

# HITACHI

2025年度 新人導入研修

## COBOLの現状と 未来へつなぐ取り組み 「COBOL×生成AI」

2025年5月  
株式会社 日立製作所  
マネージド&プラットフォームサービス事業部  
ミドルウェアサービス本部  
システム&データアプリケーション部

# コボルって、なに？

コボルって、なに？

HITACHI

# ChatGPTに聞きました。

コボルって、何ですか？

COBOL（コボル）は、「COmmon Business-Oriented Language」の略で、1959年にアメリカで生まれたビジネス向けのプログラミング言語です。

ビジネス向けのプログラミング言語ってどういうことですか？

## いい質問です！

「ビジネス向けのプログラミング言語」というのは、**企業や組織の業務処理に特化した言語**という意味です。  
たとえば、以下のような処理を想像してみてください：

### 企業の業務でよくある処理

- ・社員の**給与を計算**する
- ・銀行の**預金や引き出しを管理**する
- ・商品の**在庫や出荷を管理**する
- ・請求書を発行する

こうした処理には、次のような特徴があります：

- ・**数値や日付の計算**が多い（給料、税金など）
- ・**データの形式や桁数がきっちり決まっている**（例：8桁の口座番号）
- ・**巨大な量のデータを処理**する（何百万人分の給与など）

### COBOLは、こうした「業務処理」を得意とするように設計されました

- ・「〇〇円を引いて、××税率をかけて、△△に記録せよ」みたいな**命令が英語っぽく書ける**
- ・**処理の内容が読みやすく、間違いにくい**
- ・巨大な企業システムでも**安定して動く**

つまり、「ビジネス向け」とは、**会社の日々の仕事**（＝事務・経理・販売・物流など）を**正確に、速く、安全にこなすためのプログラミング言語**ということなんです。

# COBOLの特徴がわかるプログラム例

## 請求書金額計算の内容

| 項目        | 値                        |
|-----------|--------------------------|
| 単価        | 1,200円                   |
| 数量        | 15個                      |
| 小計        | 1,200円 × 15 = 18,000円    |
| 消費税 (10%) | 1,800円                   |
| 合計        | 18,000 + 1,800 = 19,800円 |

## 請求書金額計算用のCOBOLプログラム

```

:
01 CUSTOMER-NAME      PIC N(10)  VALUE
01 UNIT-PRICE         PIC 9(5)V99 VALUE
01 QUANTITY           PIC 9(4)    VALUE 15.
01 SUBTOTAL           PIC 9(6)V99.
01 TAX-RATE           PIC V999    VALUE .100.
01 TAX-AMOUNT         PIC 9(6)V99.
01 TOTAL-AMOUNT       PIC 9(6)V99.

PROCEDURE DIVISION.
    DISPLAY "顧客名: " CUSTOMER-NAME

    COMPUTE SUBTOTAL = UNIT-PRICE * QUANTITY
    DISPLAY "小計: " SUBTOTAL

    COMPUTE TAX-AMOUNT = SUBTOTAL * TAX-RATE
    DISPLAY "消費税: " TAX-AMOUNT

    COMPUTE TOTAL-AMOUNT = SUBTOTAL + TAX-AMOUNT
    DISPLAY "合計請求額: " TOTAL-AMOUNT
:

```

COBOLの特徴 (一部)

- データの形式や桁数がきちり決まっている
- 命令が英語っぽく書ける
- 処理の内容が読みやすく

## プログラム実行結果

|                 |
|-----------------|
| 顧客名: 株式会社テスト    |
| 小計: 18000.00    |
| 消費税: 1800.00    |
| 合計請求額: 19800.00 |

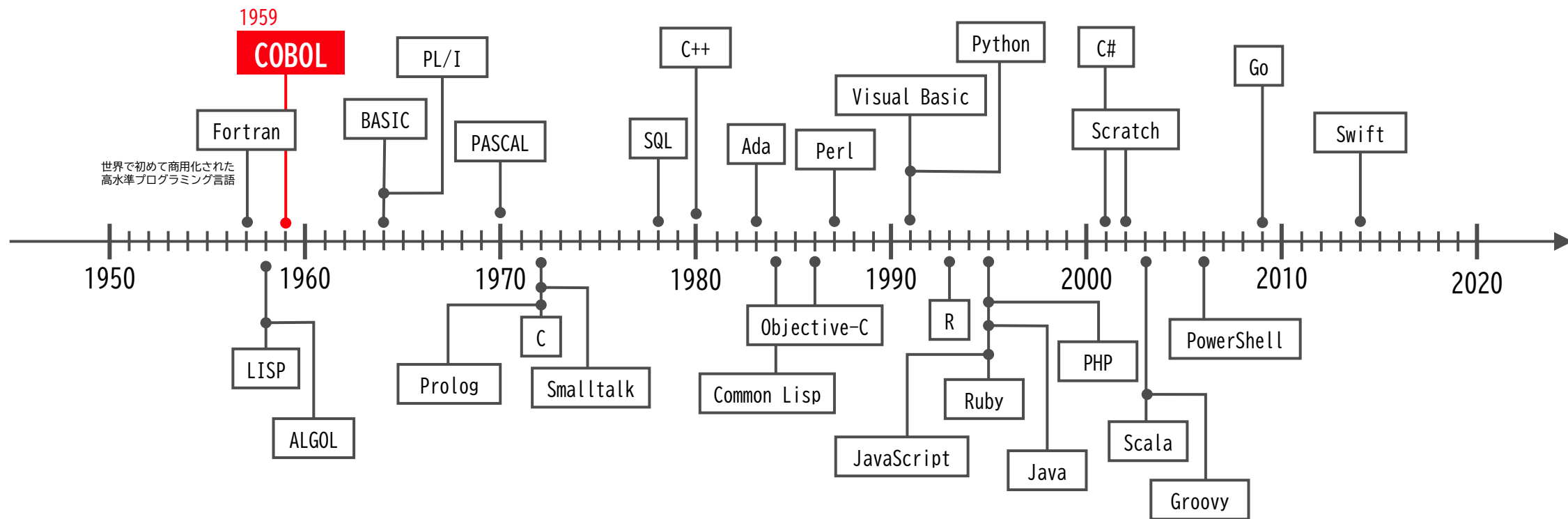
※: 掲載のプログラムは、生成AIを使って作成したプログラムの一部を抜粋したものです。

でも、COBOLって古いよね？

でも、COBOLって古いよね？

HITACHI

# 1959年COBOL誕生、最も古い高水準言語のひとつ



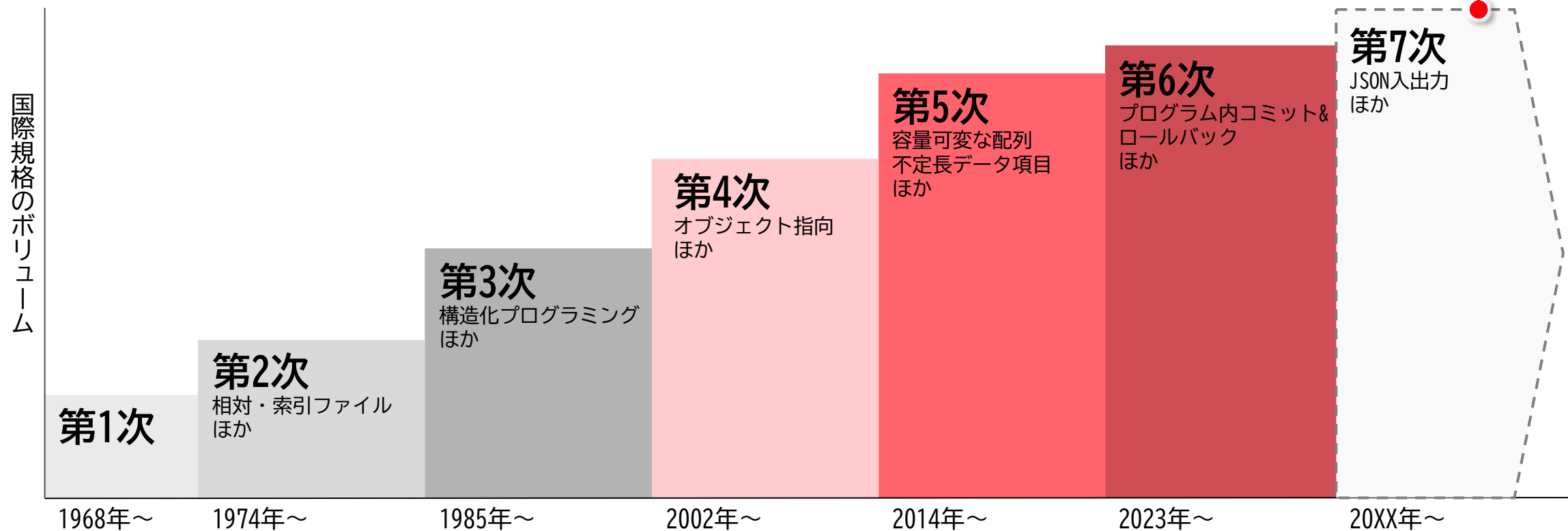
出典 : [https://en.wikipedia.org/wiki/History\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/History_of_programming_languages) より作図

でも、COBOLって古いよね？

HITACHI

# まだまだCOBOLは進化中

- 1960年にCOBOLの最初の仕様書発行
- 1968年から米国規格および国際規格に





どのくらい使われているの？

どのくらい使われているの？

HITACHI

**COBOLは社会基盤を支える基幹系システムで多く採用されています。  
膨大なCOBOLソースコード資産が稼働中です。**



銀行オンラインシステム



座席予約システム（鉄道 ・ 航空機）



国や地方自治体の税金計算システム



ガス・電力料金計算システム



カード会社のクレジットカードシステム



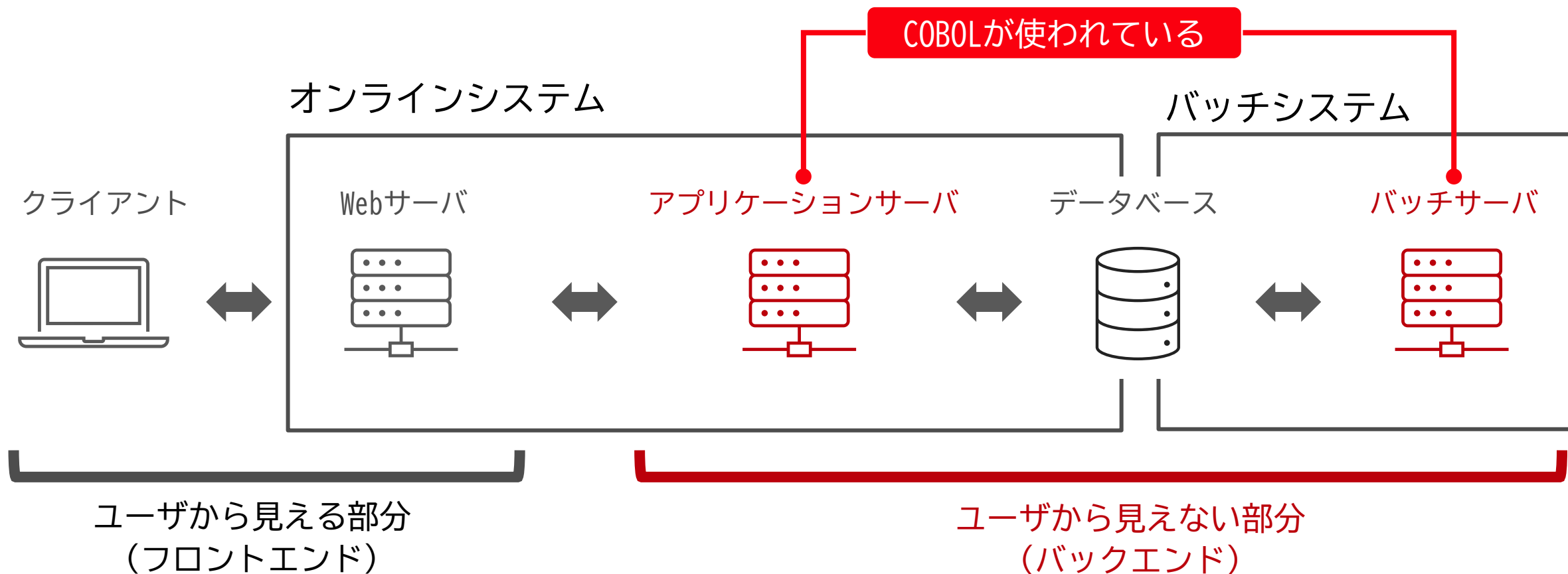
コンビニエンスストアの受発注管理システム

etc.

どのくらい使われているの？

HITACHI

**COBOLはユーザの目に触れない部分でシステムを支えています。**

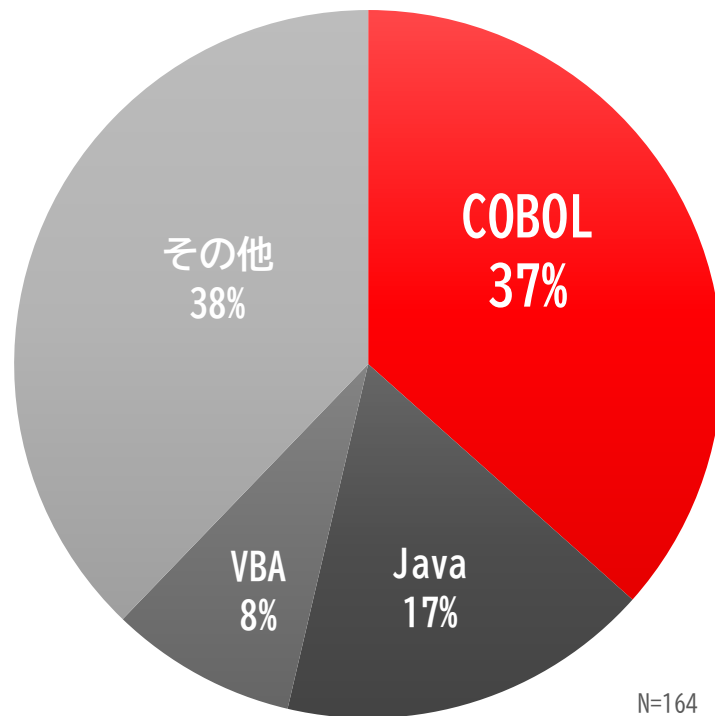


どのくらい使われているの？

HITACHI

# COBOLは、基幹系システムの主要言語として使用されています。

最も使っている言語（基幹系）



※：本スライドは、一般的なCOBOLの市場動向に関する情報の提供を目的としたものです。特定の商品・サービスを推奨しているものではありません。

出典：日経クロステック2024年11月18日掲載 最も使っている言語は「Python」が2連覇、急上昇したのは「COBOL」 (<https://xtech.nikkei.com/atcl/nxt/column/18/03006/111400004/>)より当社作成(一部を抜粋)

# 今後もCOBOLは残るの？

# 「COBOL脱却」 「COBOL継続」

「脱却」か「継続」の議論は1990年代からあります。

## ✓ COBOL脱却

- ・ COBOL技術者の確保が困難
- ・ システムのブラックボックス化
- ・ 最新技術との統合が困難
- ・ クラウドネイティブ化が困難
- ・ DX推進とのギャップ
- ・ 保守/改修に時間とコストがかかる



## 次世代のビジネス基盤への再構築

(日立にはJava移行を支援するサービスもあります)

## ✓ COBOL継続

- ・ 高い安定性と実績
- ・ 膨大な業務知識が蓄積された資産
- ・ 最新技術とのAPI連携も可能
- ・ クラウド/コンテナにも対応可能
- ・ COBOL脱却の移行コストとリスクの回避
- ・ アジリティ高めたいところだけJava化も可能



## 現実的かつ安定的な運用維持

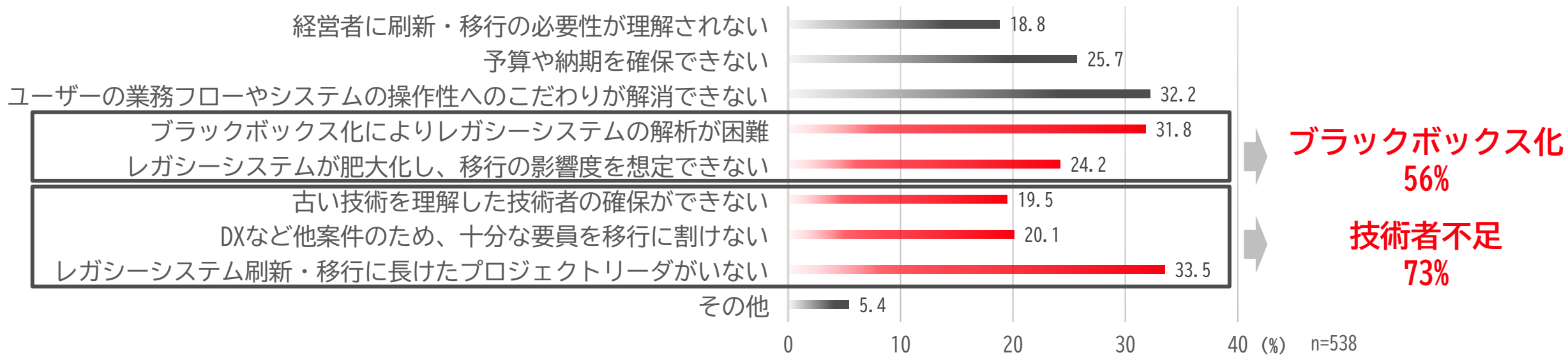


「どちらが正解か」ではなく、

「どちらがお客様にとってベストな選択か」

# 未来へつなぐ取り組み「COBOL×生成AI」

# レガシーシステムを持つお客様の課題



## 課題：ブラックボックス化

長年にわたり、代々の担当エンジニアが  
数多くのメンテナンスを繰り返し、  
システムは肥大化・複雑化、ブラックボックス化。  
整理、ドキュメント化が必要な状況。



## 課題：技術者不足

COBOL技術者の多くは高齢化により  
開発現場から引退し、  
COBOL言語の新規学習者は少ない。



# 生成AI活用によってブラックボックス化と技術者不足を解消

ロジック調査 コードを解説してもらう。



生成AI

このCOBOLプログラムは、データベースの住所録テーブルに対して接続、データの挿入、削除、更新、参照を行います。以下にプログラムの概要と各セクションの役割をまとめます。

ソースコード

```

PROCEDURE DIVISION.

EXEC SQL
  WHENEVER SQLERROR GO TO :CONNEX
END-EXEC.

MOVE 'データソースへの接続' TO STMT-
EXEC SQL
  CONNECT :ODBC-UID IDENTIFIED BY
  USING :ODBC-DSN
END-EXEC.

PERFORM DELETE-PROC.
PERFORM INSERT-PROC.
          
```

プログラミング コードを書いてもらう。



生成AI

住所録に電話番号フィールドが追加された場合、プログラムを修正して新しいフィールドを処理できるようにする必要があります。修正後のプログラムを以下に示します：

ソースコード

```

01 住所録.
02 氏名 PIC X(20).
02 住所.
03 ODBC-length PIC S9(9) USAGE CON
03 ODBC-char PIC X(255).

EXEC SQL END DECLARE SECTION END-EXEC.
77 STMT-PROC PIC X(20).
77 CRS00-ERR PIC 9 VALUE ZERO.
77 FETCH-FLG PIC 9 VALUE ZERO.

PROCEDURE DIVISION.

EXEC SQL
          
```

デバッグ バグを見つけてもらう。



生成AI

提供されたプログラムにはいくつかの潜在的なバグがあります。以下にそれらのバグを説明します：

1. CURSOR-PROCセクション内の条件判断：

ソースコード

```

CURSOR-PROC SECTION.
MOVE '表中の全データの参照'

PERFORM OPEN-PROC.
IF CRS00-ERR = ZERO
THEN
  GO TO STMT-ERROR
END-IF.
PERFORM FETCH-PROC UNTIL FE

PERFORM CLOSE-PROC.
IF CRS00-ERR NOT = ZERO
          
```

テスト テスト用のコードを書いてもらう。



生成AI

テスト用のコードを以下に示します。このコードは、COBOLプログラムの各セクションを順番に実行し、結果を表示するものです。

ソースコード

```

PROCEDURE DIVISION.

EXEC SQL
  WHENEVER SQLERROR GO TO :CONNEX
END-EXEC.

MOVE 'データソースへの接続' TO STMT-
EXEC SQL
  CONNECT :ODBC-UID IDENTIFIED BY
  USING :ODBC-DSN
END-EXEC.

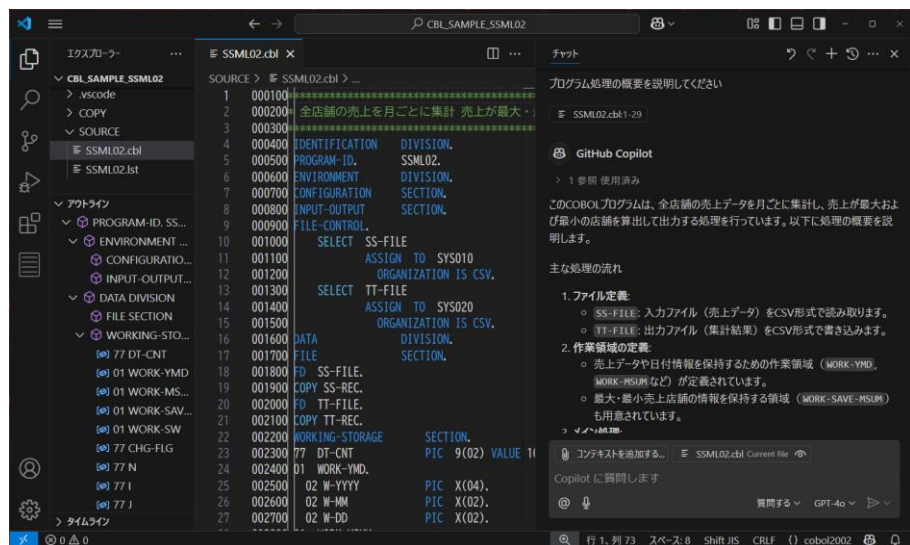
PERFORM DELETE-PROC.
PERFORM INSERT-PROC.
          
```

# 生成AIをシームレスに活用できる Visual Studio Code\* で COBOLアプリケーションの開発ができます。

\* IDEシェア 73% : Stack Overflow Developer Survey 2024  
<https://survey.stackoverflow.co/2024/technology#1-integrated-development-environment>



## Visual Studio Code



## GitHub Copilot



COBOLソースコード／質問

提案

日立COBOLは、お客様の大切なCOBOL資産を今後も安心して  
有効活用できる環境を引き続き提供します。

職場に配属された後、COBOLという用語を耳にする機会があると思います。  
その際は、本日の講演を思い出し、今もなおCOBOLは社会基盤を支える  
重要な言語・技術であることを再認識して頂ければと思います。

# 付録

# 業界団体活動・標準化団体活動

COBOL言語を普及させてCOBOLユーザの利益を守る活動をしています。



## COBOLコンソーシアム（副会長）

COBOLの現状を正しく理解いただくために、2000年に設立された非営利の業界団体

<https://www.cobol.gr.jp/>



## 情報処理学会 情報規格調査会（SC 22/COBOL WG 主査）

国際規格や国内規格の審議・調査研究など、情報技術に関する標準化を目的とする国内の標準化団体

<https://itscj.ipsj.or.jp/>

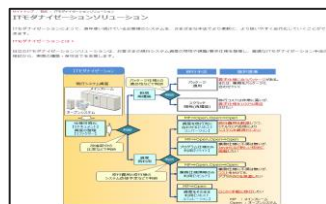
# 関連サイト

COBOL製品やサービス/ソリューションの最新情報を掲載しています。



## COBOL2002製品サイト

<https://www.hitachi.co.jp/Prod/comp/soft1/cobol/>



## ITモダナイゼーションソリューション

<https://www.hitachi.co.jp/Prod/comp/soft1/modernization/>



## ミドルウェア移行支援ソリューション

<https://www.hitachi.co.jp/Prod/comp/soft1/legacy/index.html>

# 商標

- Oracle®、Java、MySQL及びNetSuiteは、Oracle、その子会社及び関連会社の米国及びその他の国における登録商標です。
- その他記載の会社名、製品名は、それぞれの会社の商標または登録商標です。

**HITACHI**