

2024年7月17日



# 【COBOL × Google Cloud】 COBOL × AIハッカソン2024 Day2 COBOL課題への生成AI適用

  
System Sustainability VietNam

# チームについて

LE DINH TUNG

---

通訳者

TRAN DUC MANH

---

SSV COBOL Program Analyst Lead

NGUYEN DAI LOC

---

SSV COBOL Program Analyst

NGUYEN KIM THUAN

---

SSV COBOL Program Analyst

# コンテンツ

01 FE試験の解答

02 コードリファクタリング

03 設計からコード生成

04 コードから詳細設計生成

05 コードからテストケース生成

06 まとめ

## ① FE試験問題解答 - 手順

現時点ではPPTXファイルをサポートしていないため、事前にPDFに変換する必要があります。



平成27年度春期試験 基本情報技術者試験 問10 COBOL.pdf

平成27年度春期試験  
基本情報技術者試験  
問10 COBOL

問10 次のCOBOLプログラムの説明及びプログラムを読んで、設問1, 2に答えよ。

〔プログラムの説明〕

C社では、全従業員の出勤時刻と退社時刻を入口のゲートで自動的に記録し、出勤記録ファイルに格納している。このプログラムは、1か月分の出勤記録ファイルを読み込み、出勤した従業員の勤務管理ファイルに書き込むものである。

(1) 出勤記録ファイルは、図1に示すレコード様式の順ファイルで、ある月の1か月分の出勤記録が格納されている。

日付	時刻	出勤フラグ	従業員番号
8桁	4桁	1桁	6桁

図1 出勤記録ファイルのレコード様式

- ① レコードは日付、時刻、出勤フラグの昇順に格納されている。
- ② 日付には、年、月、日が、それぞれ4桁、2桁、2桁の西暦で格納されている。
- ③ 時刻には、出勤又は退社の時刻の時、分が、それぞれ2桁の24時間表記で格納されている。
- ④ 出勤フラグには、出勤時は0が、退社時は1が格納されている。
- ⑤ 従業員番号には、従業員ごとに一意な6桁の番号が格納されている。
- ⑥ 同一従業員の出勤時の記録と退社時の記録は、必ず対になっている。同じ日付のレコードは1組だけである。
- ⑦ 出勤時刻は6時から22時までである。退社時刻は同じ日付の24時までである。

(2) 勤務管理ファイルは、図2に示すレコード様式の従業員番号をキーとする索引ファイルである。

従業員番号 6桁	勤務履歴1		...	勤務履歴31		勤務時間 5桁
	出勤時刻 4桁	退社時刻 4桁		出勤時刻 4桁	退社時刻 4桁	

31回繰返し

図2 勤務管理ファイルのレコード様式

- ① 勤務履歴1～31には、当該月の1～31日の出勤時刻及び退社時刻が格納される。出勤しなかった日と暦にない日の出勤時刻及び退社時刻には、9999が格納される。
- ② 出勤時刻及び退社時刻には、時、分が、それぞれ2桁の24時間表記で格納される。
- ③ 勤務時間には、1か月の勤務時間の合計が、3桁の時間と2桁の分で格納される。毎日の出勤時刻から退社時刻までの時間が1日の勤務時間である。途中の休み時間も勤務時間に含める。

〔プログラム〕

(行番号)

```

1 DATA DIVISION.
2 FILE SECTION.
3 FD LOG-FILE.
4 01 LOG-REC.
5     02 LOG-DATE    PIC 9(8).
6     02 LOG-TIME    PIC 9(4).
7     02 LOG-FLAG    PIC 9(1).
8     88 LOG-IN      VALUE 0.
9     88 LOG-OUT     VALUE 1.
10    02 LOG-ENUM     PIC X(6).
11 FD ATD-FILE.
12 01 ATD-REC.
13    02 ATD-ENUM     PIC X(6).
14    02 ATD-HST.
15    03 ATD-ELM     OCCURS 31.
16        04 ATD-IN  PIC 9(4).
17        04 ATD-OUT PIC 9(4).
18    02 ATD-SUM     PIC 9(5).

```

```

19 WORKING-STORAGE SECTION.
20 77 KEY-FLAG      PIC 9(1).
21   88 ATD-VALID   VALUE 0.
22   88 ATD-INVALID VALUE 1.
23 77 EOF-FLAG     PIC 9(1) VALUE 0.
24   88 LOG-EOF     VALUE 1.
25 01 W-DATE.
26   02 FILLER      [ a ].
27   02 IDX         PIC 9(2).
28 01 W-TIME-FROM.
29   02 HF          PIC 9(2).
30   02 MF          PIC 9(2).
31 01 W-TIME-TO.
32   02 HT          PIC 9(2).
33   02 MT          PIC 9(2).
34 01 W-TIME-ACCUM.
35   02 HA          [ b ].
36   02 MA          PIC 9(2).
37 77 W-MINUTE     PIC 9(5).
38 PROCEDURE DIVISION.
39 MAIN-PROC.
40   OPEN INPUT LOG-FILE
41     I-O ATD-FILE.
42   PERFORM READ-PROC.
43   CLOSE ATD-FILE LOG-FILE.
44   STOP RUN.
45 READ-PROC.
46   PERFORM UNTIL LOG-EOF
47     READ LOG-FILE
48     AT END
49       SET LOG-EOF TO TRUE
50     NOT AT END
51       PERFORM UPDATE-PROC
52     END-READ
53   END-PERFORM.
54 UPDATE-PROC.
55   MOVE LOG-ENUM TO ATD-ENUM.
56   SET ATD-VALID TO TRUE.
57   READ ATD-FILE
58     INVALID KEY
59       SET ATD-INVALID TO TRUE
60     MOVE LOG-ENUM TO ATD-ENUM
61     [ c ]
62     MOVE ZERO TO ATD-SUM
63   END-READ.

```

```

64   MOVE LOG-DATE TO W-DATE.
65   EVALUATE TRUE
66     WHEN LOG-IN
67       MOVE LOG-TIME TO ATD-IN(IDX)
68     WHEN LOG-OUT
69       MOVE LOG-TIME TO ATD-OUT(IDX) W-TIME-TO
70       MOVE ATD-IN(IDX) TO W-TIME-FROM
71       MOVE ATD-SUM TO W-TIME-ACCUM
72     PERFORM CALC-PROC
73     MOVE [ d ] TO ATD-SUM
74   END-EVALUATE.
75   IF ATD-VALID THEN
76     REWRITE ATD-REC
77   ELSE
78     WRITE ATD-REC
79   END-IF.
80 CALC-PROC.
81   COMPUTE W-MINUTE = (HT - HF + HA) * 60 + MT - MF + MA.
82   DIVIDE W-MINUTE BY 60 GIVING HA REMAINDER MA.

```

設問1 プログラム中の [ ] 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 PIC 9(2)           イ PIC 9(3)           ウ PIC 9(4)  
エ PIC 9(5)           オ PIC 9(6)

cに関する解答群

ア INITIALIZE ATD-HST  
イ INITIALIZE ATD-HST REPLACING NUMERIC BY 9999  
ウ MOVE 9999 TO ATD-HST  
エ MOVE ALL ZERO TO ATD-HST

dに関する解答群

ア ATD-OUT(IDX)   イ HA           ウ W-MINUTE   エ W-TIME-ACCUM

設問2 C社では22時以降の勤務を深夜手当の対象としている。勤務管理ファイルのレコードに深夜勤務時間の領域を追加し、1か月の深夜勤務時間の合計も集計するようにプログラムを変更する。表1中の□に入れる正しい答えを、解答群の中から選べ。ここで、表1中の□dには設問1の正しい答えが入っているものとする。

表1 プログラムの変更内容

処置	変更内容
行番号18と19の間に追加	02 ATD-NIGHTSUM PIC 9(5).
行番号19と20の間に追加	77 NIGHT-TIME PIC 9(4) VALUE 2200.
行番号62を変更	MOVE ZERO TO ATD-SUM ATD-NIGHTSUM
行番号73と74の間に追加	IF ATD-OUT(IDX) > NIGHT-TIME THEN MOVE □e TO W-TIME-FROM MOVE □f TO W-TIME-ACCUM PERFORM CALC-PROC MOVE □d TO ATD-NIGHTSUM END-IF

e, fに関する解答群

- |               |                |                |
|---------------|----------------|----------------|
| ア ATD-IN(IDX) | イ ATD-NIGHTSUM | ウ ATD-OUT(IDX) |
| エ ATD-SUM     | オ NIGHT-TIME   |                |

# ① FE試験問題解答 - 手順

「マルチターン プロンプトの設計（チャット）」を選択



マルチターン プロンプトの設計（チャット）

モデルと継続的に対話し、回答を生成する際に対話の履歴をコンテキストとして使用します。

オープン

# ① FE試験問題解答 - 手順

質問とリクエストに関する  
ファイルを送信する

The screenshot shows the Consortium AI interface. At the top, there is a navigation bar with a back arrow, the name "Consortium", and several utility icons: a clock for "履歴" (History), a document icon for "NOTES", a trash can for "CLEAR CONVERSATION", and a link icon for "API リファレンス" (API Reference). Below the navigation bar is a text input field containing the system instruction "システム指示" with a help icon and an "編集" (Edit) button. The main area of the interface is titled "会話を開始する" (Start Conversation) and contains a message: "モデルの応答がここに表示されます。チャットを開始するには、以下にプロンプトを入力してください。" (The model's response will be displayed here. To start the chat, enter a prompt below). Below this message are three prompt cards: "Newsletter design" (A chatbot to help create engaging newsletters), "Video Ad script writer" (Write social media style video ad scripts), and "Company chatbot" (Create a chatbot for customers with basic company information). At the bottom of the prompt gallery is a "BROWSE PROMPT GALLERY" link. In the bottom right corner, it shows "1,037 tokens" and a "VIEW TOKENS" link. At the very bottom is a chat input area with a file upload icon, a document icon, a text input field containing "平成27年度春期試験 基..." and "答えを教えてください", and a send button.

# ① FE試験問題解答 - 手順

## 得られた結果

ユーザー

 平成27年度春期試験 基本情報

答えを教えてください

AI

設問1 a. ウ PIC 9(4) b. ア ATD-OUT(IDX) c. イ INITIALIZE ATD-HST REPLACING NUMERIC BY 9999 d. ウ W-MINUTE

設問2 e. イ ATD-NIGHTSUM f. イ ATD-NIGHTSUM

## ② コードリファクタリング

サンプルは、弊社メンバー作成のサンプルコードです。

コードの中で多くの不要なGOTO文が使用されています。

ワーキングストレージ変数がWS-で始まっていません。

## ② コードリファクタリング

例

```
PERFORM UNTIL OPTION = 0
  EVALUATE OPTION
    WHEN 1
      PERFORM INPUT-DATA
      GO TO MAIN
    WHEN 2
      PERFORM OUTPUT-DATA
      GO TO MAIN
    WHEN 3
      PERFORM SORT-DATA
      GO TO MAIN
    WHEN 4
      PERFORM SORT-DATA-MATH
      GO TO MAIN
    WHEN 5
      PERFORM SEARCH-DATA
      GO TO MAIN
    WHEN 6
      PERFORM SEARCH-DATA-MATH
      GO TO MAIN
    WHEN 0
      GO TO MAIN
    WHEN OTHER
      DISPLAY "INPUT FALSE! INPUT PLEASE"
      GO TO MAIN
  END-EVALUATE
END-PERFORM.
```

## ② コードリファクタリング

コードとリクエストを送信する

ユーザー

 sample.txt

コードを改善する (GO TO、作業領域変数は WS によって開始される必要があります)

## ② コードリファクタリング

AIが編集したコード

```
PERFORM UNTIL WS-OPTION = 0
  EVALUATE WS-OPTION
    WHEN 1
      PERFORM WS-INPUT-DATA
    WHEN 2
      PERFORM WS-OUTPUT-DATA
    WHEN 3
      PERFORM WS-SORT-DATA
    WHEN 4
      PERFORM WS-SORT-DATA-MATH
    WHEN 5
      PERFORM WS-SEARCH-DATA
    WHEN 6
      PERFORM WS-SEARCH-DATA-MATH
    WHEN 0
      GO TO WS-MAIN-EXIT
    WHEN OTHER
      DISPLAY "INPUT FALSE! INPUT PLEASE"
  END-EVALUATE
END-PERFORM.
```

### ③ 設計からコード生成

これは弊社メンバーが作成した設計書サンプルです。

ファイルを送信し、AIにコード作成を依頼する。

ユーザー

 02 - Bootcamp 1 - Create Client

この設計のCOBOLコードを作成する

### ③ 設計からコード生成

生成されたコードをコンパイルしてみたところ、いくつかの小さな間違いがあり、修正が必要です。

```
sample.cbl:128: error: continuation character expected
126 |         ACCEPT WS-DATE-OF-BIRTH.
127 |         DISPLAY "Enter marriage status (M for Married, S for Single):..
128 >        ACCEPT WS-MARRIAGE-STATUS.
129 |         DISPLAY "Enter address: " WITH NO ADVANCING
130 |         ACCEPT WS-ADDRESS.
sample.cbl:520: warning: line not terminated by a newline [-Wmissing-newline]
518 |         WHEN OTHER
519 |             DISPLAY "Invalid choice. Please try again."
520 >        END-EVALUATE.<EOF>
sample.cbl: in paragraph 'WS-CREATE-PERSONAL-CLIENT':
sample.cbl:123: error: syntax error, unexpected NO
121 |         DISPLAY "Enter given name: " WITH NO ADVANCING
122 |         ACCEPT WS-GIVEN-NAME.
123 >        DISPLAY "Enter gender (M for Male, F for Female): " WITH NO AD..
124 |         ACCEPT WS-GENDER.
125 |         DISPLAY "Enter date of birth (YYYYMMDD): " WITH NO ADVANCING
sample.cbl:127: error: 'Enter' is not defined
125 |         DISPLAY "Enter date of birth (YYYYMMDD): " WITH NO ADVANCING
126 |         ACCEPT WS-DATE-OF-BIRTH.
127 >        DISPLAY "Enter marriage status (M for Married, S for Single):..
128 |         ACCEPT WS-MARRIAGE-STATUS.
129 |         DISPLAY "Enter address: " WITH NO ADVANCING
sample.cbl:127: error: syntax error, unexpected :, expecting ENTRY or PROGRAM or Identifier
sample.cbl:127: error: invalid symbol '?' - skipping word
sample.cbl:127: error: invalid symbol '!' - skipping word
sample.cbl:130: error: syntax error, unexpected WHEN
```

### ③ 設計からコード生成

#### 修正後のテスト実行結果

```
*****  
* CLIENT MANAGEMENT *  
*****  
A - Create new client  
B - Modify client information  
C - View client information  
D - Delete client profile  
E - Exit  
Enter your choice: 
```

### ③ 設計からコード生成

クライアントを作成してみる

```
*****  
* CLIENT MANAGEMENT *  
*****  
A - Create new client  
B - Modify client information  
C - View client information  
D - Delete client profile  
E - Exit  
Enter your choice: A
```

### ③ 設計からコード生成

クライアント作成が成功した

```
*****  
* CLIENT INFORMATION *  
*****  
Client number: 11111111  
ID: 215297247  
Client type: P  
Surname: MANH  
Given name: TRAN  
Gender: M  
Date of birth: 19940315  
Marriage status: S  
Address: manhtd@ssv-corp.com  
  
Postal code: 740000  
Area code: 084  
Phone: 00977632570  
Tax code: 2222221112  
Created date: 00000000  
Do you want to create this profile? (Y/N): Y  
Client created successfully!
```

### ③ 設計からコード生成

クライアントの情報を探してみる

```
*****  
* CLIENT MANAGEMENT *  
*****  
A - Create new client  
B - Modify client information  
C - View client information  
D - Delete client profile  
E - Exit  
Enter your choice: C  
*****  
* VIEW CLIENT INFORMATION *  
*****  
Enter client number: 1111111█
```

### ③ 設計からコード生成

ほとんどが正しい

```
*****  
* PERSONAL CLIENT INFORMATION *  
*****  
Client number: 11111111  
ID: 215297247  
Client type: P  
Name: MANH  
Gender: M  
Date of birth: 19940315  
Marriage status: S  
Address: manhtd@ssv-corp.com  
Postal code: 740000  
Area code: 084  
Phone: 00977632570  
Tax code: 2222221112  
Created date: 00000000  
Press any key to continue...█
```

## ④ コードから詳細設計生成

修正したコードとAIへのリクエストを送信する。

ユーザー

 sample2.txt

詳細設計を生成する

## ④ コードから詳細設計生成

### AIに詳細設計を生成してもらった

**AI** 申し訳ありませんが、私はまだ COBOL コードを生成できません。しかし、私はあなたがこれに基づいて詳細な設計を作成するのに役立ちます。

詳細な設計

#### 1. モジュール設計

- メインモジュール：プログラムのエントリポイント。メニューを表示し、ユーザー入力に基づいて適切なモジュールを呼び出します。
- メニュー表示モジュール：プログラムのオプションメニューを表示し、ユーザーからの選択を受け入れます。
- クライアント作成モジュール：新しいクライアントの情報を取得し、データベースに保存します。
- クライアント情報更新モジュール：既存のクライアントの情報を取得し、ユーザーからの更新を受け入れてデータベースを更新します。
- クライアント情報表示モジュール：クライアントの情報をデータベースから取得して表示します。
- クライアント削除モジュール：クライアントの情報をデータベースから削除します。

#### 2. データ構造

## ④ コードから詳細設計生成

Markdown を取得するために プロンプトの設計（シングルターン）を使用する



### プロンプトの設計（シングルターン）

プロンプトを通じてモデルに指示し、対話の履歴を使用せずに回答を得ます。

オープン

## ④ コードから詳細設計生成

同様にリクエストを送信する

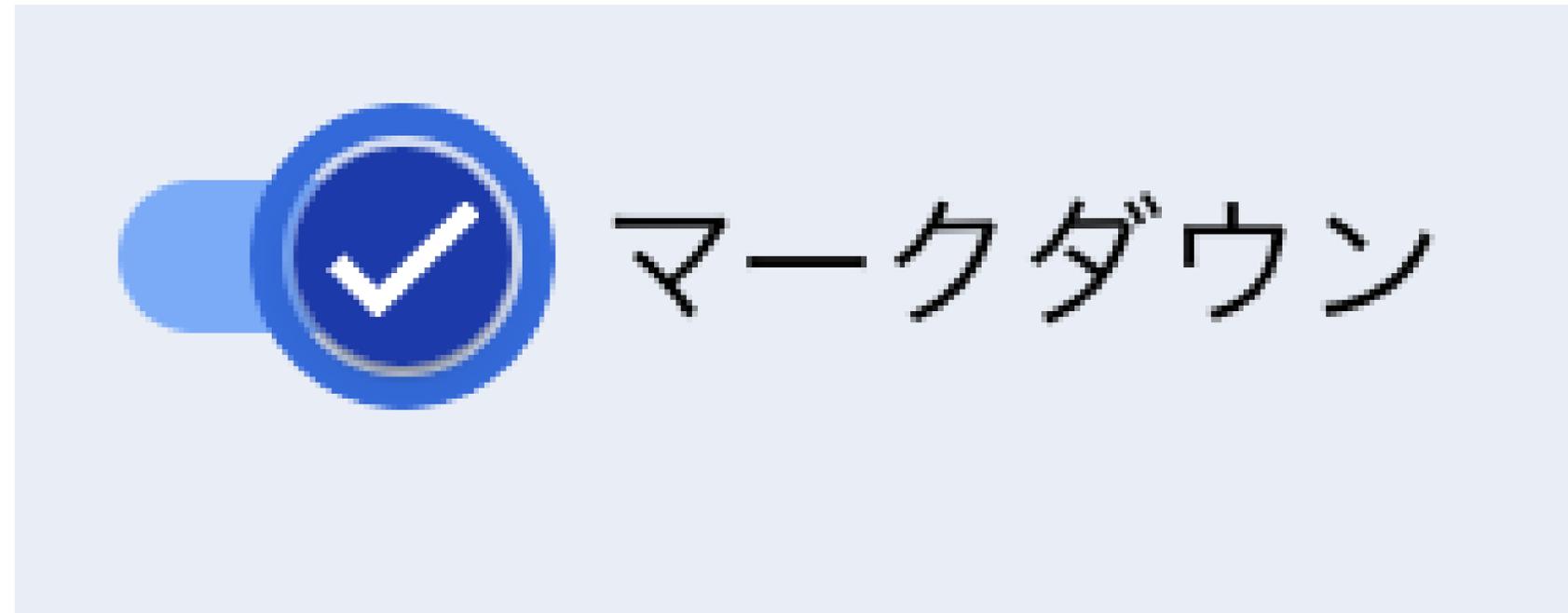


sample2.txt

詳細設計を生成する

## ④ コードから詳細設計生成

詳細設計が生成された後、以下のように選択する



## ④ コードから詳細設計生成

Markdown をファイルにコピーして、  
詳細設計を取得することが出来る

### 詳細設計

プログラム名: STUDENT-MANAGEMENT

目的: 学生情報を管理する。

機能:

#### 1. 新規学生登録:

- 学生番号、学生種別（個人・団体）、氏名、性別、生年月日、婚姻状況、住所、郵便番号、電話番号、税金コード、登録日を登録する。
- 入力された情報は確認画面に表示し、登録するかキャンセルするか選択する。

#### 2. 学生情報変更:

- 学生番号を入力し、変更対象の項目を選択する。
- 選択された項目に応じて新しい値を入力する。
- 変更前に確認画面を表示し、変更を確定するかキャンセルするか選択する。

#### 3. 学生情報参照:

- 学生番号を入力し、登録されている学生情報を表示する。
- 学生種別（個人・団体）に応じて、氏名、性別、生年月日、婚姻状況、住所、郵便番号、電話番号、税金コード、登録日、学生ステータス、死亡日を適切に表示する。

#### 4. 学生情報削除:

- 学生番号を入力し、削除対象の学生情報を表示する。
- 削除を確定するかキャンセルするか選択する。

データ項目:

データ項目	データ型	説明
WS-MENU-OPTION	X(1)	メニュー選択項目
WS-CLIENT-NUMBER	9(8)	学生番号
WS-CLIENT-TYPE	X(1)	学生種別 (P:個人, C:団体)
WS-ID	X(12)	学生ID

## ⑤ テストデータ生成

テストデータ作成のリクエストについて同様に行う

 sample2.txt

テストケースとテストデータを生成する

## ⑤ テストデータとテストケース生成

テストケースとテストデータの作成リクエストについて同様に行う

 sample2.txt

テストケースとテストデータを生成する

## ⑤ テストデータとテストケース生成

AIにテストケースを  
提供してもらったが、  
まだ複数の修正が必要です。

### テストケースとテストデータ生成

このプログラムのテストケースとテストデータは、以下の項目に基づいて生成されます。

#### 1. 機能別テストケース

- **新規クライアント作成:**
  - 正常ケース: すべての入力値が有効な場合（個人、法人）
  - 異常ケース:
    - クライアント番号が重複している場合
    - クライアント種別が不正な場合
    - 必須項目が未入力の場合
    - 入力値がデータ型と一致しない場合（例: 生年月日が不正な形式）
- **クライアント情報変更:**
  - 正常ケース: すべての入力値が有効な場合
  - 異常ケース:
    - 変更対象のクライアントが存在しない場合
    - 変更対象の項目が不正な場合
    - 入力値がデータ型と一致しない場合
- **クライアント情報表示:**
  - 正常ケース: 存在するクライアントの情報が正常に表示される
  - 異常ケース: 存在しないクライアントの情報が表示されない
- **クライアント情報削除:**
  - 正常ケース: 存在するクライアントの情報が削除される
  - 異常ケース: 存在しないクライアントの情報が削除されない
  - 異常ケース: 削除確認で「N」を選択した場合、削除されない

# まとめ

## 利点

素早いコードリ  
ファクタリング

迅速なデザイン  
生成

迅速なテスト  
ケース生成

時間削減

# まとめ

## 留意点

リファクタリング後に微調整が必要な部分があります。

2024年7月17日

THANK FOR  
WATCHING