IBM i 2026

IBM i コンテンツ

IBM Bobを使いこなすための基礎知識と操作方法

日本アイ・ビー・エム株式会社
テクノロジー事業本部
Power & Cloud テクニカル・セールス

© IBM 2026

IBM Bobを使いこなすための基礎知識と操作方法

「IBM Bobを使ってみたいけれど、どのように操作すればよいのか分からない」そんな声に応えるべく、今回はIBM Bobの基本操作に焦点を当てます。
IBM Bobは、シンプルなチャット画面から利用できる生成AIアシスタントですが、会話の開始方法や履歴の管理、ファイルのアップロードなど、知っておくと便利な機能が数多く備わっています。
『何ができるか』を知るだけでなく、『どのように使うか』を理解することで、IBM Bobをよりスムーズに業務へ取り入れることができます。
今回は、画面構成から基本的な操作手順まで、初めて利用する方にも分かりやすく解説します。IBM Bob活用の第一歩として、ぜひ参考にしてください。

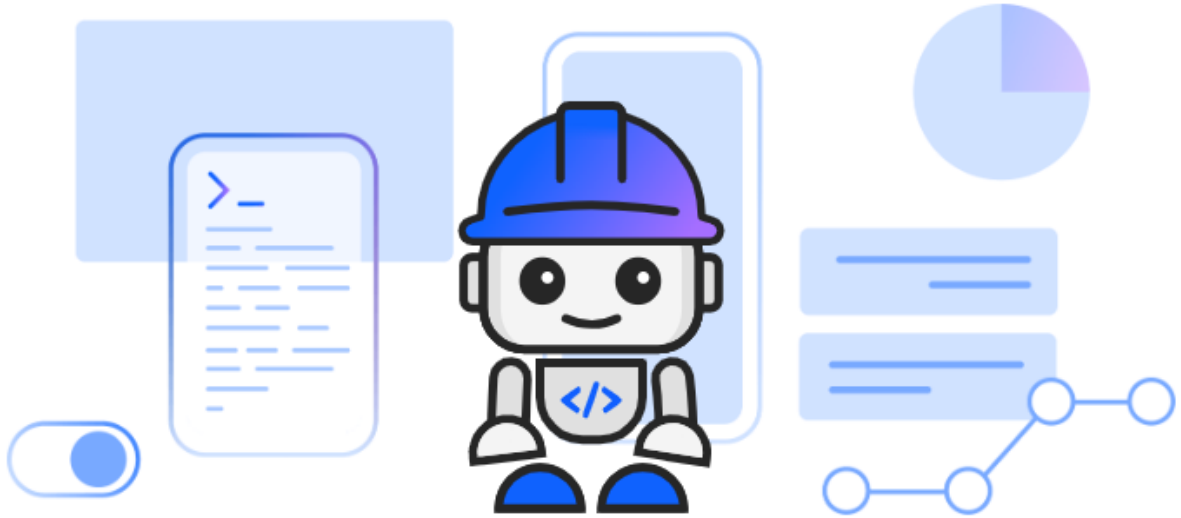
目次

© IBM 2026

1. IBM Bob 概要

© IBM 2026

IBM Bobとは



IBM Bobへようこそ:
あなたのAI駆動開発パートナー

こんにちは、Bobです！私はあなたのコード・ベースで一緒に作業し、
高品質なソフトウェアを迅速に構築するお手伝いをします

無料トライアルを取得　ダウンロード

IBM Bob スタートページ

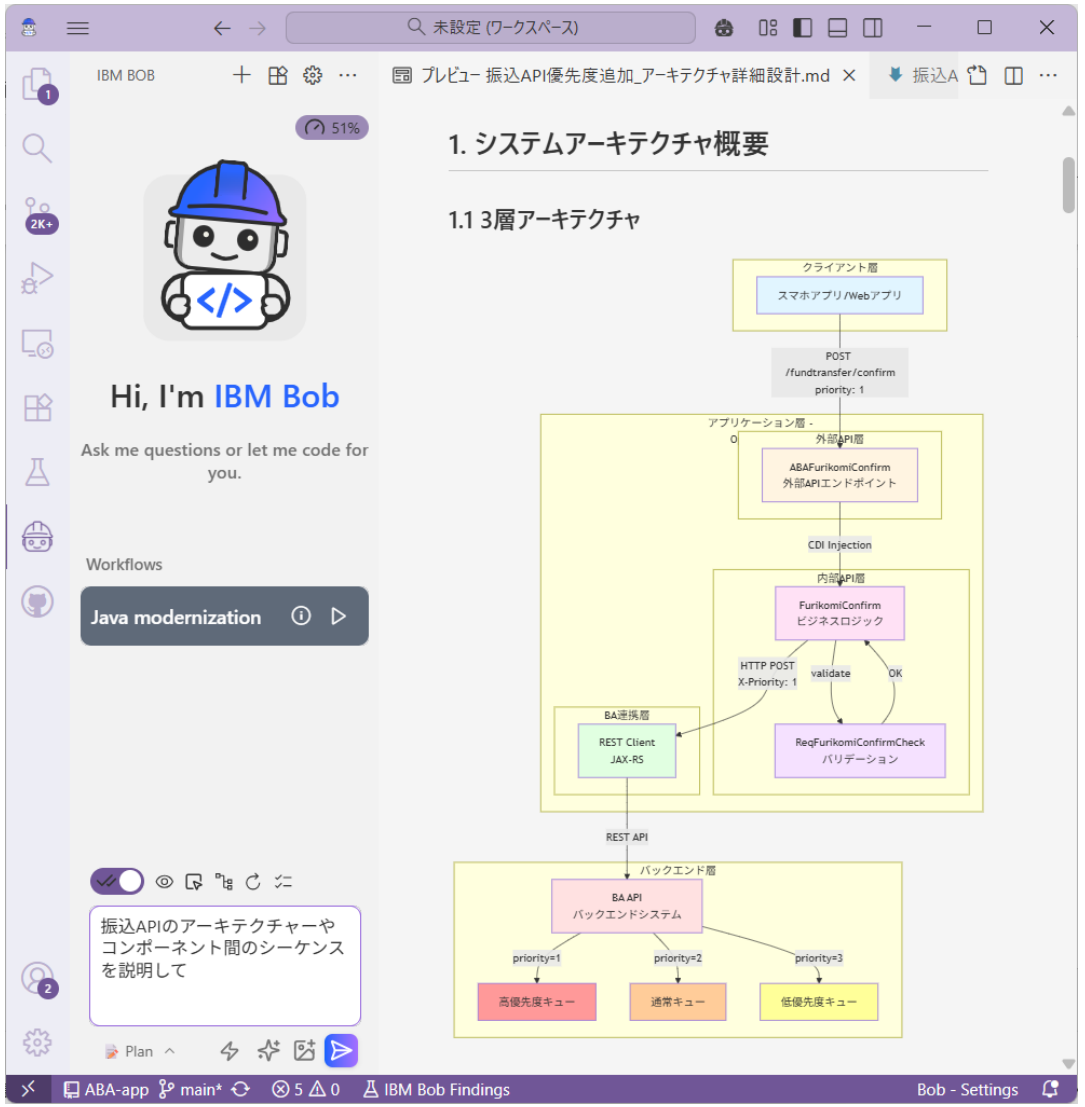
© IBM 2026

AIエージェント駆動の
エンタープライズ向け開発支援パートナー

1	2	3
エンタープライズ志向	ハイブリッド/拡張性	全方位モダナイゼーション
• 統制・制御権 • 安全性 • 監査性 • 再現性	• オンプレミス • クラウド • API • MCP	• モダン言語 • 既存資産 (COBOL、RPG等) • システム開発運用

利用する基盤モデル (LLM or SLM)

- 利用者が意識することなく、タスクに応じた適切なモデルに**自動的に振り分ける**ことにより高い品質 (正確性・性能・コスト) を実現
  - フロンティア・モデル : Claude (Anthropic社)
  - オープンソース・モデル : Mistral, Devstral (Mistral AI社)
  - 専門モデル : Granite (IBM)、カスタム・モデル

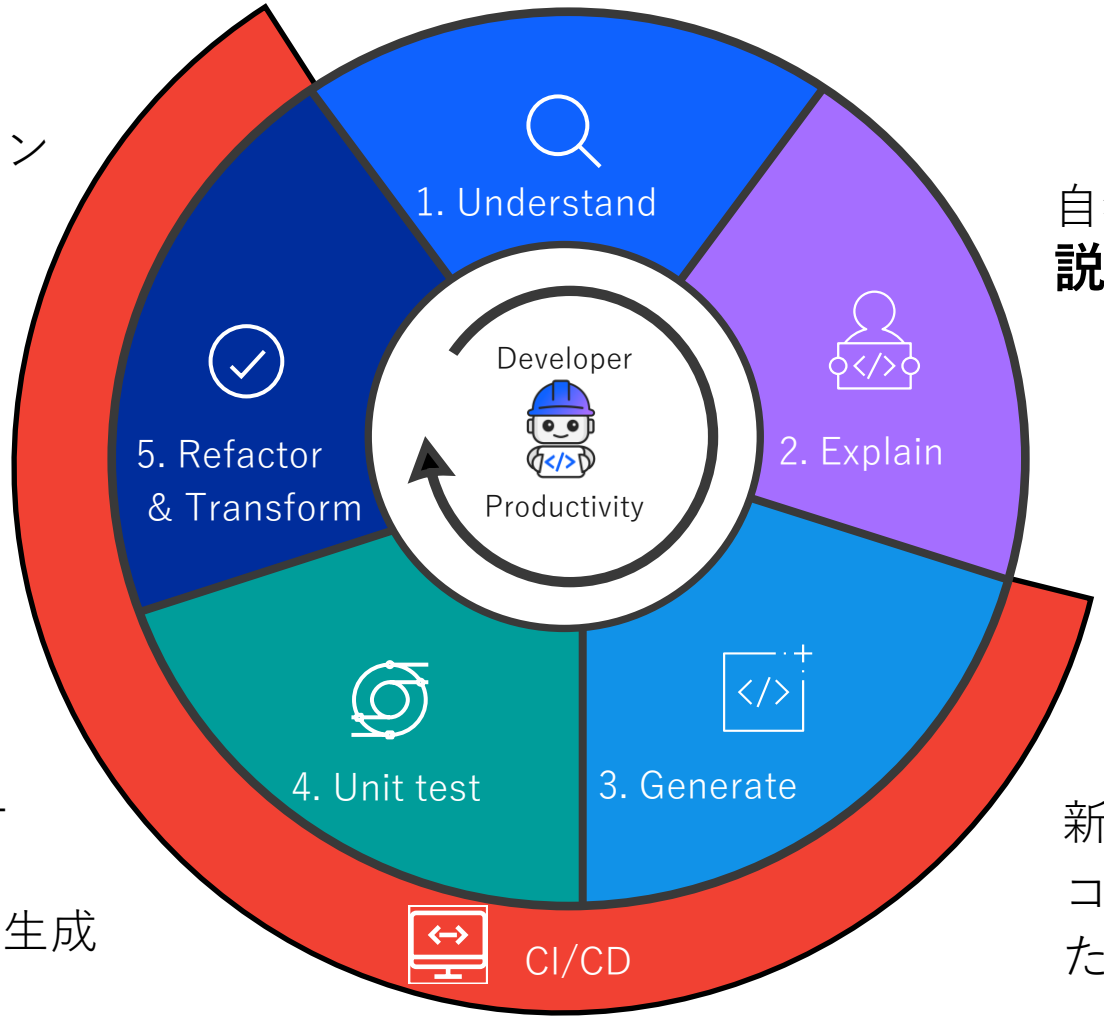


© IBM 2026

AIが支援するIBM i アプリケーションライフサイクル

RPG /COBOLアプリケーションの全体像や
仕様書の関連性を**理解**

モノリシックから
モジュラー構造に**再構成**
古いバージョンのプログラミング言語とデータ定義から、
より新しく、より価値のある
バージョンへ**移行**



自然言語でRPG /COBOLコードを
説明し、ドキュメントを生成

より高品質なRPG /COBOL
コードのために
単体テストプログラムを生成

新しいアプリのRPG /COBOL
コード**生成**や既存アプリ更新の
ためのコード**生成**

© IBM 2026

IBM Bobの導入手順はとても簡単です。

以下はWindows環境におけるインストール手順です。

インストール | Docs | IBM Bob

クライアント環境 システム要件
OS: Windows、mac OS、Linux
メモリ: 4GB(最低)、8GB(推奨)
ストレージ: 500MB以上
その他: アクティブなインターネット環境

IBM Bobのダウンロード
1. Windows用の.exeインストーラーをダウンロードします。
2. ダウンロードした.exeインストーラーを実行します。
3. インストール・ウィザードの指示に従います。
4. インストール・ディレクトリーを選択します（デフォルトを推奨）。
5. 完了をクリックしてインストールを完了します。

IBM id でサインイン
1. IBMid認証情報を求めるサインイン・ダイアログが表示されます。
2. IBMidのメールアドレスとパスワードを入力します。
3. プロンプトが表示されたら、ブラウザーで認証フローを完了します。
4. 認証が完了したら、Bobに戻ります。

© IBM 2026

2. IBM Bob 画面基本構成

© IBM 2026

ここから先の画面構成説明、各種操作デモなどは次のGitHubで公開されているハンズオンワークショップ資料をベースにしております。
操作デモで扱うソースファイルなどは取得頂くことが可能となっています。

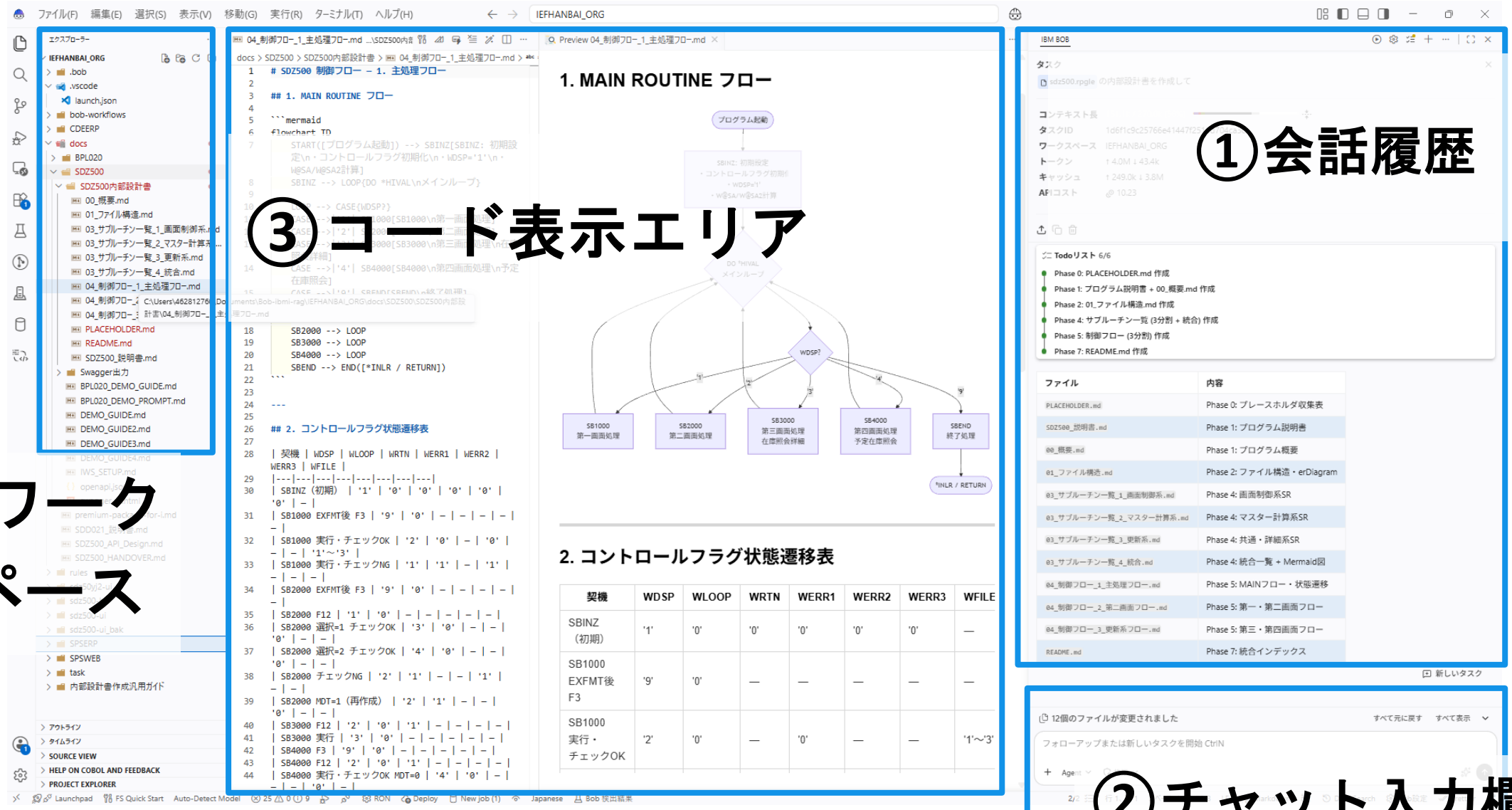
https://github.com/koinumasaoko/ibmi-bob-jp/tree/main/1_ibmi-hands-on-JPAN-TechSales

セクション	内容	配分時間
準備と設定	環境確認とファイルの準備	5-10分
IBM Bobの画面構成	基本操作とインターフェースの理解	5-10分
LAB1: 既存プログラムの理解	RPGコードの解析	15-20分
LAB2: プログラムの修正	画面・プログラムへの項目追加	10-15分
LAB3-FIX: 新規プログラムの作成（固定形式）(オプション)	RPG III固定形式での開発ワークフ	
LAB3-FF: 固定形式から自由形式への変換 (オプション)	固定形式RPGを自由形式に変換	

🚧 **このラボは現在作成中です**

© IBM 2026

IBM Bob 見るべき場所



© 2026 IBM Corporation

3.デモ操作：プログラム修正

© IBM 2026

こちらの章ではパソコンに保存された既存のプログラムソースをIBM Bobに読み込ませ、画面イメージやプログラムに対して項目を追加する内容の操作デモをご覧いただきます。

使用するファイルは8ページでご案内しましたGitHubから取得した以下のファイルとなります。

```
QEOL400/QEOLRPG/iph210.rpg          # RPGプログラム（参照元）
QEOL400/QEOLDSP/iph210s.dspf        # 画面ファイル（参照元）
QEOL400/QEOLDBF/tokmsp.pf           # 得意先マスター定義（参照用）
```

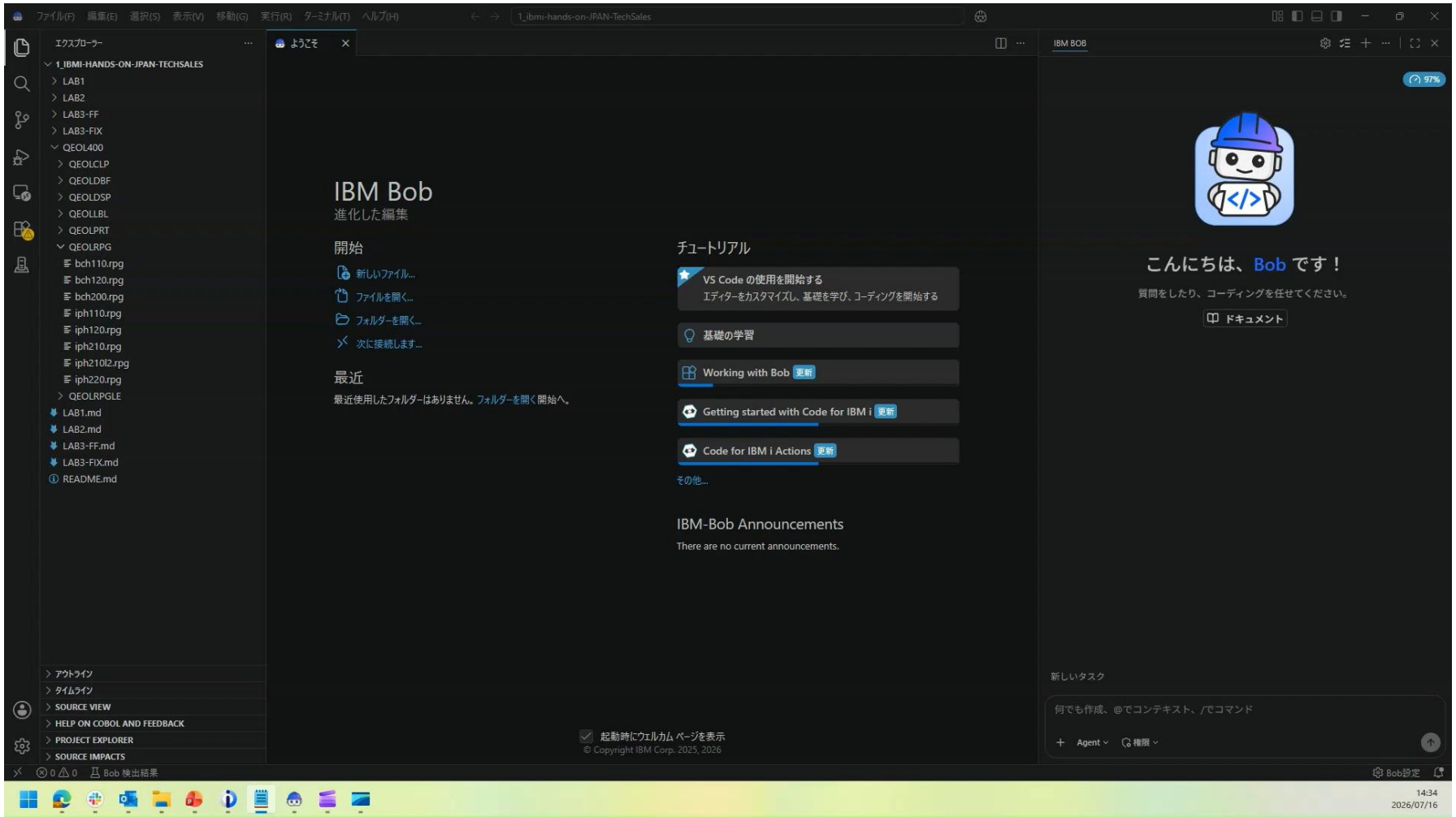
【ご留意事項】
生成AIの特性上、同一のプロンプト（指示）および利用環境であっても、実行の都度、回答内容や生成される成果物が異なる場合があります。
このため、本動画でご紹介するデモ内容と、皆様が実際に利用された際の結果が一致しないことがありますので、あらかじめご了承ください。

© IBM 2026

ステップ1　事前準備

①. 現状の確認操作

「QEOL400/QEOLRPG/iph210.rpgとQEOL400/QEOLDSP/iph210s.dspfの関係を説明してください。このプログラムは何をしていますか？」



© IBM 2026

ステップ2　追加する項目の決定
既存データベースにどのような項目があるか確認した上で追加操作を行います。

①. 利用可能な項目を確認
「tokmsp.pfファイルに定義されている全項目を表形式で教えてください。
現在iph210s.dspfで表示されていない項目はどれですか？」

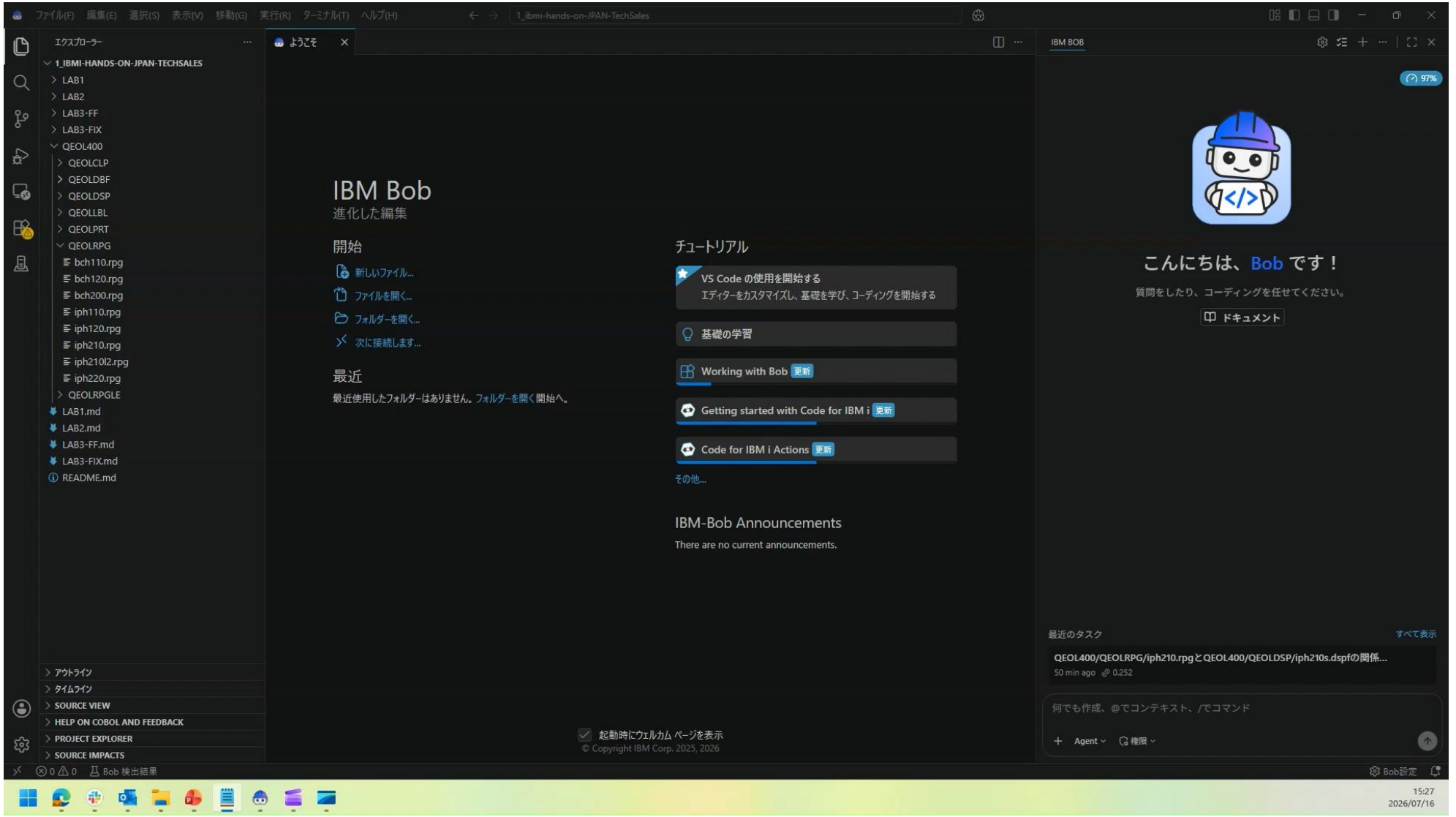
②.追加項目の選定
今回の追加項目
- 郵便番号（TKPOST） - 6桁
- 電話番号（TKTELE） - 13桁

指示内容
「iph210l2.dspfの画面に郵便番号（TKPOST）と電話番号（TKTELE）を追加したいです。
現在の画面レイアウトを考慮して、どこに配置するのが適切か提案してください。」

※途中で生成される会話欄外の情報は動画デモの後に記載しています。

© IBM 2026

ステップ2　追加する項目の決定



© IBM 2026

ステップ2　追加する項目の決定

BobがHTML形式で回答したtokmsp.pfの定義

tokmsp.pf フィールド定義一覧

ファイル: QEOL400/QEOLDBF/tokmsp.pf　（得意先マスター物理ファイル）
対象画面: QEOL400/QEOLDSP/iph210s.dspf（得意先照会画面）

全フィールド一覧（15フィールド）

#	フィールド名	桁数／型	見出し（COLHDG）	iph210s.dspf での表示	表示レコード
1	TKBANG	5桁 文字	得意先番	表示あり	PANEL80（SFLサブファイル行）
2	TKNAKN	20桁 文字	得意先仮名	表示あり	PANEL80（SFLサブファイル行）
3	TKNAKJ	20桁 DBCS(O)	得意先漢字	表示なし	—
4	TKADR1	20桁 DBCS(O)	住所１	表示あり	PANEL80（SFLサブファイル行8行目）
5	TKADR2	20桁 DBCS(O)	住所２	表示あり	PANEL80（SFLサブファイル行9行目）
6	TKTIKU	2桁 文字	地区コード	表示なし	—
7	TKPOST	6桁 文字	郵便番号	表示なし	—
8	TKTELE	13桁 文字	電話番号	表示なし	—
9	TKGURI	9桁0小数 数値	当月売上高	表示なし	—
10	TKNURI	9桁0小数 数値	当年売上高	表示なし	—
11	TKZURI	9桁0小数 数値	前年売上高	表示なし	—
12	TKUZAN	9桁0小数 数値	売掛金残高	表示あり	PANEL80（SFLサブファイル行）
13	TKGEND	9桁0小数 数値	信用限度額	表示あり	PANEL80（SFLサブファイル行）
14	TKNYUK	6桁0小数 数値	最終入金日	表示なし	—
15	TKSIME	1桁 文字	締め日コード	表示なし	—

※ TOKTAN（探索コード入力項目）はiph210s.dspfが独自に定義しているフィールドで、tokmsp.pfには存在しません。

まとめ

- **表示あり（6項目）**：TKBANG（得意先番）、TKNAKN（得意先仮名）、TKADR1（住所１）、TKADR2（住所２）、TKGEND（信用限度額）、TKUZAN（売掛金残高）
- **表示なし（9項目）**：TKNAKJ（得意先漢字）、TKTIKU（地区コード）、TKPOST（郵便番号）、TKTELE（電話番号）、TKGURI（当月売上高）、TKNURI（当年売上高）、TKZURI（前年売上高）、TKNYUK（最終入金日）、TKSIME（締め日コード）

© IBM 2026

ステップ2　追加する項目の決定

iph210s.dspf への TKPOST・TKTELE 追加レイアウト提案

iph210s.dspf —— レイアウト変更提案

郵便番号（TKPOST）・電話番号（TKTELE）の追加配置

現在の画面レイアウト（iph210s.dspf）

```
IPH210S                      得意先照会                      日付: 00/00/00
XXXXXX-XX                                                  時間: 00:00:00

得意先探索コード  XXXXX
                                                        F3=終了  F8=継続

   得番  得意先名                住所１／２              信用限度       売掛残
   XXXXX XXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX  999,999,999  999,999,999
                               XXXXXXXXXXXXXXXXXX
   XXXXX XXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX  999,999,999  999,999,999
                               XXXXXXXXXXXXXXXXXX
   XXXXX XXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX  999,999,999  999,999,999
                               XXXXXXXXXXXXXXXXXX
```

※ サブファイル1レコードが2行使用（住所2行目）。SFLPAG(7) で7件表示。

変更後の画面レイアウト（提案）

既存フィールド　新規追加フィールド　入力フィールド

```
IPH210S                      得意先照会                      日付: 00/00/00
XXXXXX-XX                                                  時間: 00:00:00

得意先探索コード  XXXXX
                                                        F3=終了  F8=継続

   得番  得意先名                住所１／２              信用限度       売掛残
         郵便番号   電話番号

   XXXXX XXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX  999,999,999  999,999,999
         XXXXXX  XXXXXXXXXXXX   XXXXXXXXXXXXXXXXXX
   XXXXX XXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX  999,999,999  999,999,999
         XXXXXX  XXXXXXXXXXXX   XXXXXXXXXXXXXXXXXX
```

※ iph210l2.dspf の解答例に基づくレイアウト。
サブファイル1レコードを2行構成のまま維持し、2行目（住所２と同じ行）に郵便番号・電話番号を並べる。

配置座標の詳細

項目	フィールド名	桁数	行（SFL内）	列	備考
郵便番号ラベル	定数 '郵便番号'	—	8行目（PANEL90）	col 8	列見出し行に追加
電話番号ラベル	定数 '電話番号'	—	8行目（PANEL90）	col 20	郵便番号の右隣
郵便番号	TKPOST	6桁	+1行目 → SFL 11行目	col 10	住所2行目（TKADR2）と同じ行
電話番号	TKTELE	13桁	+1行目 → SFL 11行目	col 17	TKPOSTの右隣（col 10+6+1）

iph210s.dspf への追記内容（DDS差分）

```
A          R PANEL80                   SFL
A            TKBANG        5     8  4
A            TKNAKN       20     8 11
A            TKADR1       20     8 32
+A           TKPOST        6    11 10
+A           TKTELE       13    11 17
A            TKADR2       20     9 32
A            TKGEND        9  0  8 55EDTCDE(J)
A            TKUZAN        9  0  8 68EDTCDE(J)
...
A                                7 32'住所１／２'
A                                7 57'信用限度'
A                                7 72'売掛残'
+A                               8  8'郵便番号'
+A                               8 20'電話番号'
```

設計ポイント

- **行数維持**：SFLPAG(7) 設定のまま、1レコード=2行構成を変えずに2行目を有効活用。
- **住所２との共存**：TKADR2（col 32〜）と TKPOST/TKTELE（col 10〜30）は重ならない。
- **ラベル行の拡張**：PANEL90 の7行目見出しに対し、8行目に郵便番号・電話番号の副見出しを追加。
- **列の整合性**：TKPOST(6桁) は col 10 〜 15、TKTELE(13桁) は col 17 〜 29 に収まり80桁内に収まる。

© IBM 2026

ステップ3　画面ファイルの修正

①. 現在のレイアウトの確認
「iph210l2.dspfのPANEL80レコード（サブファイル明細）の
現在のレイアウトを図で説明してください。」

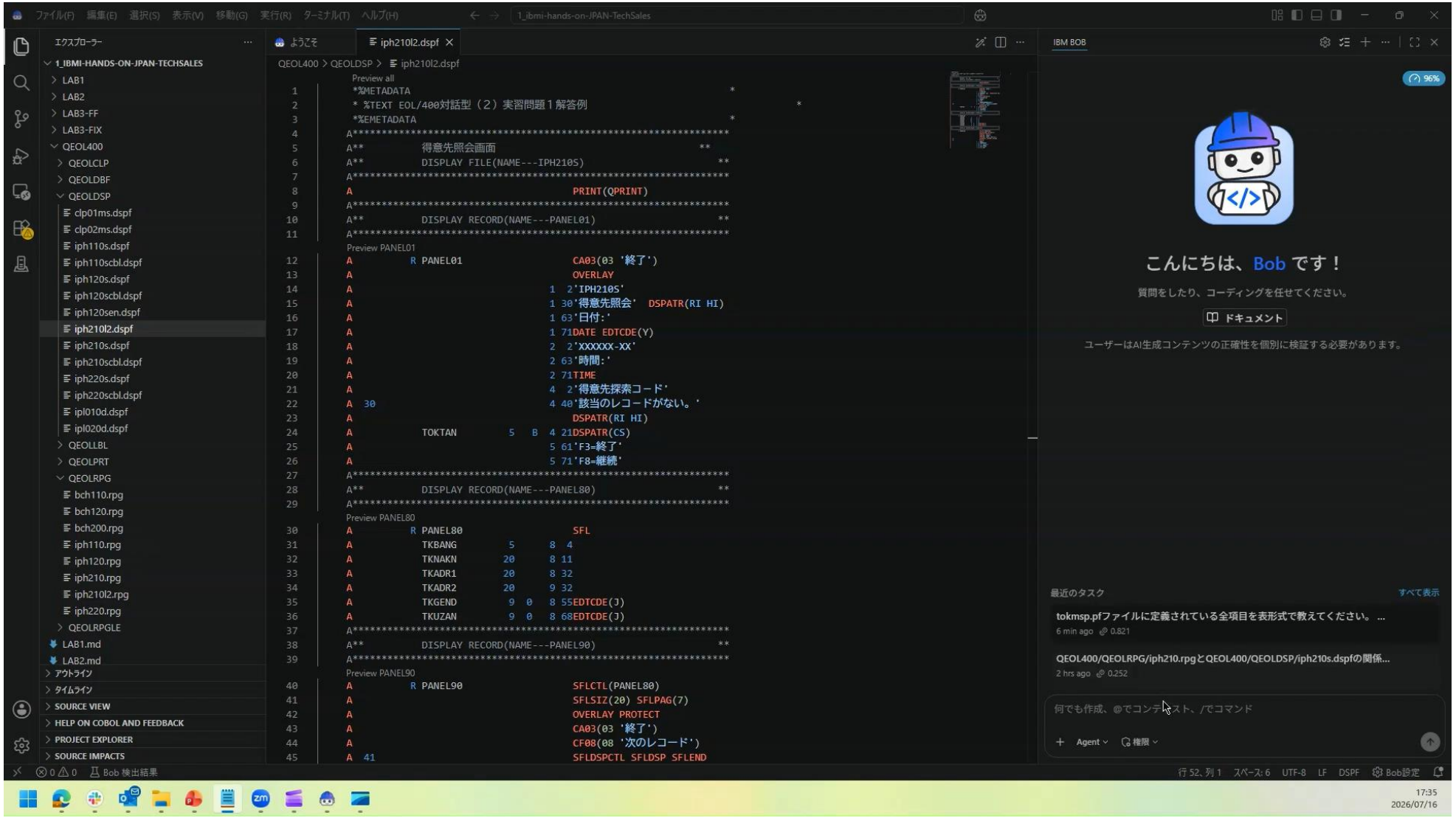
②.修正案の作成
「iph210l2.dspfのPANEL80レコードに以下の項目を追加してください：
　　郵便番号（TKPOST）：6桁、行11の10桁目から表示
　　電話番号（TKTELE）：13桁、行11の17桁目から表示
修正後のDDSコードを提示してください。」

③.見出し行の修正
「iph210l2.dspfのPANEL90レコードの見出し行も、追加した項目に合わせて修正してください。」

※途中で生成される会話欄外の情報は動画デモの後に記載しています。

© IBM 2026

ステップ3　画面ファイルの修正



© IBM 2026

ステップ3　画面ファイルの修正

サブファイル明細のレコード図

PANEL80 — サブファイル明細レコード レイアウト図

ファイル: **iph210l2.dspf** | レコード: **PANEL80 (SFL)** | 対応制御レコード: **PANEL90 (SFLCTL)** | SFLSIZ: 20 / SFLPAG: 7

画面イメージ（行7〜行9）

```
IPH210S                    得意先照会                          日付: 99/99/99
XXXXXX-XX                                                      時間: 99:99:99

  得番  得意先名                住所１／２              信用限度  売掛残
  BBBBB AAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAA 999,999,999 999,999,999
                                        AAAAAAAAAAAAAAAAAAAA
  BBBBB AAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAA 999,999,999 999,999,999
                                        AAAAAAAAAAAAAAAAAAAA
                                  F3=終了     F8=継続
```

列位置マップ（行 8 — データ行）

フィールド定義詳細

フィールド名	桁数	属性	行	列	列終端	編集	説明
TKBANG	5	文字	8	4	8	—	得意先番号
TKNAKN	20	文字	8	11	30	—	得意先名（カナ）
TKADR1	20	文字	8	32	51	—	住所１（同行に表示）
TKADR2	20	文字	**9**	32	51	—	住所２（次行・同列に表示）
TKGEND	9,0	数値	8	55	63	EDTCDE(J)：カンマ区切り	信用限度額
TKUZAN	9,0	数値	8	68	76	EDTCDE(J)：カンマ区切り	売掛残高

行構造のポイント

行	内容
行 7（制御: PANEL90）	列 3:「得番」/ 列 11:「得意先名」/ 列 32:「住所１／２」/ 列 57:「信用限度」/ 列 72:「売掛残」（カラムヘッダ）
行 8（SFL 1行目）	TKBANG / TKNAKN / TKADR1 / TKGEND / TKUZAN
行 9（SFL 2行目）	TKADR2 のみ（列 32）— 住所２を次の行に折り返し表示

⚠ **TKADR2 は行 9 に折り返しされます。**つまり 1 サブファイルレコードが **2 行を占有**しているため、SFLPAG(7) で最大 7 レコード表示時は実質 14 行を使用します。

© IBM 2026

ステップ4　RPGプログラムの修正

①. 元のプログラムからの変更点を確認

「iph210.rpgを参考に、iph210l2.rpgを以下の要件で修正を検討してください：
1. 画面ファイル名をiph210l2.dspfに変更
2. 郵便番号（TKPOST）と電話番号（TKTELE）の初期化処理を追加
3. 得意先番号（TKBANG）で検索するように変更（現在は仮名検索）
RPG III画面開発の規約（.bob/rules/rules-code/rpg3-screen-development.md）を
参照して、適切な修正を提案してください。」

②. 具体的な修正内容の確認

「iph210l2.rpgの修正内容を、修正前と修正後のコードで示してください。
特に以下の点を明確にしてください：
1. F仕様書のファイル定義
2. 検索ロジック（SETLL、READ命令）
3. 初期化サブルーチン」

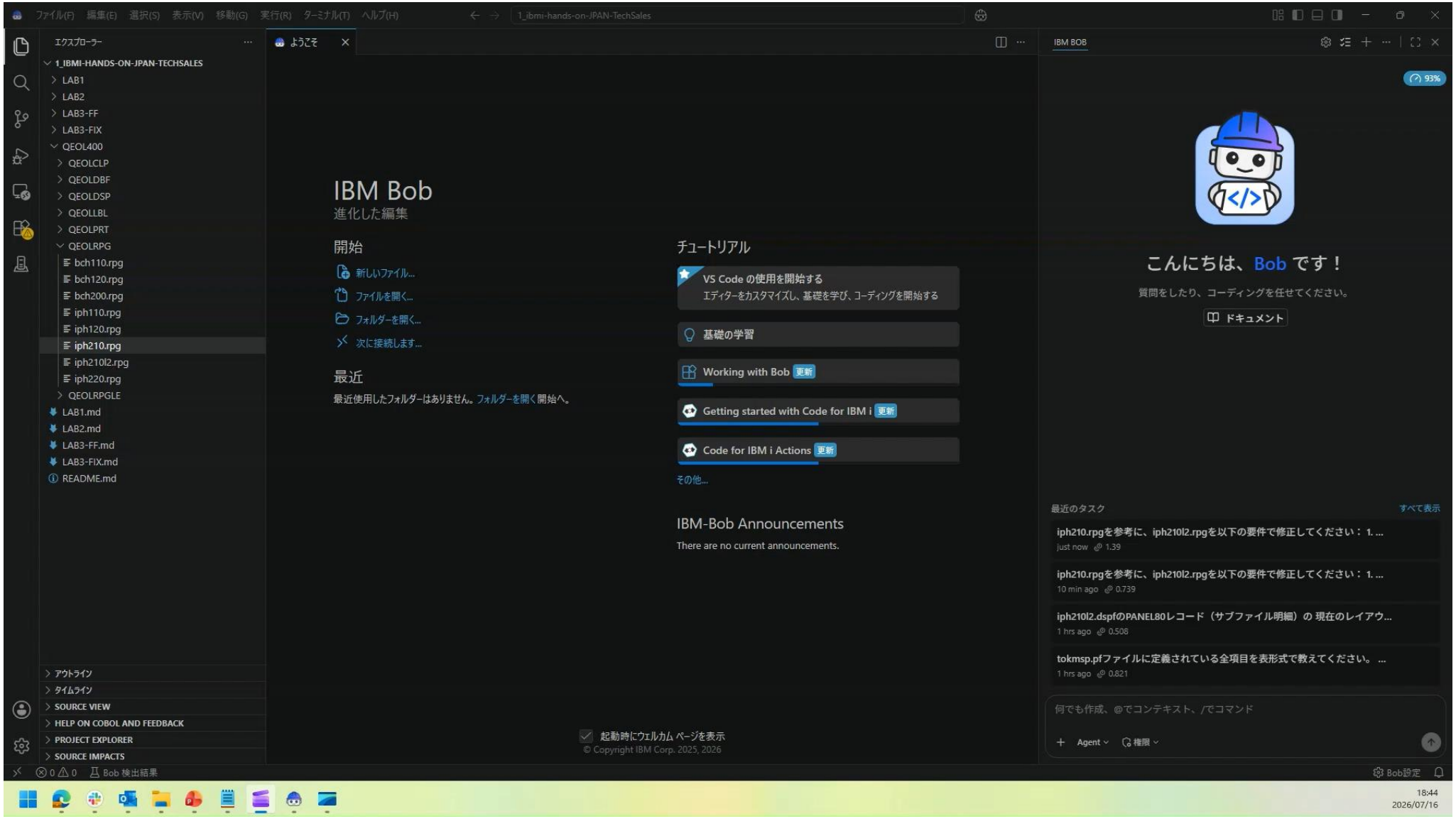
③. 修正内容の確認

「今回の修正でプログラムの動作に影響はありますか？テストで確認すべきポイントを教えてください。」

※途中で生成される会話欄外の情報は動画デモの後に記載しています。

© IBM 2026

ステップ4　RPGプログラムの修正



© IBM 2026

ステップ4　RPGプログラムの修正
プログラム修正レポート

iph210l2.rpg 修正レポート

QEOL400/QEOLRPG/iph210l2.rpg — LAB2 実習問題 1 修正適用

3	8	0
修正要件数	変更行数（追加含む）	既存ロジック削除

修正サマリー

#	種別	対象	修正内容
1	変更	F仕様・Hコメント	画面ファイル名を `IPH210S` → `IPH210L2` に変更。 Hコメント行（プログラム識別子）を追加。 DBファイルを `TOKMSL02`（仮名キー論理）→ `TOKMSP`（物理ファイル）に変更。
2	変更	C仕様 ファイル・ポインターのセット	検索キーを `TKNAKN`（仮名）→ `TKBANG`（得意先番号）に変更。 SETLL対象も `TOKMSL02` → `TOKMSP` に変更。
3	追加	CLEARサブルーチン	`TKPOST`（郵便番号）と `TKTELE`（電話番号）の `MOVE *BLANK` 初期化処理を追加。 サブルーチン名を `¥CLEAR` → `CLEAR` に統一（EXSR呼び出し側も同時修正）。

変更①：Hコメント追加 ＋ 画面ファイル・DBファイル名変更（F仕様）

iph210l2.rpg — 行 4〜11

```
-     F* IPH210 * 得意先採索
-     FIPH210S CF  E                    WORKSTN
-     F                                       RRN01 KSFILE PANEL80
-     FTOKMSL02IF  E           K        DISK
+     H*************************************************************
+     H* IPH210L2 * 得意先採索
+     H*************************************************************
+     H           Y
+     F* IPH210L2 * 得意先採索
+     FIPH210L2CF  E                    WORKSTN
+     F                                       RRN01 KSFILE PANEL80
+     FTOKMSP  IF  E           K        DISK
```

参照画面ファイル	`QEOL400/QEOLDSP/iph210l2.dspf`（既存）
参照DBファイル	`QEOL400/QEOLDBF/tokmsp.pf`（物理ファイル、KEY=TKBANG）
旧参照DBファイル	`QEOL400/QEOLDBF/tokmsl02.lf`（論理ファイル、KEY=TKNAKN）

変更②：検索キーを仮名 → 得意先番号（TKBANG）に変更

iph210l2.rpg — 行 17〜20, 26

```
C* ファイル・ポインターのセット
-     C                   MOVEL*BLANK    TKNAKN
-     C                   MOVELTOKTAN    TKNAKN
-     C           TKNAKN  SETLLTOKMSL02          30
+     C                   MOVEL*BLANK    TKBANG
+     C                   MOVELTOKTAN    TKBANG
+     C           TKBANG  SETLLTOKMSP            30
 C   30              GOTO @START
 ...
-     C                   READ TOKMSL02              50
+     C                   READ TOKMSP                50
```

画面入力値 `TOKTAN` を `TKBANG`（得意先番号）に転送し、物理ファイル `TOKMSP` に対して SETLL でポジショニング。READも同ファイルに統一。

変更③：CLEAR サブルーチンに TKPOST/TKTELE 初期化を追加

iph210l2.rpg — 行 32, 52〜61

```
 C           RRN01     IFEQ 0
-     C                   EXSR ¥CLEAR
+     C                   EXSR CLEAR
 ...
-     C           ¥CLEAR  BEGSR
+     C           CLEAR   BEGSR
      C                   MOVE *BLANK    TKBANG
      C                   MOVE *BLANK    TKNAKN
      C                   MOVE *BLANK    TKADR1
      C                   MOVE *BLANK    TKADR2
+     C                   MOVE *BLANK    TKPOST
+     C                   MOVE *BLANK    TKTELE
      C                   Z-ADD0         TKGEND
      C                   Z-ADD0         TKUZAN
      C                   ENDSR
```

TKPOST	郵便番号（6桁文字）— `MOVE *BLANK` でスペース初期化
TKTELE	電話番号（13桁文字）— `MOVE *BLANK` でスペース初期化
サブルーチン名	`¥CLEAR` → `CLEAR`（EXSR呼び出し側も統一）

© IBM 2026

ステップ4　RPGプログラムの修正

修正前後のコード比較

iph210l2.rpg 修正前後コード比較

QEOL400/QEOLRPG/iph210l2.rpg — 3点の修正箇所を並列で比較

1 F仕様書のファイル定義

修正前

```
     F* IPH210 * 得意先探索
     FIPH210S CF  E                    WORKSTN
     F                                       RRN01 KSFILE PANEL80
     FTOKMSL02IF  E           K        DISK
```

修正後

```
     H*****************************
     H* IPH210L2 * 得意先探索
     H*****************************
     H           Y
     F* IPH210L2 * 得意先探索
     FIPH210L2CF  E                    WORKSTN
     F                                       R
     FTOKMSP  IF  E           K        DISK
```

変更点：

① 画面ファイル名 IPH210S → IPH210L2（= iph210l2.dspf を参照）
② DBファイル TOKMSL02（論理ファイル、KEY=TKNAKN 仮名）→ TOKMSP（物理ファイル、KEY=TKBANG 得意先番号）
③ Hコメント行（プログラム識別ブロック）を新規追加

旧DBファイル	TOKMSL02（tokmsl02.lf）— 仮名キー（TKNAKN）の論理ファイル
新DBファイル	TOKMSP（tokmsp.pf）— 得意先番号キー（TKBANG）の物理ファイル

2 検索ロジック（SETLL・READ 命令）

修正前（仮名検索）

```
C* ファイル・ポインターのセット
C                   MOVEL*BLANK   TKNAKN
C                   MOVELTOKTAN   TKNAKN
C           TKNAKN  SETLLTOKMSL02         30
C   30              GOTO @START
...
C           @READ   TAG
C                   READ TOKMSL02         50
```

修正後（得意先番号検索）

```
C* ファイル・ポインターのセット
C                   MOVEL*BLANK   TKBANG
C                   MOVELTOKTAN   TKBANG
C           TKBANG  SETLLTOKMSP           30
C   30              GOTO @START
...
C           @READ   TAG
C                   READ TOKMSP           50
```

変更点：

① 検索キーフィールドを TKNAKN（得意先仮名）→ TKBANG（得意先番号）に変更
② SETLL のターゲットファイルを TOKMSL02 → TOKMSP に変更
③ READ のターゲットファイルを TOKMSL02 → TOKMSP に変更
④ 画面入力値 TOKTAN を MOVEL で TKBANG の先頭からセットし、SETLL でポジショニング後に順次 READ

MOVEL*BLANK	TKBANG フィールドをスペースで初期化してから TOKTAN を転送することで、入力値未満の番号を確実に除外
SETLL + 30	該当レコードなし（EOF以降）の場合にインジケーター 30 がオン → @START へ戻り再入力を促す
READ + 50	EOF 到達でインジケーター 50 がオン → ループ終了

© IBM 2026

4.まとめ・ネクストステップ

© IBM 2026

本日のまとめ

- ◆ IBM Bobは自然言語で対話しながら利用するAI開発支援ツール
- ◆ 基本操作は「チャットで指示 → 結果を確認」というシンプルな流れ
- ◆ コード理解や修正、ドキュメント生成など、日常的な開発作業を効率化できる
- ◆ IBM Bob単体ではIBM iに直接接続はできない

ネクストステップ

① サンプルコードを使ってプログラム説明を試してみる
② 自分が担当しているソースコードの解析を試す
③ 小規模な修正やリファクタリングを依頼してみる
④ チーム内で活用事例を共有する

最初に試してほしいプロンプト例

- このプログラムの処理内容を説明してください
- このプログラムの入出力を整理してください
- このプログラムの改善点を提案してください
- この処理の仕様書を作成してください

© IBM 2026

© IBM 2026