COBOL活用ガイド No.001

入門

HITACHI
Inspire the Next

単体テスト支援 ファーストステップガイド

COBOL2002 Developer Professional 単体テスト支援

株式会社 日立製作所

version 1.2(201705)

はじめに

本資料はCOBOL2002製品群をより便利に活用するためのガイドです。

本資料は以下の製品が対象です。
- ◆ COBOL2002 Developer Professional 03-04以降
- ◆ COBOL2002 Developer Professional(64) 03-04以降

「単体テスト支援ファーストステップガイド」では
単体テストの必要性や単体テスト支援のメリットを紹介します。
さらに，単体テスト支援の基本的な使い方をサンプルコードをベースに体験できます。

体験できる内容は以下です。
- ◆ 単体テスト支援の基本的な一連の流れ

体験にかかる時間の目安は約30分程度です。

入門

COBOL活用ガイド No.001

単体テスト支援 ファーストステップガイド

HITACHI
Inspire the Next

Contents

入門

COBOL活用ガイド No.001

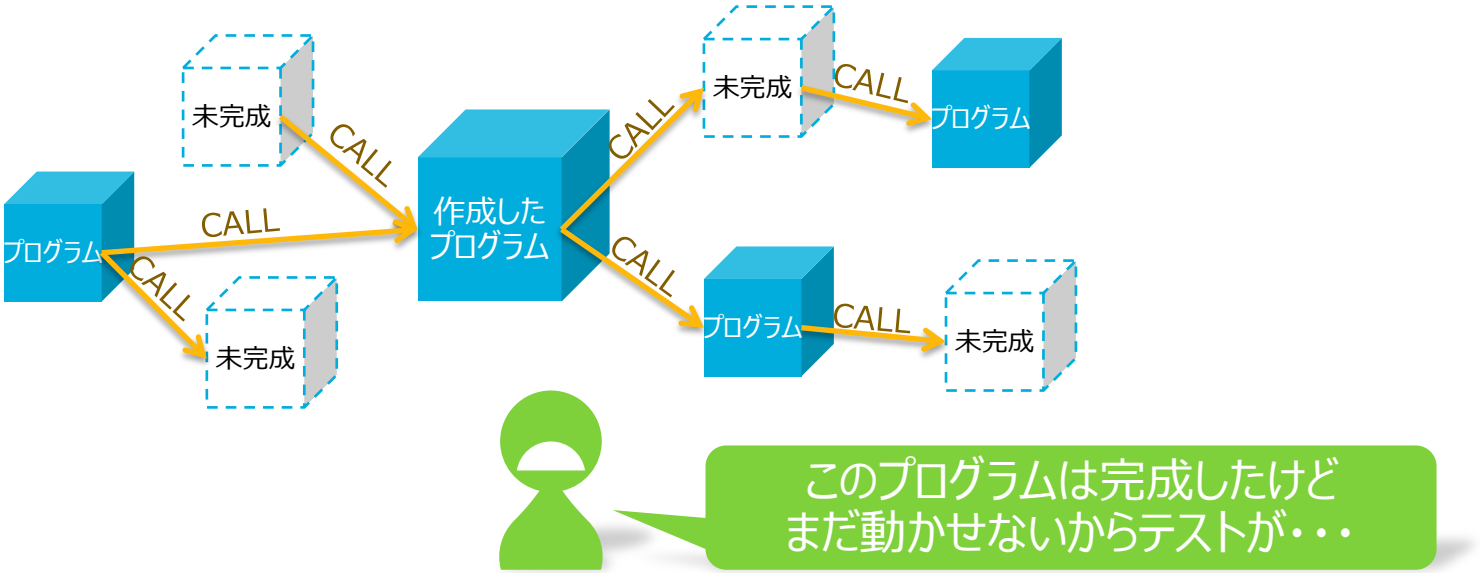
単体テスト支援 ファーストステップガイド

HITACHI
Inspire the Next

1. 単体テストとは

1-1. 単体テストとは(1)

システムの開発/改修ではテストが不可欠です。



しかし，全てが完成してからようやくテスト開始だと
バグの特定や修正などコストが高くなってしまいます。

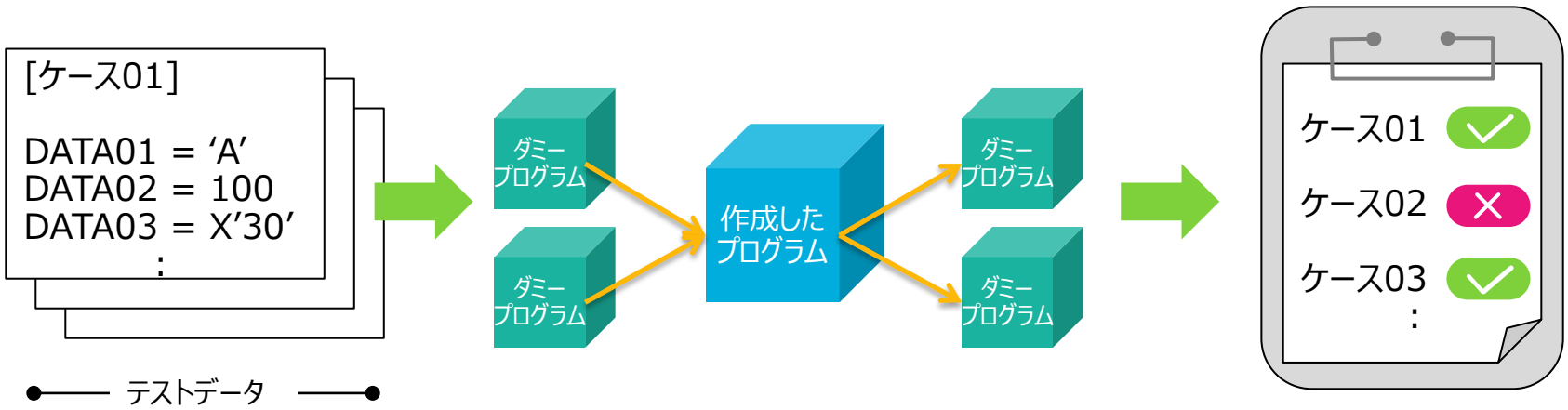
そこで早期に品質を向上させる手段の一つとして「単体テスト」があります。

1-2. 単体テストとは(2)

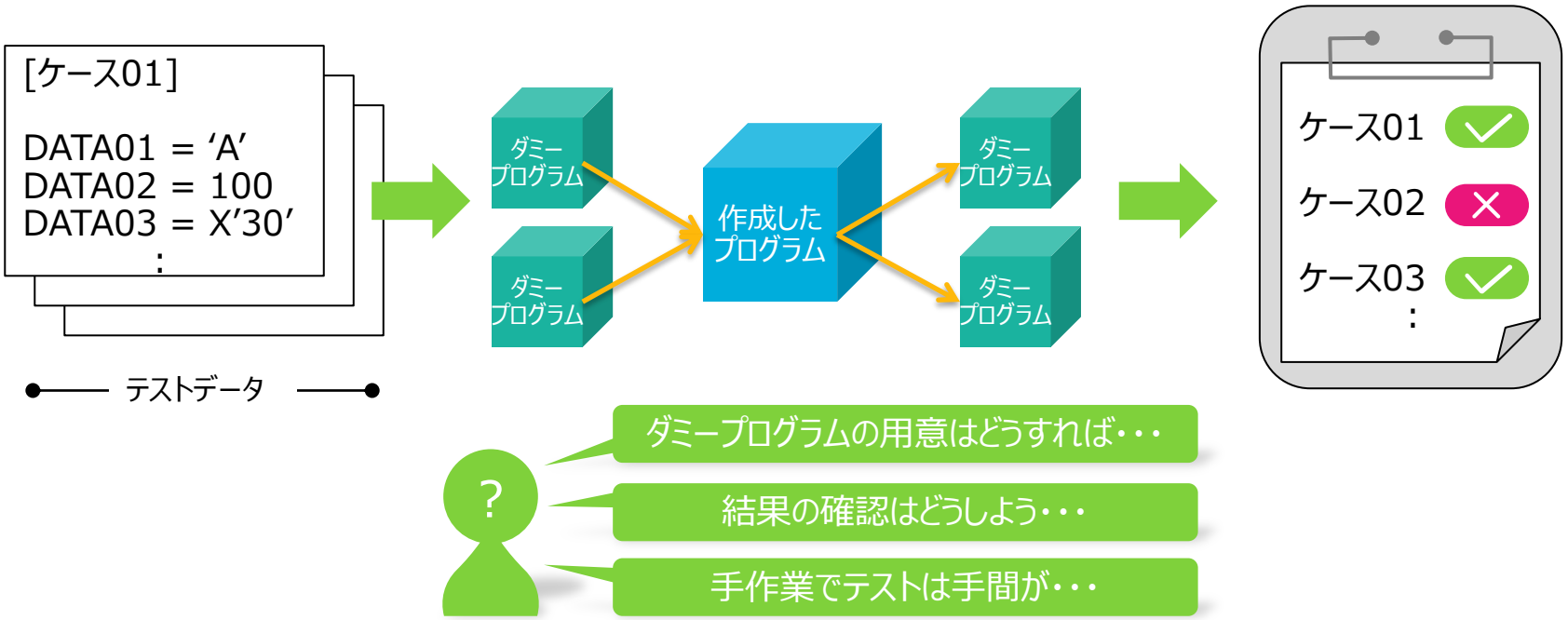
単体テストとは
「対象のプログラムを単独で動作させ，動作が想定通りかどうかを確認する」テスト技法です。

呼び出し元や先のプログラムはダミーを用意し，入力値と期待値をテストデータとして設定して想定通りの動作を行うかの確認が一般的です。

システム全体が完成していなくても完成したプログラムから順次テストすることができるため，早期に品質確保を図ることができます。



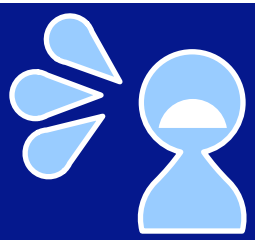
1-3. 単体テスト導入のポイント



単体テストの導入には初期コストや運用コストなど投資が必要です。
投資に見合う結果が得られるよう様々な点を考慮することが重要です。

COBOL2002にはこれらのコストを下げる「単体テスト支援」があります。

1-4. 単体テスト支援とは

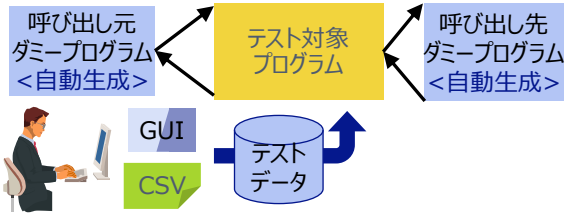


バグを早期に発見したい！

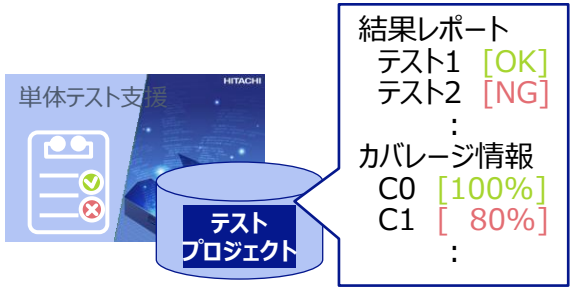
手作業のテストではもう対応できない！

そのお悩み　**単体テスト支援が解決します！**

準備が簡単に！



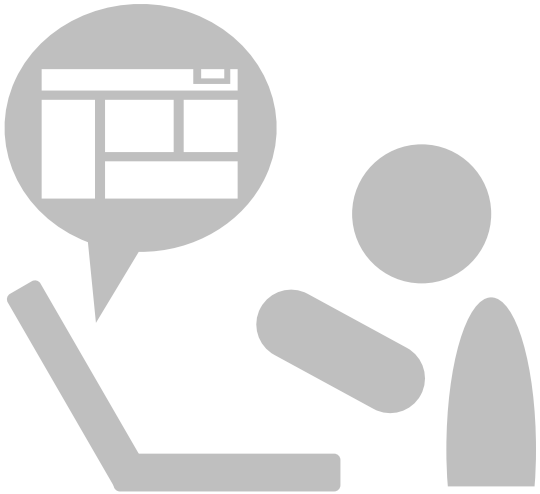
結果確認が簡単に！



繰り返しテストが簡単に！

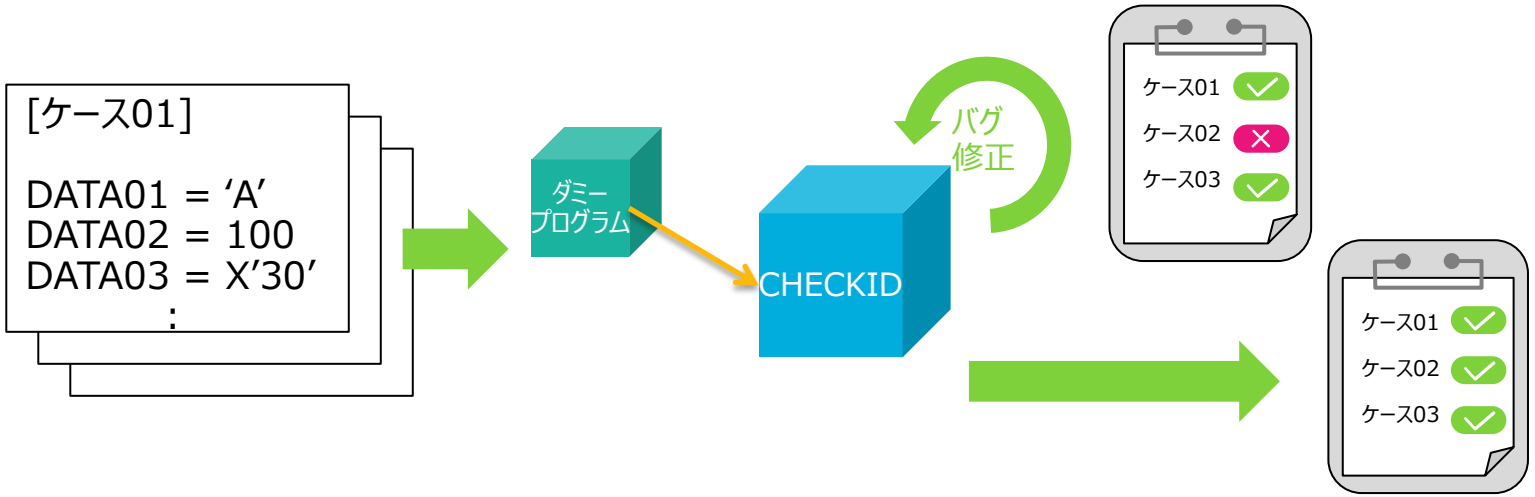
HITACHI
Inspire the Next

2. 単体テスト支援を実際に操作してみましょう



2-1. 体験する内容

単体テスト支援で，テストプロジェクトの作成からテストデータの設定，単体テストの実行を行います。
また，サンプルコードにはあらかじめバグが含まれており，テストによるバグの発見から修正/再実行による確認も行います。



Goal

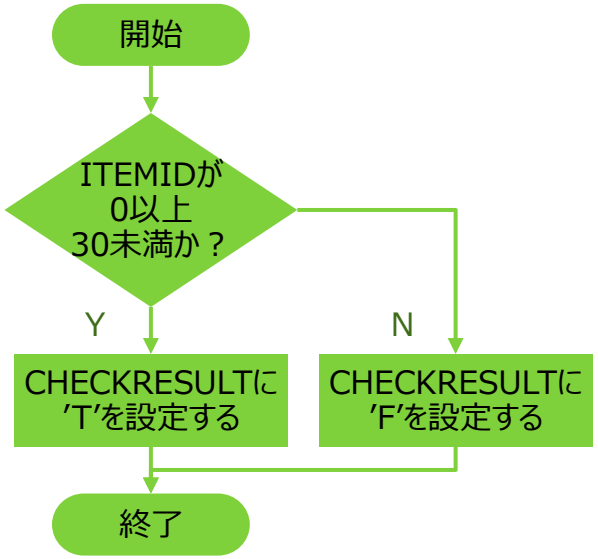
ゴールは
「単体テスト支援を用いて，単体テストの準備から実行，
バグの修正と再実行を実施し，全てのテストケースをOKとすること」
です。

2-2. テスト対象のプログラムとテストデータ

テスト対象のプログラムについて説明します。

機能仕様：
商品のID番号が有効かどうかを判定するプログラムです。

引数のITEMIDにID番号として2けたの数字を入力し
「0以上30未満なら有効」としてCHECKRESULTに'T'を設定します。
上記に当てはまらない場合，CHECKRESULTに'F'を設定します。



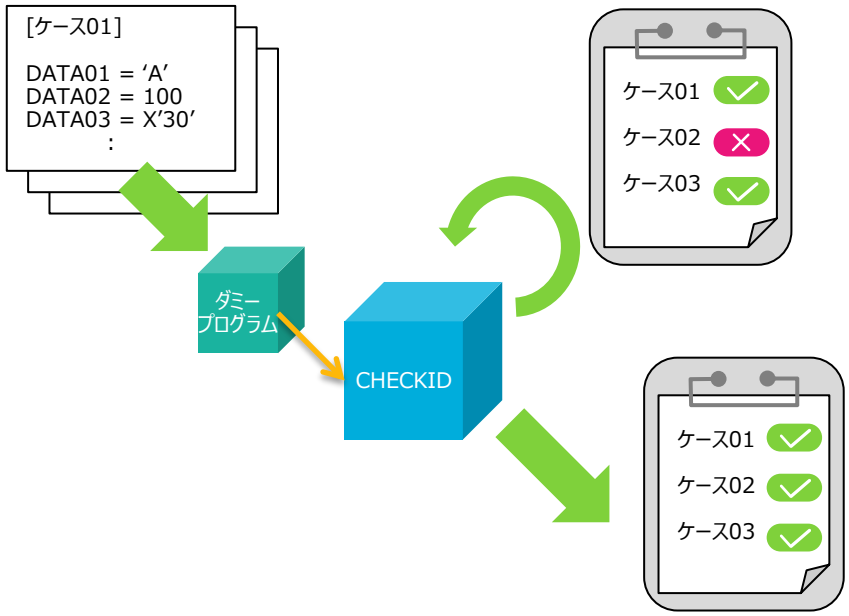
CHECKIDの処理フロー

テスト項目：
機能仕様の境界条件を元に，以下の組み合わせのテストを行います。

#	入力値(ITEMID)	期待値(CHECKRESULT)
1	0	'T'
2	29	'T'
3	30	'F'

2-3. 全体の流れ

大まかな手順です。



Point

テスト対象のプログラム"CHECKID"に
単体テストを実行し，バグを修正して
テストを完了する一連の流れを体験します。

- (0) サンプルコードの準備
- (1) テストプロジェクトの作成
- (2) テストケース作成
  - (2-1) テストケース1作成
  - (2-2) テストケース2作成
  - (2-3) テストケース3作成
- (3) テスト実行(NG)
- (4) 結果の確認およびバグ修正
  - (4-1) 結果の確認
  - (4-2) バグ修正
- (5) テスト実行(OK)

2-4. サンプルコードの準備

HITACHI
Inspire the Next

Progress

準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

サンプルコードの準備
サンプルコードのファイルを任意のフォルダに作成します。
以降は例として以下に置いたものとして説明します。

```
D:¥cblsample
```

準備するファイルは以下の1つです。

```
D:¥cblsample¥checkid.cbl
```

TIPS

サンプルコードは「付録1. サンプルコード」に記載しています。
テキストをコピー&ペーストして，サンプルコードのファイルを作成してください。

2-5. テストプロジェクトの作成

HITACHI
Inspire the Next

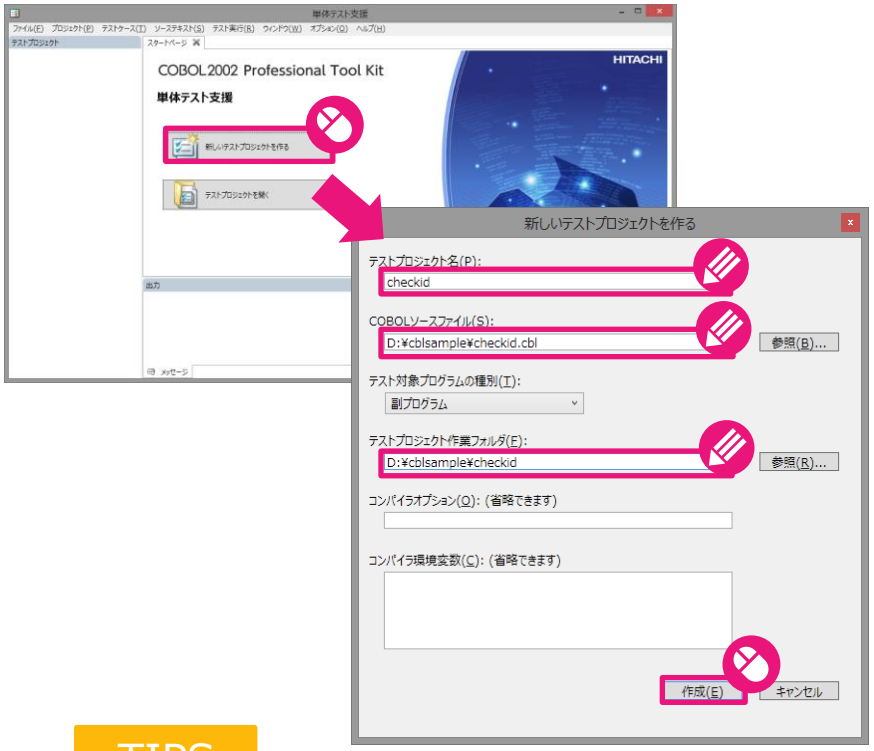
Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

単体テスト支援を起動します。
スタートメニューから「COBOL2002 Professional Tool Kit」→「単体テスト支援」をクリックします。

スタートページから「新しいプロジェクトを作る」をクリックします。

「新しいプロジェクトを作る」ダイアログで以下の値を入力して，「作成」をクリックします。

テストプロジェクト名
checkid
COBOLソースファイル
D:¥cblsample¥checkid.cbl
テストプロジェクト作業フォルダ
D:¥cblsample¥checkid



TIPS

テストプロジェクト作成時に，テストプロジェクト作業フォルダが存在しない場合には，作成するかどうかを問い合わせるダイアログが表示されるので，「OK」をクリックしてください。

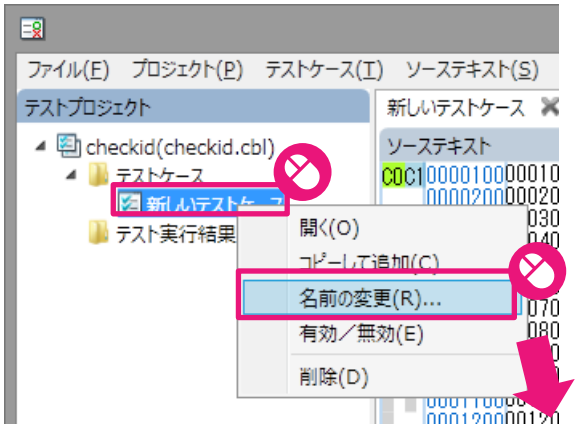
2-6. テストケース1作成(1)

HITACHI
Inspire the Next

Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

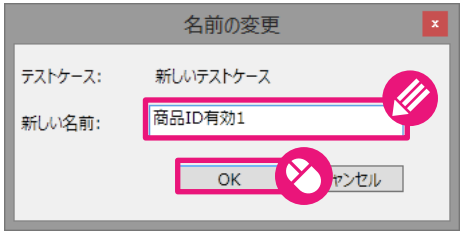
これからテストケースを合計3つ作成します。
まずは1つ目のテストケースを作成します。

画面左側の「テストプロジェクト」のツリーから”新しいテストケース”を右クリックして，右クリックメニューから「名前の変更」をクリックします。



「名前の変更」ダイアログで新しい名前に以下の値を入力して，「OK」をクリックします。

新しい名前
商品ID有効1



次にこのテストケースのテストデータを設定します。

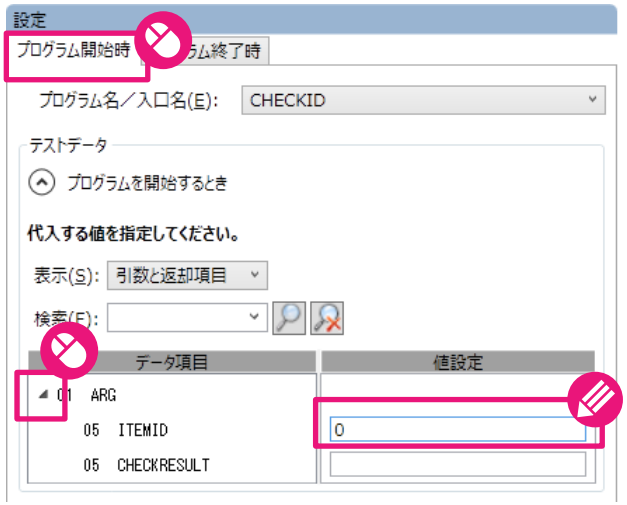
2-7. テストケース1作成(2)

HITACHI
Inspire the Next

Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

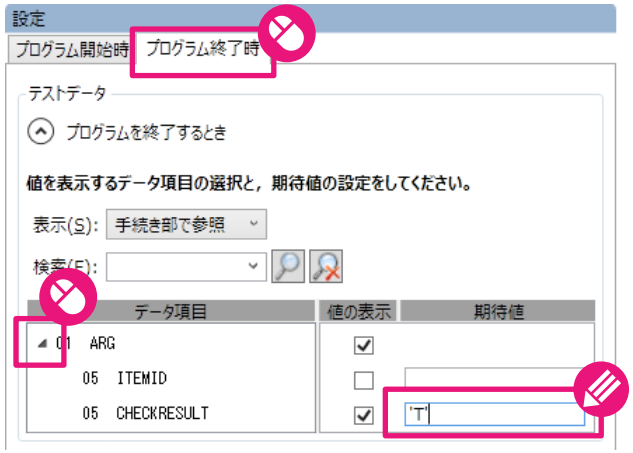
画面右側の「設定」の「プログラム開始時」タブを選び，データ項目のリストから以下の値を入力します。

ITEMIDの値設定
0



画面右側の「設定」の「プログラム終了時」タブを選び，データ項目のリストから以下の値を入力します。

CHECKRESULTの期待値設定
'T'



TIPS

テストデータとして文字を設定する場合には
シングルクォート(')で前後を囲んでください。
また，「期待値」に値を設定すると自動的に
「値の表示」にチェックが付きます。
期待値を判定するのにチェックは必要なので
そのままにしてください

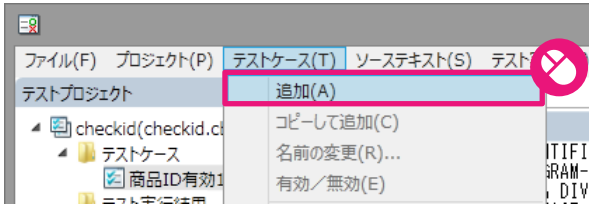
これでテストケース1つ目の作成が完了しました。

2-8. テストケース2作成(1)

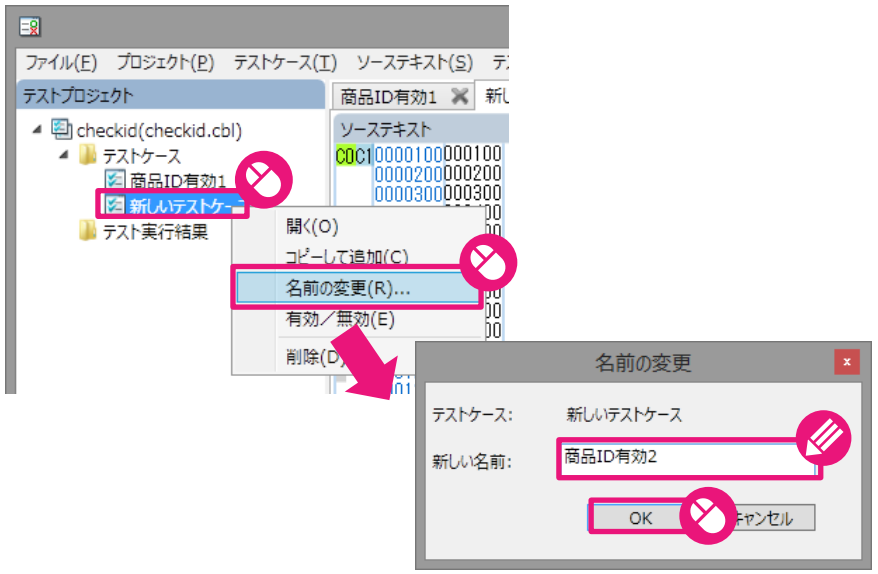
HITACHI
Inspire the Next

Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

2つ目のテストケースを作成します。
メニュー「テストケース」→「追加」をクリックします。
これで「テストプロジェクト」のテストケースのツリーに”新しいテストケース”が1つ追加されます。



画面左側の「テストプロジェクト」のツリーから”新しいテストケース”を右クリックして，右クリックメニューから「名前の変更」をクリックします。

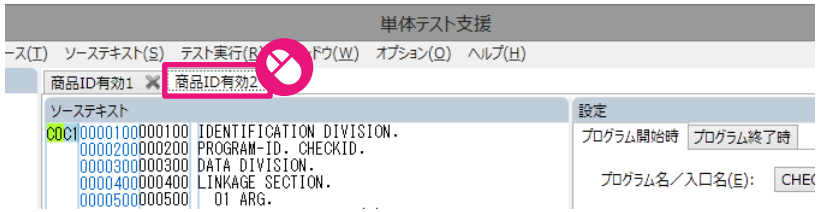


「名前の変更」ダイアログで新しい名前に以下の値を入力して，「OK」をクリックします。

新しい名前
商品ID有効2

次にこのテストケースのテストデータを設定します。

画面中央で”商品ID有効2”のテストケースのタブが選ばれていることを確認して次へ進んでください。



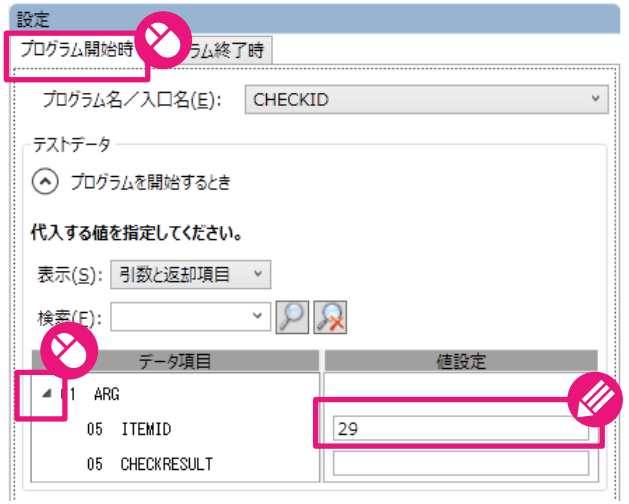
2-9. テストケース2作成(2)

HITACHI
Inspire the Next

Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

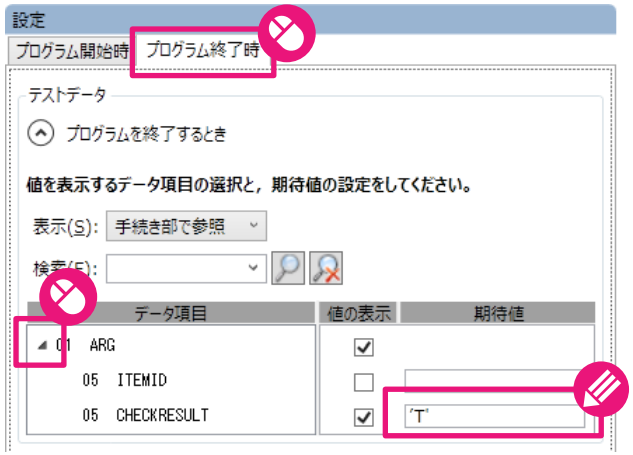
画面右側の「設定」の「プログラム開始時」タブを選び，データ項目のリストから以下の値を入力します。

ITEMIDの値設定
29



画面右側の「設定」の「プログラム終了時」タブを選び，データ項目のリストから以下の値を入力します。

CHECKRESULTの期待値設定
'T'



TIPS

テストデータとして文字を設定する場合には
シングルクォート(')で前後を囲んでください。
また，「期待値」に値を設定すると自動的に
「値の表示」にチェックが付きます。
期待値を判定するのにチェックは必要なので
そのままにしてください

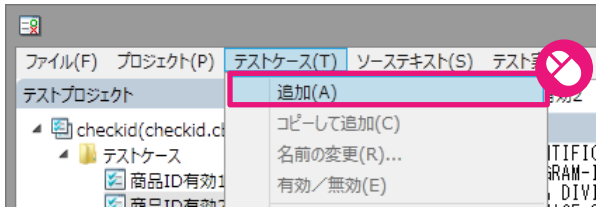
これでテストケース2つ目の作成が完了しました。

2-10. テストケース3作成(1)

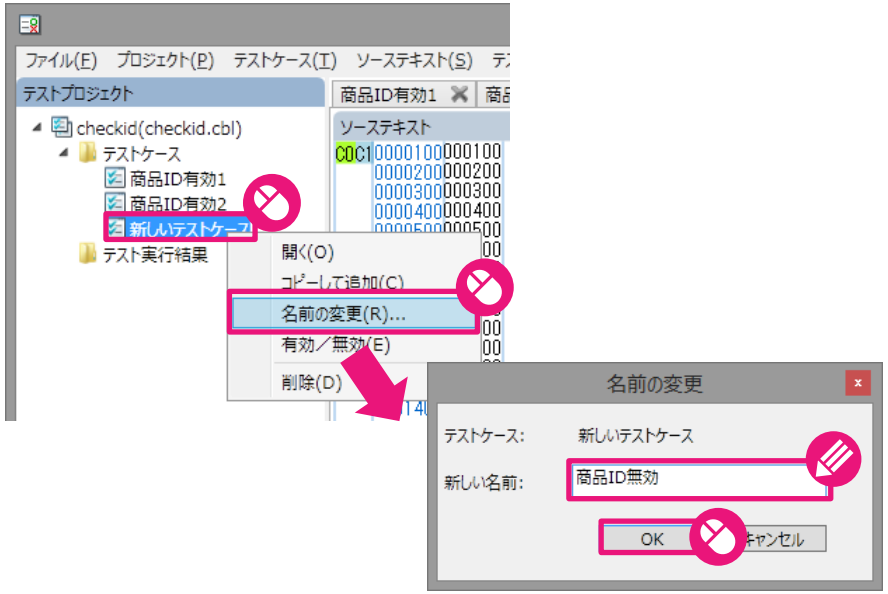
HITACHI
Inspire the Next

Progress: 準備 → Proj作成 → テストケース1作成 → テストケース2作成 → テストケース3作成 → 実行(NG) → 確認とバグ修正 → 実行(OK)

3つ目のテストケースを作成します。
メニュー「テストケース」→「追加」をクリックします。
これで「テストプロジェクト」のテストケースのツリーに”新しいテストケース”が1つ追加されます



画面左側の「テストプロジェクト」のツリーから”新しいテストケース”を右クリックして，右クリックメニューから「名前の変更」をクリックします。

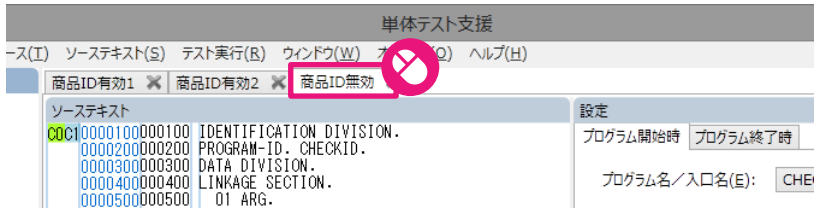


「名前の変更」ダイアログで新しい名前に以下の値を入力して，「OK」をクリックします。

新しい名前
商品ID無効

次にこのテストケースのテストデータを設定します。

画面中央で”商品ID無効”のテストケースのタブが選ばれていることを確認して次へ進んでください。



2-11. テストケース3作成(2)

HITACHI
Inspire the Next

Progress: 準備 → Proj作成 → テストケース1作成 → テストケース2作成 → テストケース3作成 → 実行(NG) → 確認とバグ修正 → 実行(OK)

画面右側の「設定」の「プログラム開始時」タブを選び，データ項目のリストから以下の値を入力します。

ITEMIDの値設定
30

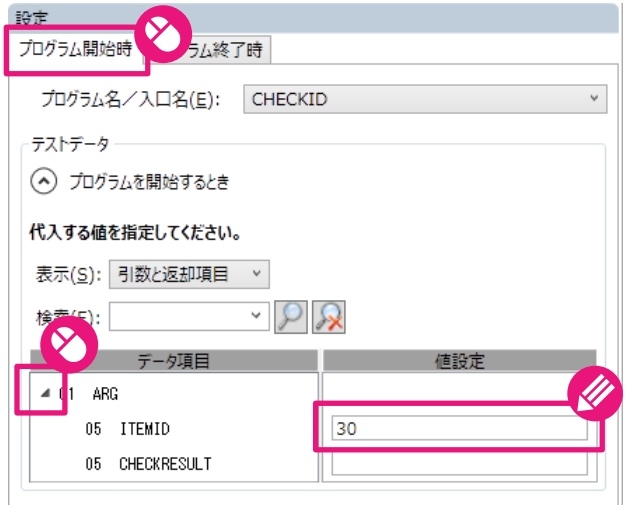
画面右側の「設定」の「プログラム終了時」タブを選び，データ項目のリストから以下の値を入力します。

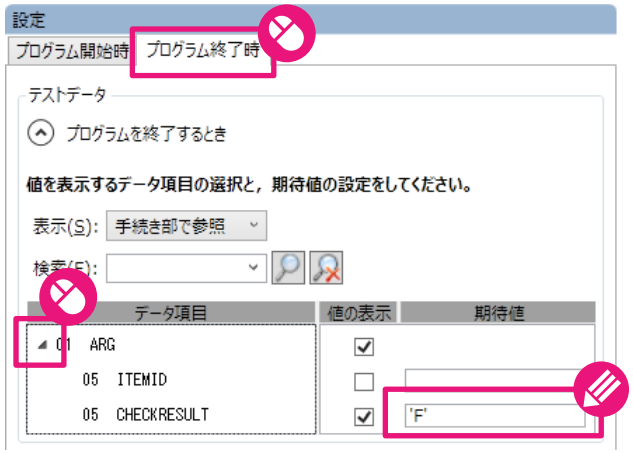
CHECKRESULTの期待値設定
'F'

TIPS

テストデータとして文字を設定する場合には
シングルクォート(')で前後を囲んでください。
また，「期待値」に値を設定すると自動的に
「値の表示」にチェックが付きます。
期待値を判定するのにチェックは必要なので
そのままにしてください

これでテストケース3つ目の作成が完了しました。





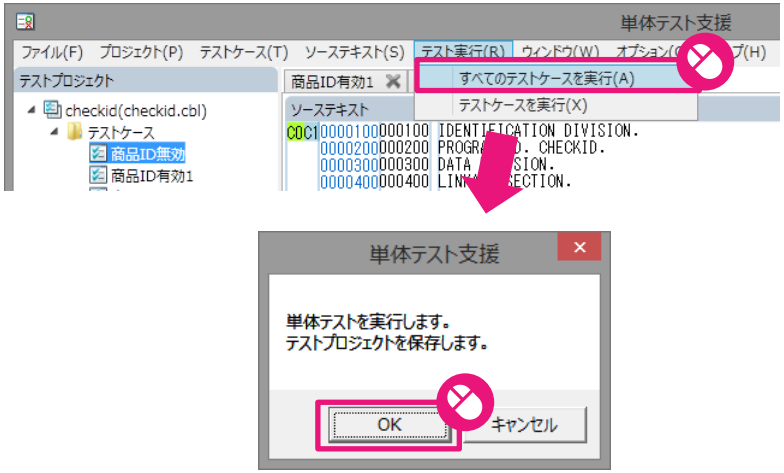
2-12. テスト実行(NG)

HITACHI
Inspire the Next

Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

単体テストを実行します。
メニューから「テスト実行」→「すべてのテストケースを実行」をクリックし，表示されるダイアログで「OK」をクリックします。

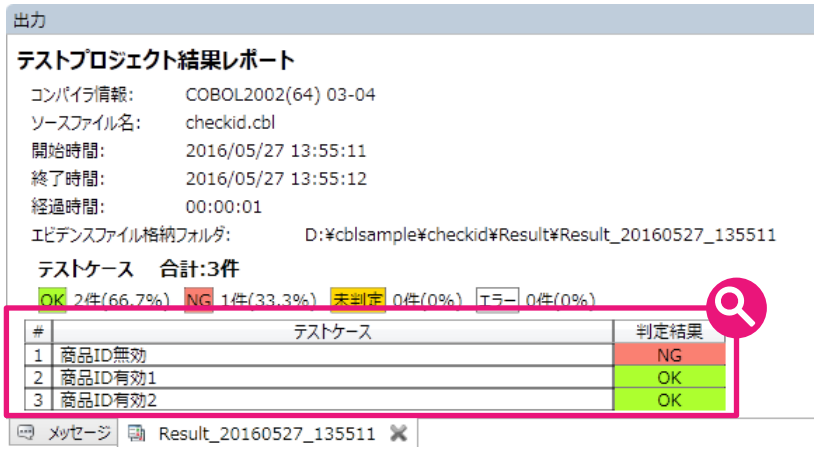
単体テストが実行されますので少し待ちます。



すべてのテストケースの実行が完了すると，画面下部の出力画面にテストプロジェクト結果レポートとして，実行した結果が一覧で表示されます。

3つのテストケースのうち”商品ID無効”の判定結果が”NG”と表示されます。
すなわち，このテストケースにおいて想定とは異なる動作となっていることがわかります。

プログラムのバグの可能性があるため，結果の詳細を確認します。

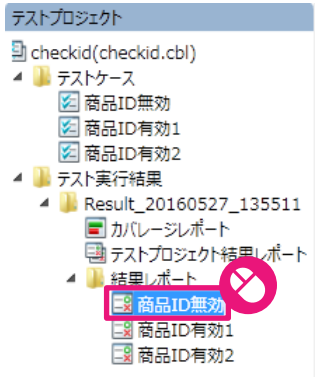


2-13. 結果の確認

HITACHI
Inspire the Next

Progress: 準備 | Proj作成 | テストケース1作成 | テストケース2作成 | テストケース3作成 | 実行(NG) | 確認とバグ修正 | 実行(OK)

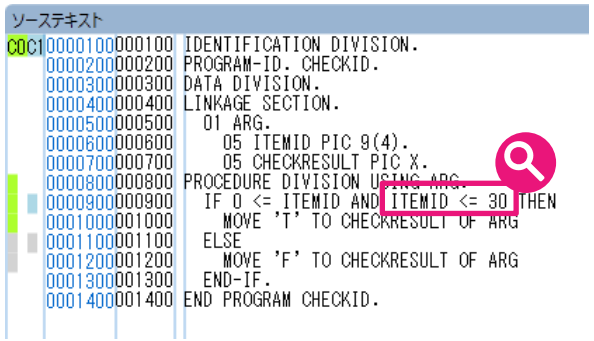
画面左側のテストプロジェクトツリーのテスト実行結果から，先ほど実行した結果のツリーを展開し，結果レポートの”商品ID無効”をダブルクリックします。



画面下部に”商品ID無効”のテストケース結果レポートが表示されます。
結果の値と判定結果の一覧を参照すると，期待値が’F’であるのに対して，実際の値は’T’が設定されていることが分かります。

値が異なるということは，IF文での判定が元々想定している仕様と異なっていることが考えられます。
実際に，ソーステキスト画面の9行目のIF文の条件文を見ると，「30未満(<30)」という仕様なのに「30以下(<=30)」となっています。

この条件文がバグであることが分かりましたので，この部分を修正します。

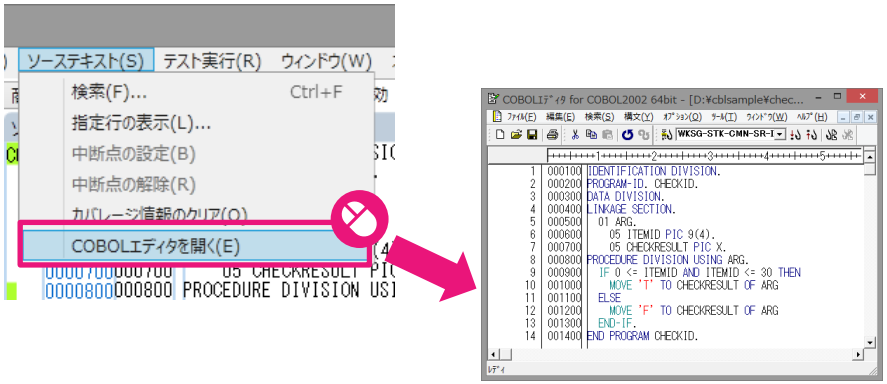


2-14. バグの修正

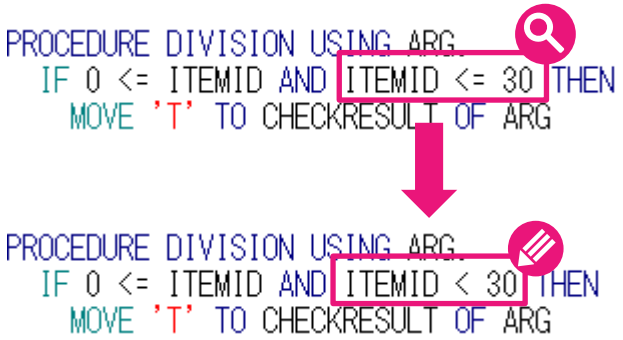
HITACHI
Inspire the Next

Progress: 準備 → Proj作成 → テストケース1作成 → テストケース2作成 → テストケース3作成 → 実行(NG) → 確認とバグ修正 → 実行(OK)

メニューから「ソーステキスト」→「COBOLエディタを開く」をクリックします。
別ウィンドウでCOBOLエディタが表示されます。

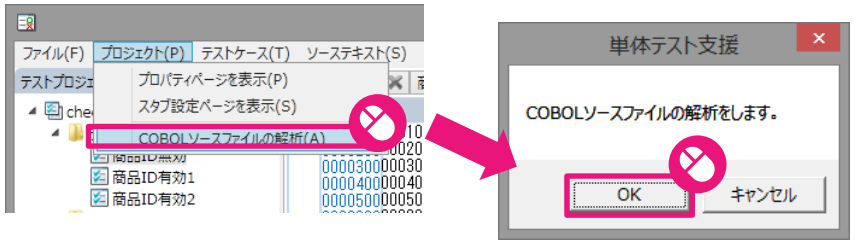


9行目の
”ITEMID <= 30 THEN”
の記載を
“ITEMID < 30 THEN”
に修正します。



COBOLエディタのメニューから「ファイル」→「上書き保存」をクリックして，ファイルを保存し，COBOLエディタを閉じます。

単体テスト支援のメニューから「プロジェクト」→「COBOLソースファイルの解析」をクリックし，表示されるダイアログで「OK」をクリックします。



これで，バグの修正がテストプロジェクトに反映されました。それでは再度テストの実行を行います。

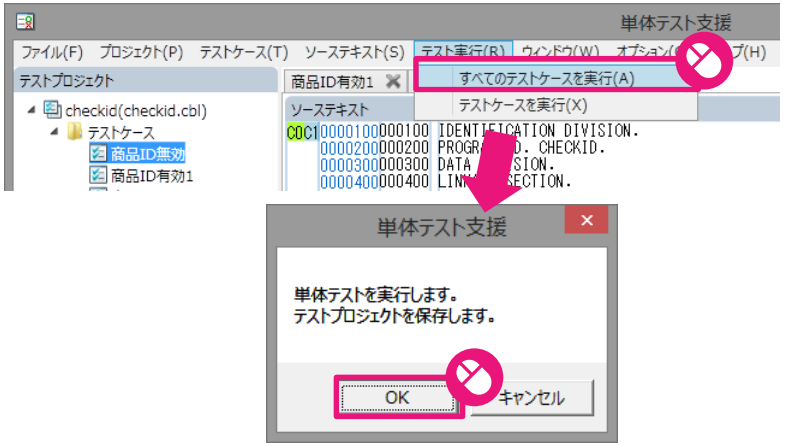
2-15. テスト実行(OK)

HITACHI
Inspire the Next

Progress: 準備 → Proj作成 → テストケース1作成 → テストケース2作成 → テストケース3作成 → 実行(NG) → 確認とバグ修正 → 実行(OK)

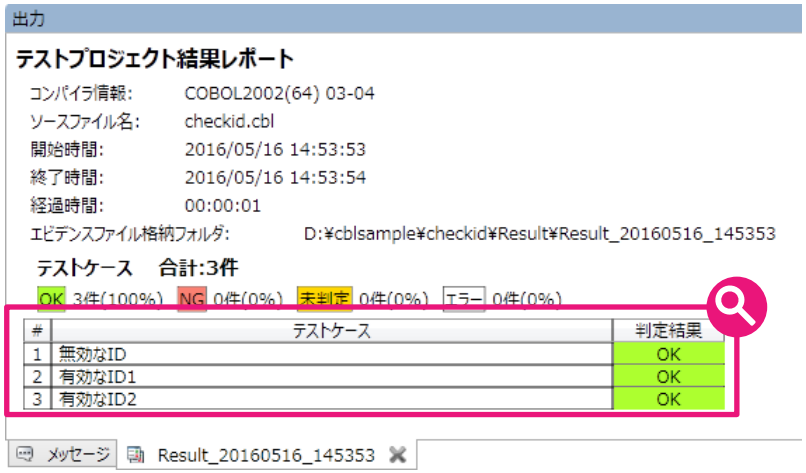
単体テストを実行します。
メニューから「テスト実行」→「すべてのテストケースを実行」をクリックし，表示されるダイアログで「OK」をクリックします。

単体テストが実行されますので，少し待ちます。



すべてのテストケースの実行が完了すると，画面下部の出力画面にテストプロジェクト結果レポートとして，実行した結果が一覧で表示されます。

バグを修正したことにより，3つのテストケースすべてが”OK”となりました。

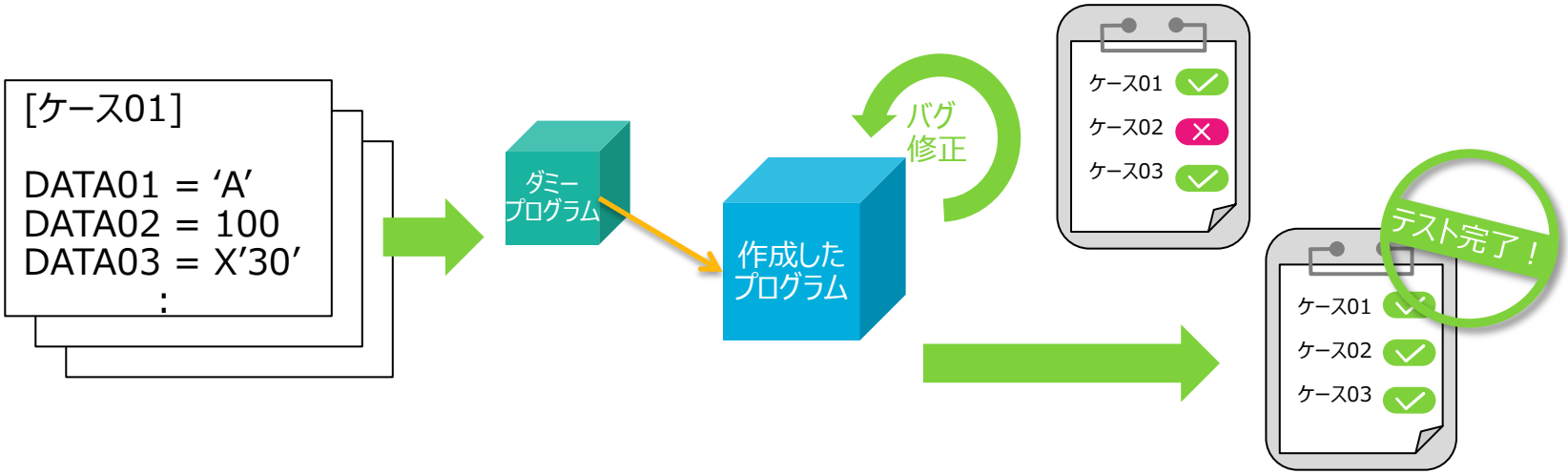


必要な組み合わせのテストを全て実施しました。
これでこの"CHECKID"プログラムに対する単体テストは完了です！

HITACHI
Inspire the Next

3. まとめ

3-1. 体験内容のまとめ



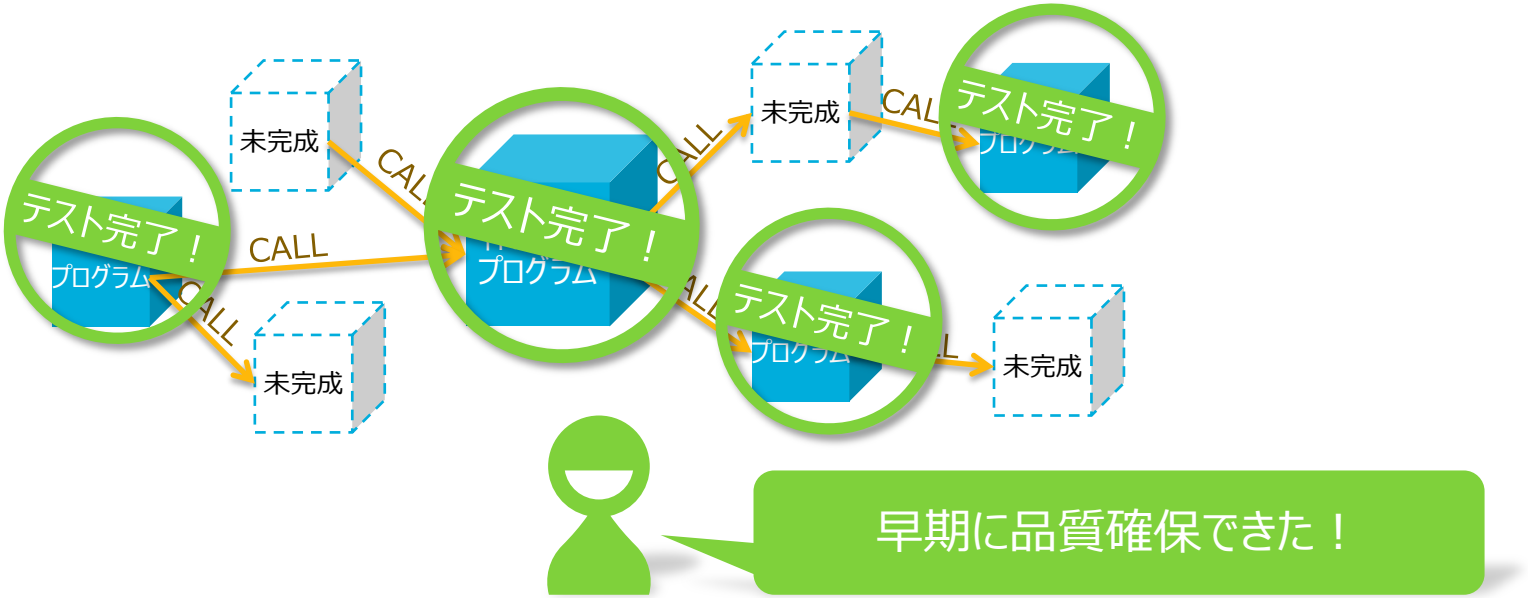
Goal

単体テスト支援による単体テストの基本的な流れを体験しました。

単体テストを実行することで，バグの発見やその修正をすぐにでき品質向上に大きく貢献します。

3-2. まとめ

単体テスト支援を活用することで単体テストの準備や実行を簡単に行えます。
これにより完成したプログラムからテストを実施でき，品質の早期確保ができます。



実際のシステムの場合，ファイルアクセスを含むなどより複雑なプログラムとなります。
今回の体験内容ではシンプルなプログラムの場合の例でしたが
それらに対応する機能も単体テスト支援は用意しています。
それらの機能の詳細についてはCOBOL活用ガイドのトレーニングカテゴリの資料で
より詳しく説明しますので，ご参照ください。

3-3. 補足

HITACHI
Inspire the Next

■ 他の活用ガイドの紹介
COBOL2002活用ガイドシリーズは随時拡充/更新する予定です。
詳細はCOBOL2002のWebサイトまたは問い合わせください。

■ お問い合わせ先
URL:http://www.hitachi.co.jp/soft/cobol/

付録1. サンプルコード

HITACHI
Inspire the Next

checkid.cbl

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. CHECKID.
000300 DATA DIVISION.
000400 LINKAGE SECTION.
000500   01 ARG.
000600     05 ITEMID PIC 9(4).
000700     05 CHECKRESULT PIC X.
000800 PROCEDURE DIVISION USING ARG.
000900   IF 0 <= ITEMID AND ITEMID <= 30 THEN
001000     MOVE 'T' TO CHECKRESULT OF ARG
001100   ELSE
001200     MOVE 'F' TO CHECKRESULT OF ARG
001300   END-IF.
001400 END PROGRAM CHECKID.
```

商標類など

■　商標類
HITACHIは株式会社 日立製作所の商標または登録商標です。
その他記載の会社名，製品名などは，それぞれの会社の商標もしくは登録商標です。