示する。
- 検索IDを受け付け、入力が文字列 "0" なら終了。それ以外は社員IDを照合し、見つければ表示、見つからなければエラーを表示して繰り返す。
- Day3挑戦： instanceof で型別の件数を計算し、キャストでメールアドレスを取り出す。部署名「システム開発事業部」に対応する部署IDを検索した後、そのIDを持つ社員をすべて抽出する。

4.2 BaseData (Day3)

抽象メソッド	戻り値と契約
toString()	String：タブ区切りの表示用文字列
toCSV()	String：カンマ区切りの文字列
getPrimaryKeyValue()	String：社員IDまたは部署ID

4.3 EmployeeData

全6項目は private String。各項目にpublic getter/setterを設ける。引数なし・6引数の2種類のpublicコンストラクタを提供し、どちらも生成時に empCount を1増やす。6引数コンストラクタは必ずセッター経由で設定する。getEmpCount() は生成件数を返すstaticメソッド。

`toString()` は6項目をタブ区切りで返し、`toCSV()` はカンマ区切りで返し、`getPrimaryKeyValue()` は社員IDを返す。Day1の `isMatchId(String)` は検索用の初期実装であり、Day3は親クラス共通の主キー取得による照合に置き換える。

4.4 DepartmentData

`private String idDepartment, nmDepartment` を保持。各getter/setterを用意し、`toString()` で2項目をタブ区切り、`toCSV()` でカンマ区切り、`getPrimaryKeyValue()` で部署IDを返す。引数なしコンストラクタは自動生成のものを利用できる。

5. 入力チェック仕様

対象セッター	不正条件	不正時の処理
<code>setNmEmployee</code>	引数がnull、または前後の空白を除いた結果が空文字	「社員名が空のため設定できませんでした」と表示し、代入しない。
<code>setIdEmployee</code>	引数がnull、または前後の空白を除いた結果が空文字	「社員IDが空のため設定できませんでした」と表示し、代入しない。

チェックは必ず `null` 確認を先に行い、無効な値の場合はフィールドの既存値を変更しない。6引数コンストラクタも対応するセッターを呼び出すため同じ検査が適用される。その他4項目の入力チェックは本課題で指定されていない。

6. データ項目定義

6.1 EmployeeData

フィールド	型	内容
<code>idEmployee</code>	String	社員ID／主キー
<code>nmEmployee</code>	String	社員名
<code>knEmployee</code>	String	社員名カナ
<code>mailAddress</code>	String	メールアドレス
<code>password</code>	String	パスワード
<code>idDepartment</code>	String	所属部署ID

6.2 DepartmentData

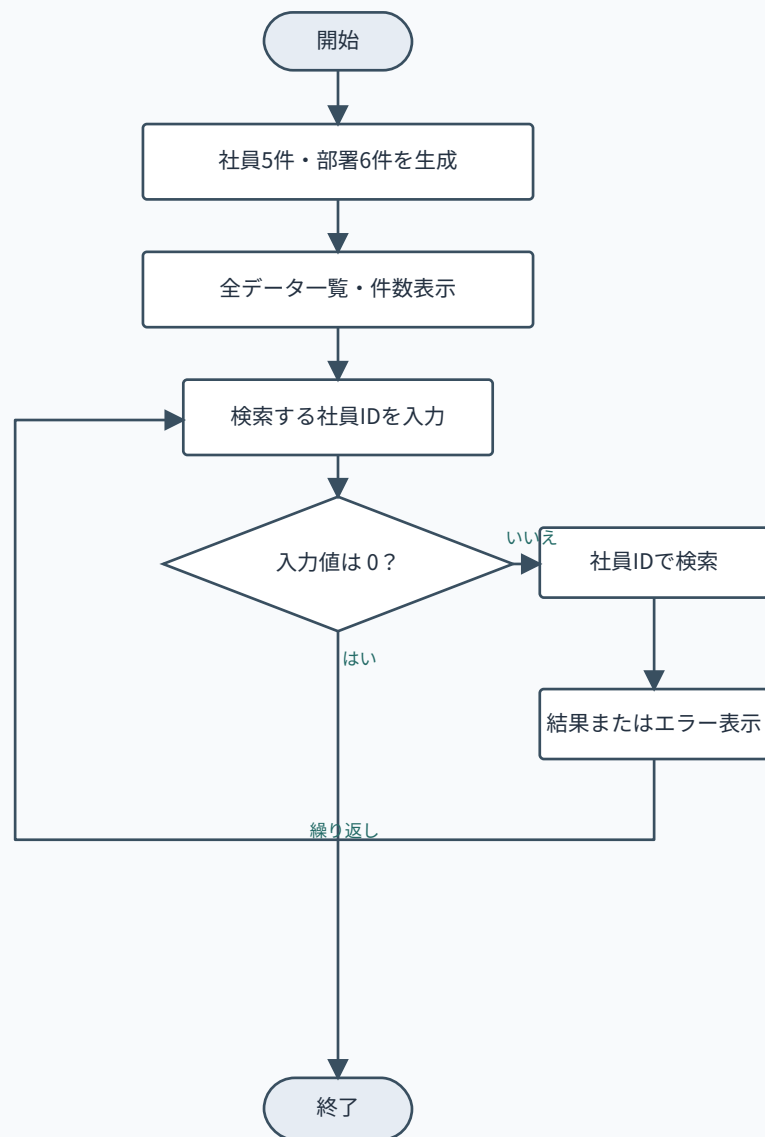
フィールド	型	内容
<code>idDepartment</code>	String	部署ID／主キー
<code>nmDepartment</code>	String	部署名

`empCount` : `private static int`、社員インスタンスの累積生成数。`allData` : 11要素、社員5件＋部署6件。

7. フローチャート

7.1 全体フローチャート（Day1～Day3統合）

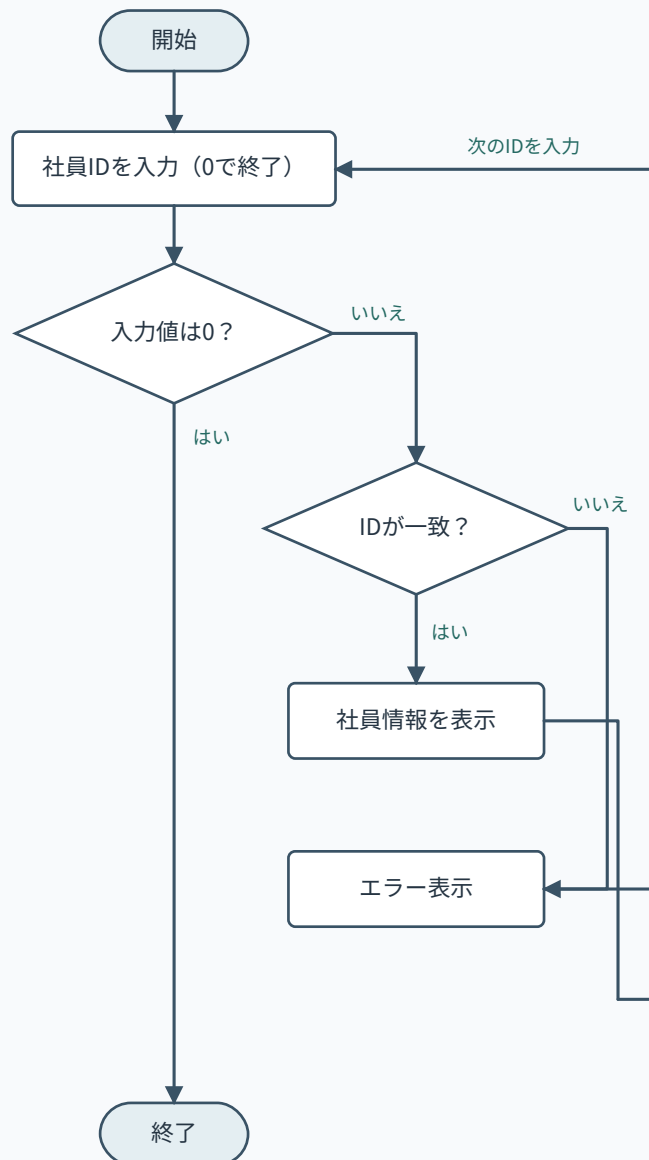
凡例：ひし形=条件判定 / 長方形=処理 / 楕円形=開始・終了



全体像：初期化 → 一覧表示 → 社員ID検索ループ → 0で終了。Day3の部署抽出は別フローで示す。

7.2 Day1のみ（ID検索ループ）

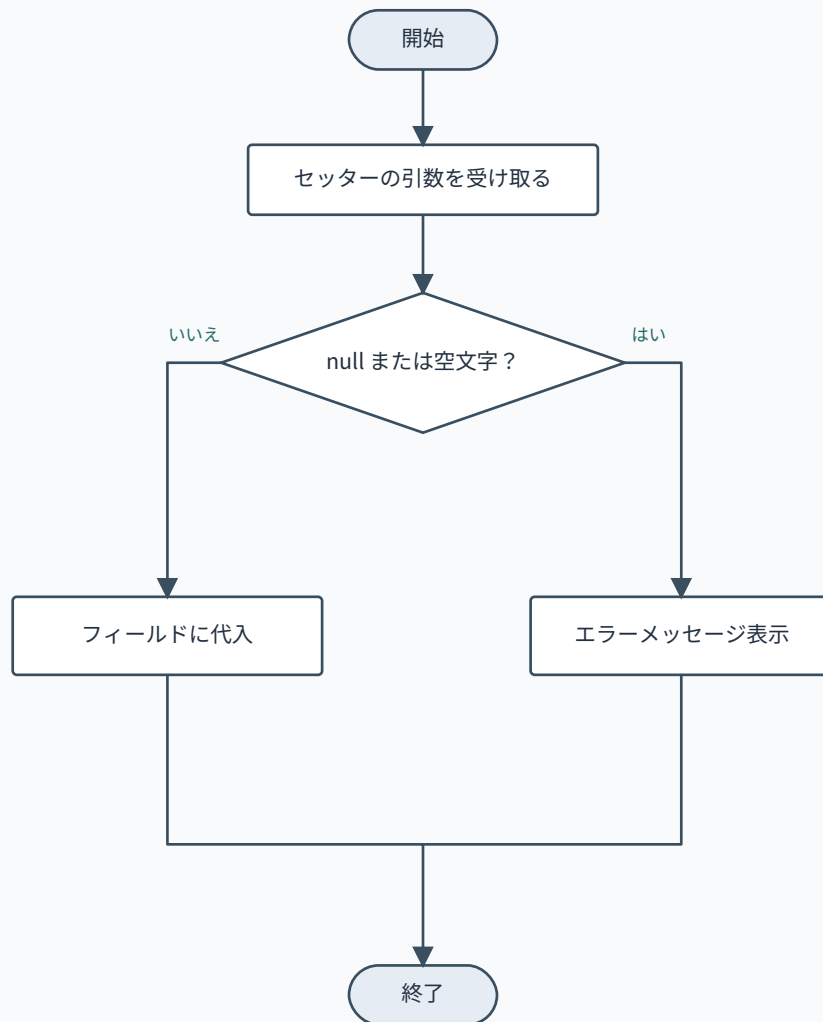
凡例：ひし形＝条件判定／長方形＝処理／楕円形＝開始・終了



0なら終了。0以外は右へ分岐して検索し、成功・失敗とも次の入力へ戻る。

7.3 Day2のみ（セッターの入力チェック）

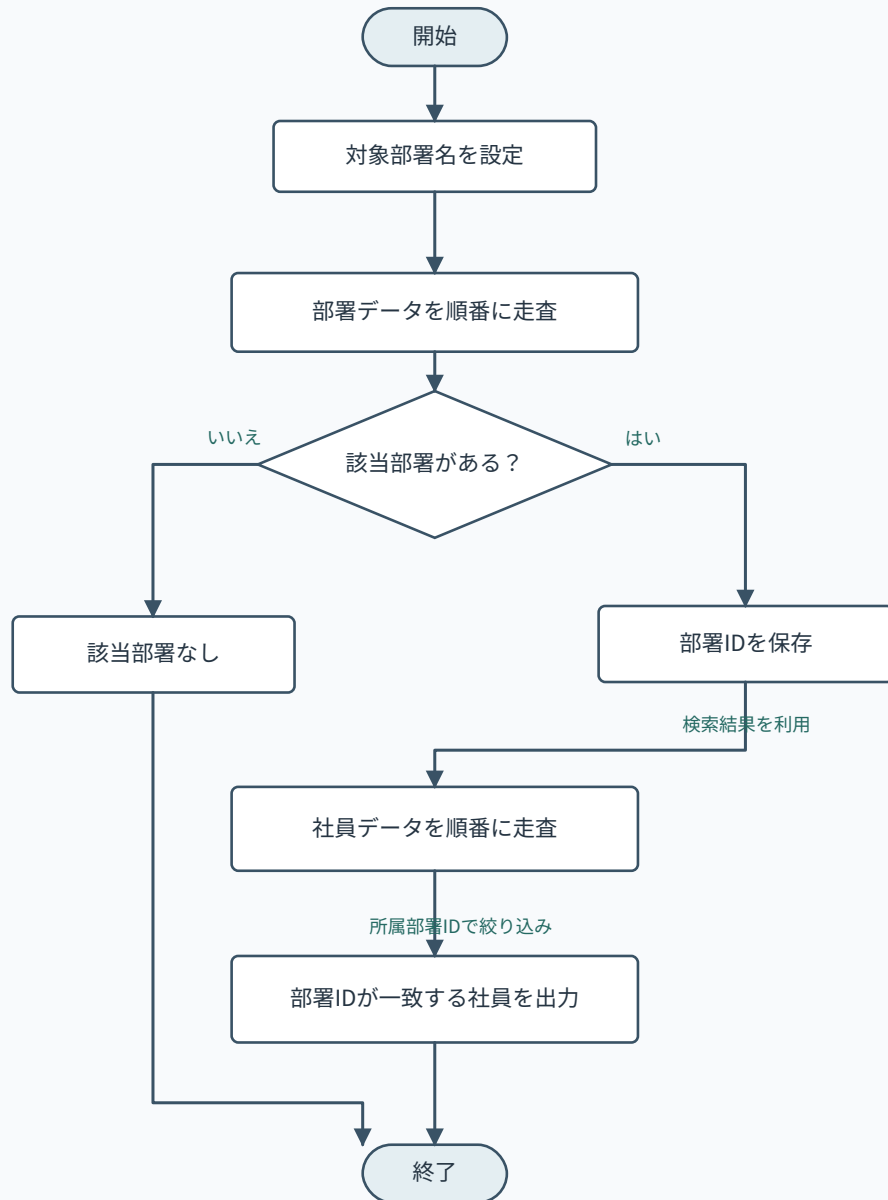
凡例：ひし形=条件判定／長方形=処理／楕円形=開始・終了



null判定を先に行い、無効値は変更せず終了する。

7.4 Day3のみ（部署名からの社員抽出口ジック）

凡例：ひし形＝条件判定／長方形＝処理／楕円形＝開始・終了



部署データを順番に調べて対象部署IDを確定し、社員データを走査して所属部署IDが一致する社員を抽出する。

8. 疑似コード（言語非依存）

```
社員5件を生成する
部署6件を生成する（Day3）
全11件を共通の配列に格納する（Day3）
各要素を表示する

繰り返す：
    「検索したい社員IDを入力してください（終了する場合は0）」を表示
    IDを読み取る
    もし ID が「0」なら：
        「システムを終了します」と表示し、繰り返し終了
    見つかった ← 偽
    社員データの各要素を順番に調べる：
        もし主キーが入力IDと一致すれば：
            社員情報を表示
            見つかった ← 真
            この走査を終了
    もし見つかった が 偽なら：
        「該当する社員が見つかりませんでした」を表示

【Day3 部署名から社員を抽出】
対象部署名 ← 「システム開発事業部」
対象部署ID ← 未設定
部署データを順番に調べる：
    もし部署名が対象部署名と一致すれば：
        対象部署ID ← その部署ID
        部署走査を終了
対象部署IDが見つかった場合：
    社員データを順番に調べる：
        もし社員の所属部署IDが対象部署IDに一致すれば：
            社員名とメールアドレスをタブ区切りで表示
```

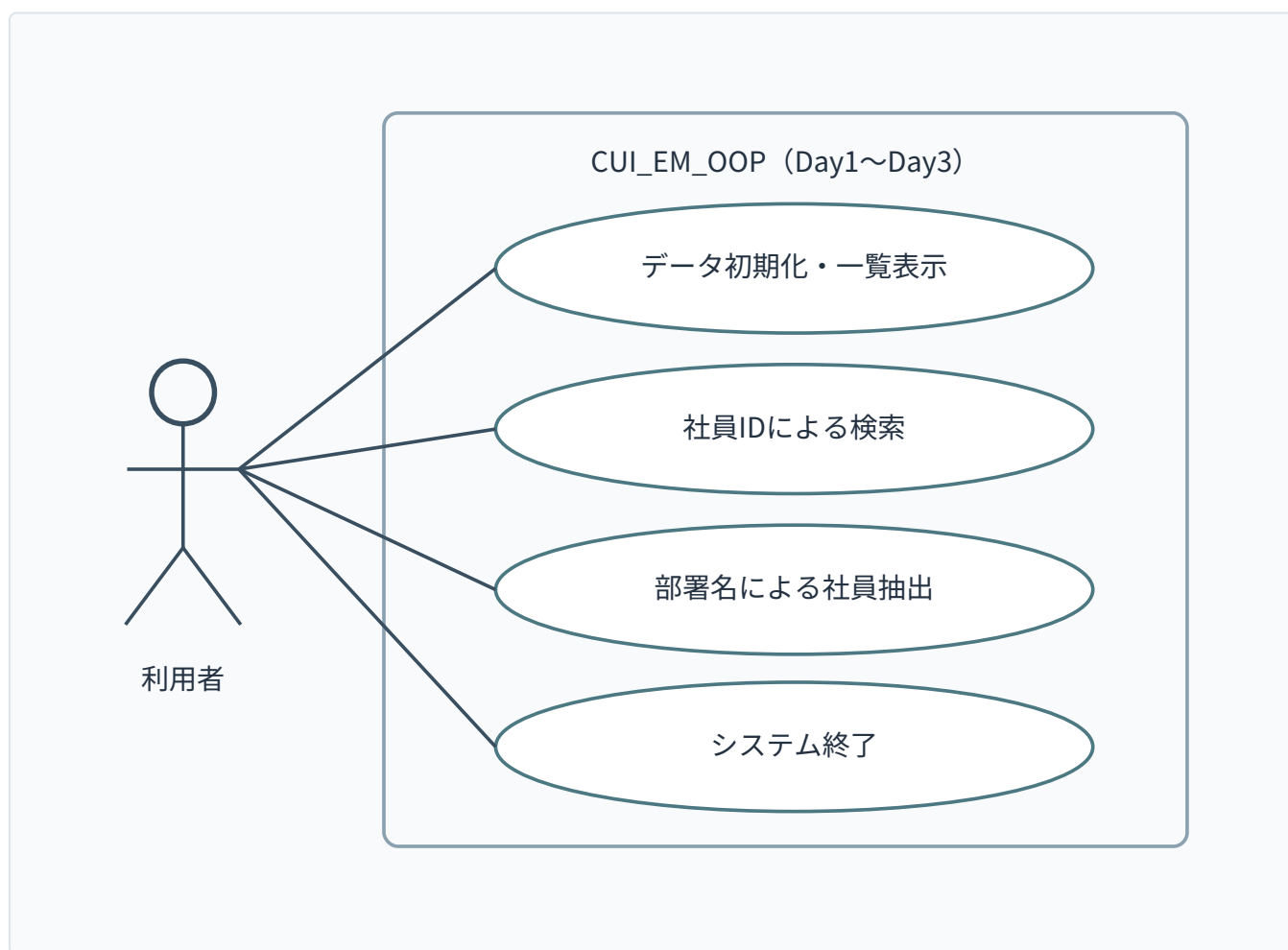
9. テストケース一覧

No.	操作・入力	期待結果
1	Day1 制作ゾーンを実行	社員5件をすべて一覧表示する。
2	社員ID「10001」	高橋準の行を表示し、再入力を受け付ける。
3	社員ID「99999」	「該当する社員が見つかりませんでした」。
4	社員ID「0」	「システムを終了します」を表示して終了。
5	社員名にnull、空文字、空白のみ	氏名のエラーを表示し代入しない。
6	社員IDにnull、空文字、空白のみ	IDのエラーを表示し代入しない。
7	6引数コンストラクタに空の氏名	セッターと同じエラーを表示し氏名がnullのまま。
8	社員を5件生成	

		生成件数5件。※既に生成した追加インスタンスがない状態。
9	Day3 一覧表示	社員5件+部署6件を表示する。
10	instanceofで件数を集計	社員5件、部署6件。
11	対象部署「システム開発事業部」	「三宅武」と「miya@hoge.jp」をタブ区切りで表示する。

10. ユースケース図

操作主体は利用者。初期化と一覧表示は起動時に自動実行し、社員ID検索・終了を標準機能、部署名からの抽出をDay3の挑戦課題として位置づける。



Day3の部署名をキーボード入力可能にする機能は任意課題の拡張とする。

11. 画面入出力例

表示上の区切りはタブ文字を用いる。以下はそれぞれのゾーンの完成形を独立して実行したときのイメージであり、すべての出力が1回の起動で連続表示されるという意味ではない。

11.1 Day1 — CUI_EM_OOP制作ゾーンの完成形

```
10000  山田龍也ヤマダタツヤ  yamada@hoge.jp  aaa    10
10001  高橋準   タカハシジュン  jtaka@hoge.jp   bbb    10
10002  麻枝史明マエダフミアキ  fmak@hoge.jp    ccc    11
10003  丸戸章介マルトショウスケ marusyo@hoge.jp ddd    12
10004  三宅武   ミヤケタケシ   miya@hoge.jp    eee    13
```

11.2 Day1 — 挑戦問題ゾーンの完成形

```
検索したい社員IDを入力してください（終了する場合は0）
10001
10001  高橋準   タカハシジュン  jtaka@hoge.jp   bbb    10
検索したい社員IDを入力してください（終了する場合は0）
99999
該当する社員が見つかりませんでした
検索したい社員IDを入力してください（終了する場合は0）
0
システムを終了します
```

11.3 Day2 — CUI_EM_OOP制作ゾーンの完成形

```
10000  山田龍也ヤマダタツヤ  yamada@hoge.jp  aaa    10
10001  高橋準   タカハシジュン  jtaka@hoge.jp   bbb    10
10002  麻枝史明マエダフミアキ  fmak@hoge.jp    ccc    11
10003  丸戸章介マルトショウスケ marusyo@hoge.jp ddd    12
10004  三宅武   ミヤケタケシ   miya@hoge.jp    eee    13
```

Day1と一覧の見た目は同じ。内部処理がコンストラクタ・セッター・`toString` 中心に変わる。

11.4 Day2 — 挑戦問題ゾーンの完成形（空文字エラー含む）

```
現在の社員データ件数:5件
社員名が空のため設定できませんでした
社員IDが空のため設定できませんでした
```

確認用に不正値を渡した場合の出力例。生成件数は起動中に生成した社員インスタンスの数で変動する。

11.5 Day3 — CUI_EM_OOP制作ゾーンの完成形

```
10000  山田龍也ヤマダタツヤ  yamada@hoge.jp  aaa    10
10001  高橋準   タカハシジュン  jtaka@hoge.jp   bbb    10
10002  麻枝史明マエダフミアキ  fmak@hoge.jp    ccc    11
10003  丸戸章介マルトショウスケ marusyo@hoge.jp ddd    12
```

```
10004  三宅武  ミヤケタケシ  miya@hoge.jp  eee  13
10      役員
11      人事・総務部
12      経理部
13      システム開発事業部
14      教育事業部
15      営業企画部
```

11.6 Day3 — 挑戦問題ゾーンの完成形（部署名検索）

```
社員データ:5件
部署データ:6件
yamada@hoge.jp
jtaka@hoge.jp
fmak@hoge.jp
marusyo@hoge.jp
miya@hoge.jp
三宅武  miya@hoge.jp
```

部署名「システム開発事業部」→ 部署ID「13」→ 所属社員「三宅武」。CSV入出力や検索用メニューの追加は本範囲外。