



## 単体テスト支援 テストケースのコピーを利用したテストデータ作成手順

COBOL2002 Developer Professional 単体テスト支援

株式会社 日立製作所

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

---

本資料は「COBOL活用ガイド 単体テスト支援ファーストステップガイド」の内容を前提としています。

「単体テスト支援 テストケースのコピーを利用したテストデータ作成手順」ではテストケースをコピーして効率よく多数のテストケースを作成する手順を演習形式で紹介します。

演習内容は以下です。

- ◆ 単体テスト支援のGUIを用いたテストケースのコピー
- ◆ テストケースCSVファイルを用いたテストケースのコピー

演習にかかる時間の目安は約50分程度です。

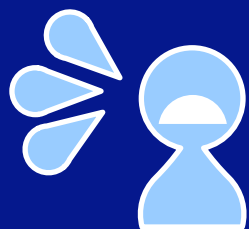
---

## Contents

1. 単体テスト支援
2. テストケースをコピーしてテストパターンを増やしてみよう
3. テストケースCSVを用いてテストパターンを増やしてみよう
4. まとめ

# 1. 単体テスト支援





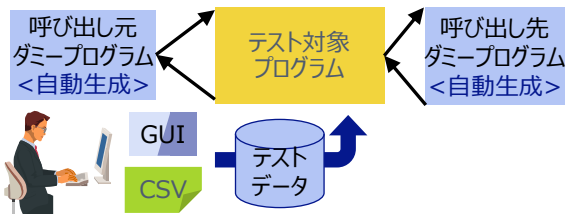
バグを早期に発見したい！

手作業のテストではもう対応できない！

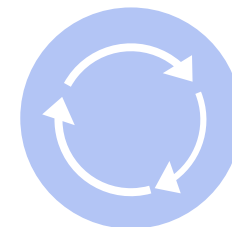
そのお悩み 単体テスト支援が解決します！



準備が簡単に！



結果確認が簡単に！

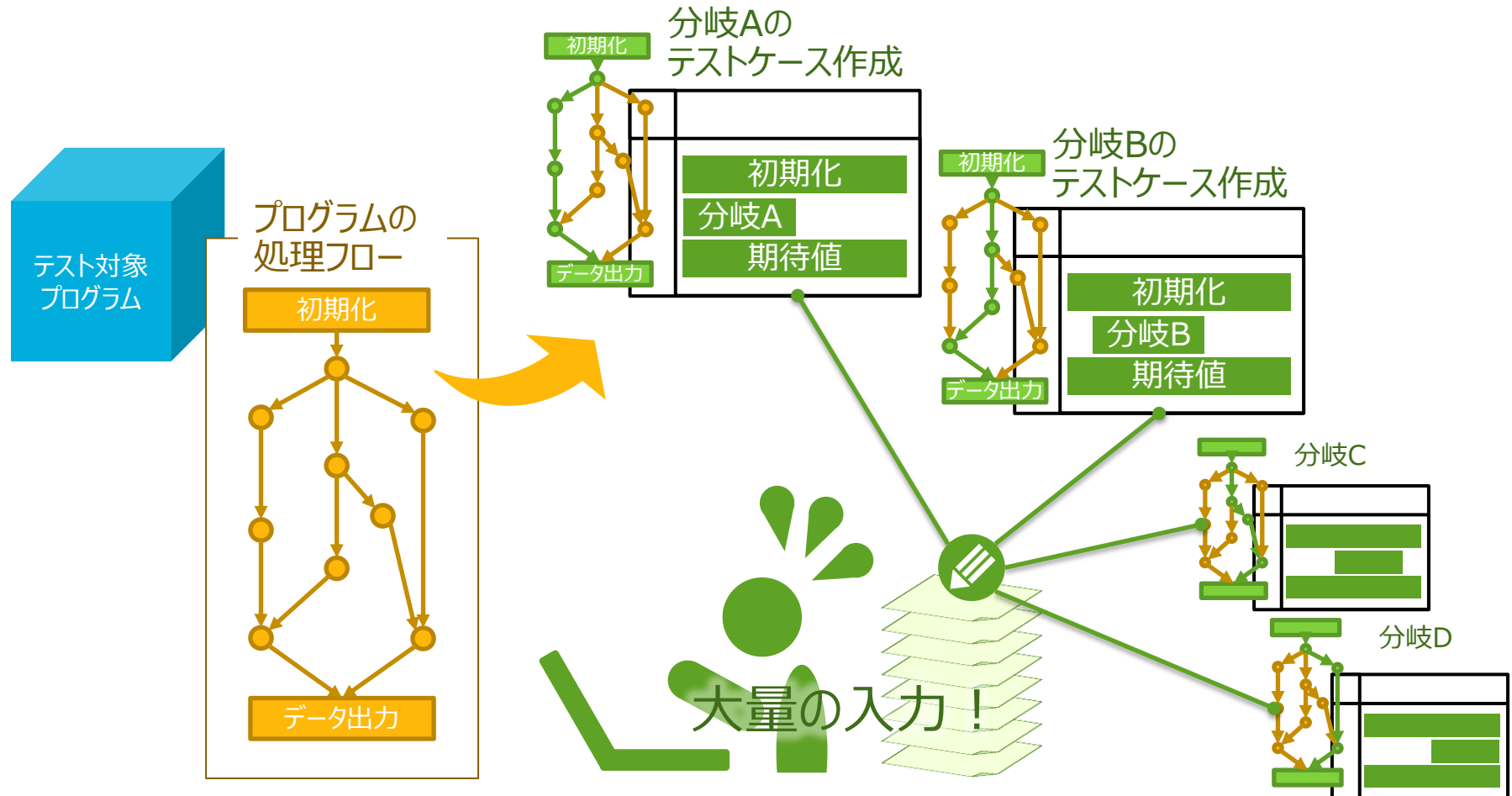


繰り返しテストが簡単に！



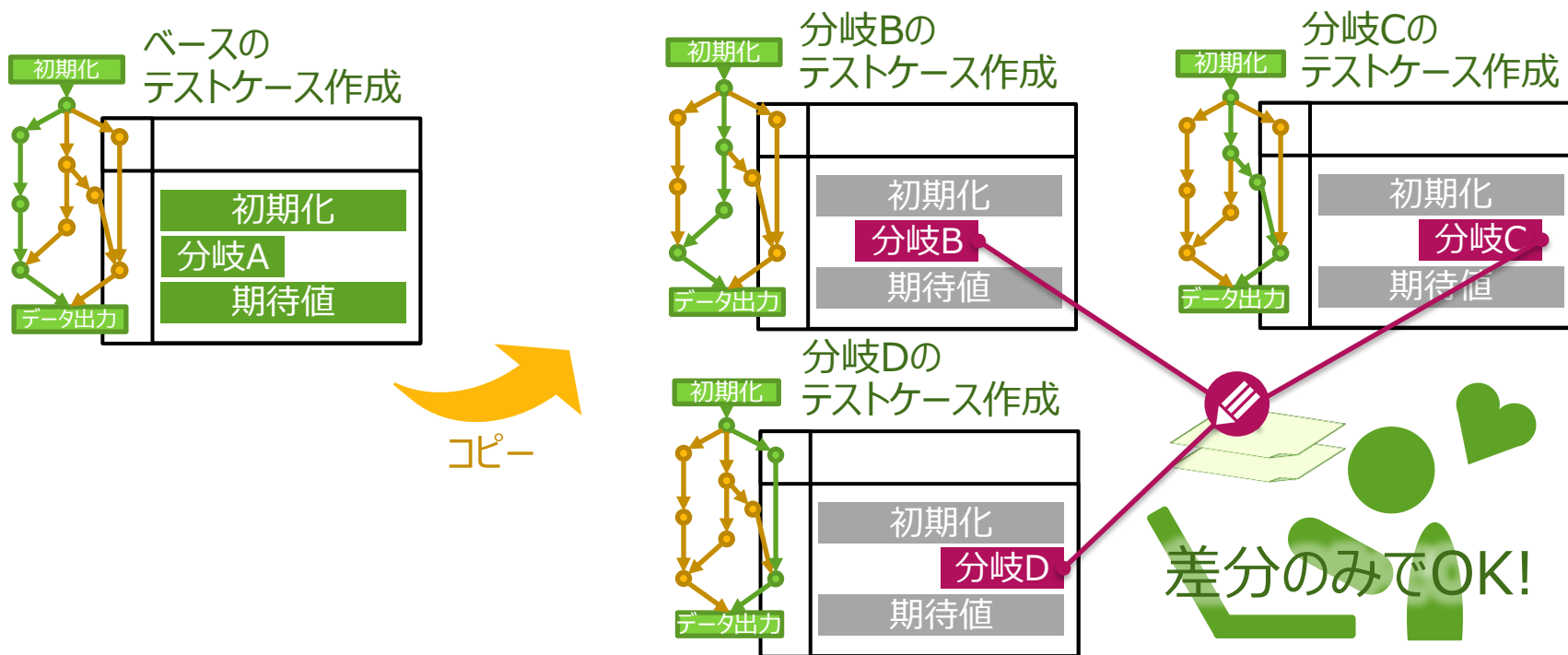
# 1-2. 多数のテストケース作成の課題

分岐処理などを多く含むプログラムをテストする場合に  
そのプログラムの処理を網羅するように**多数のテストパターン**を用意する必要があります。  
それらのテストパターンを1つ1つ手作業で設定するのは**非常に手間**がかかります。



# 1-3. 効率良いテストケースの作成

複数のテストパターンで共通のテストデータの値を利用するケースは少なくありません。これらのデータについては1つ1つ設定するのではなく、1つのテストケースに設定した後にそれをコピーすることで効率よくテストデータの設定やテストケースの作成を行うことができます。

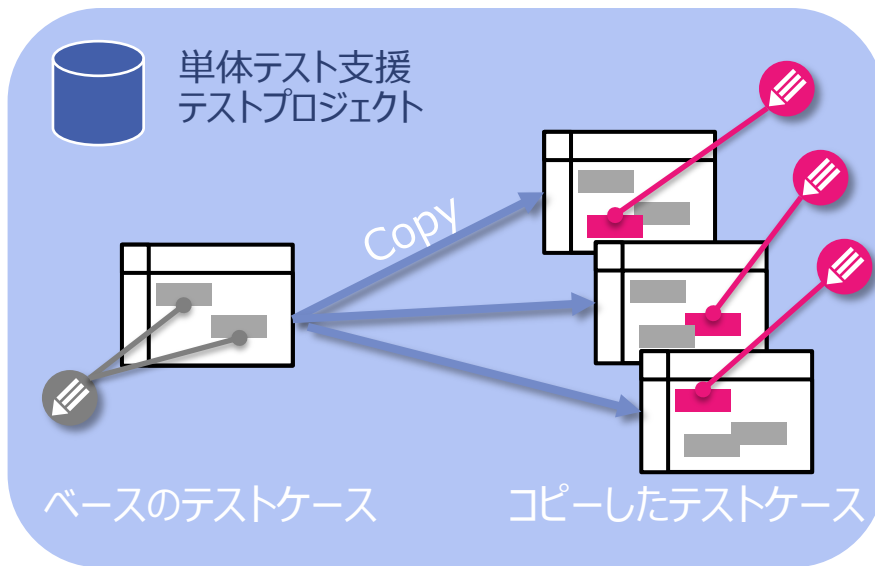


# 1-4. テストケースのコピーによるテストデータ作成方法

単体テスト支援でテストケースのコピーを用いたテストデータ作成方法には2つあります。

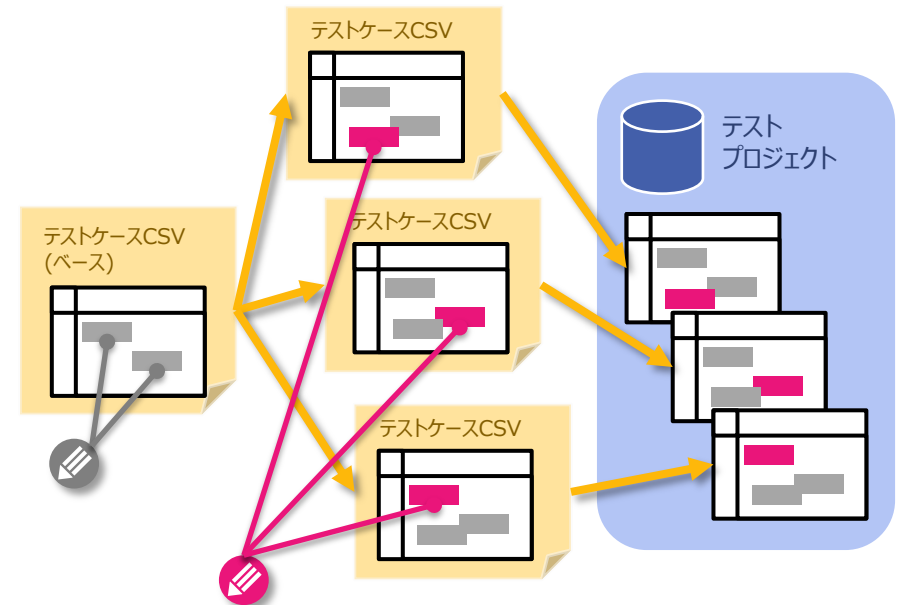
## GUIでテストケースのコピー

単体テスト支援のGUIで素早く簡単にコピー  
少量のデータのコピーや編集に向いています



## テストケースCSVファイルのコピー

テストケースCSVファイルを用いてテストデータを作成  
大量のデータを機械的に生成する場合などに向いています



以降では単体テスト支援でどのような手順で実現するかを演習形式で説明します。

演習は大きく2部に分かれています。

演習1ではGUIでテストケースのコピー機能を利用する方法を説明します。

演習2ではテストケースCSVファイルを利用する方法を説明します。



## 2. テストケースをコピーして テストパターンを増やしてみよう

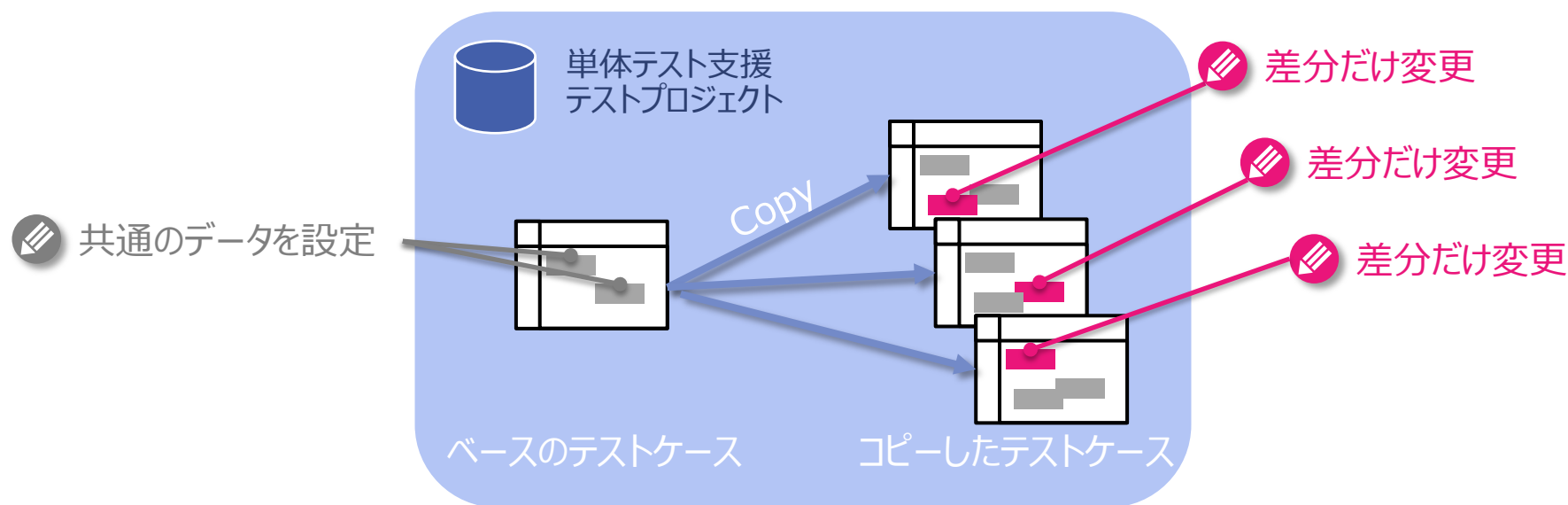


## 2-1. 演習する内容(演習1)

テストケースをコピーしてテストパターンを増やす演習を行います。

演習1では単体テスト支援のGUIでベースとなるテストケースを作成し、そのテストケースをコピーして一部のテストデータを編集することでテストパターンを増やすことを行います。

演習1のゴールは「テストパターンをコピーを利用して増やし、カバレッジのC0メジャー(命令網羅)を100%とする単体テストを実施する」です。

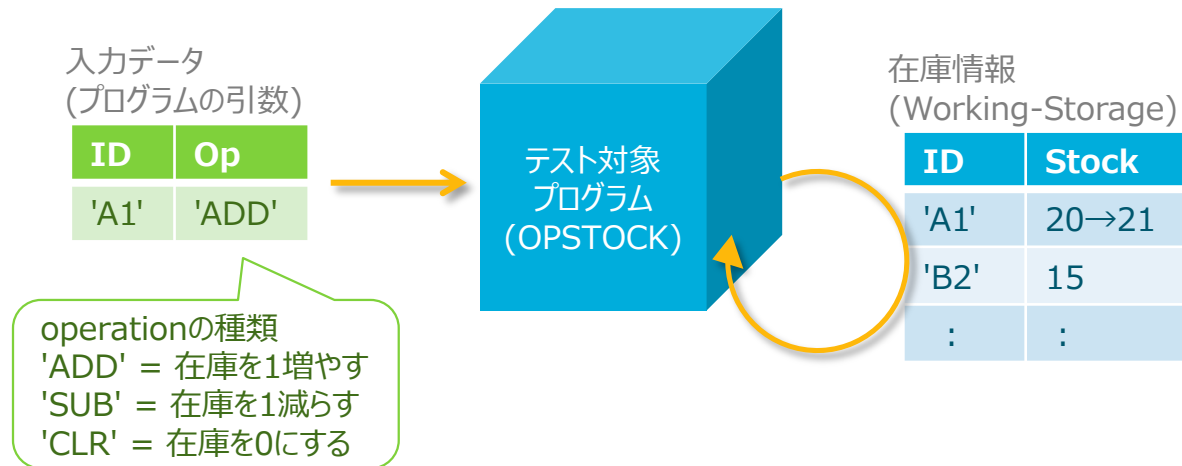


テスト対象のプログラム[OPSTOCK]について説明します。  
(演習1および演習2で共通です)

機能仕様：

商品の在庫を操作するプログラムです。

プログラムでは在庫情報として商品IDとその個数を持ちます。  
プログラムの引数から入力データとして商品IDと操作の種類を受け取り、  
それに合わせて在庫情報を更新します。



## 2-3. 作成するテストケース

演習1では3つのテストケースを作成します。

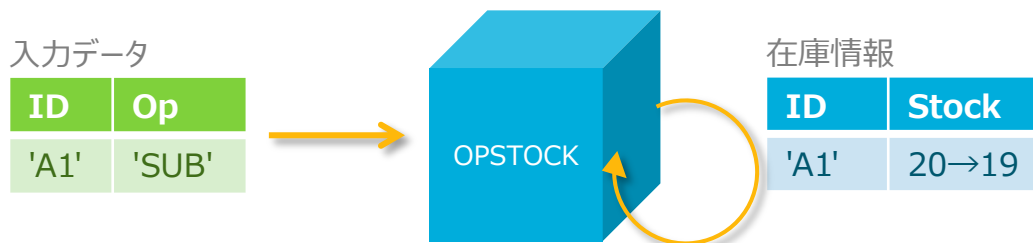
### 演習1テストケース1 [ADDのテスト]

商品ID'A1'に対して操作'ADD'を実行した時のテストです。  
在庫情報のStockを期待値として確認します。



### 演習1テストケース2 [SUBのテスト]

商品ID'A1'に対して操作'SUB'を実行した時のテストです。  
在庫情報のStockを期待値として確認します。

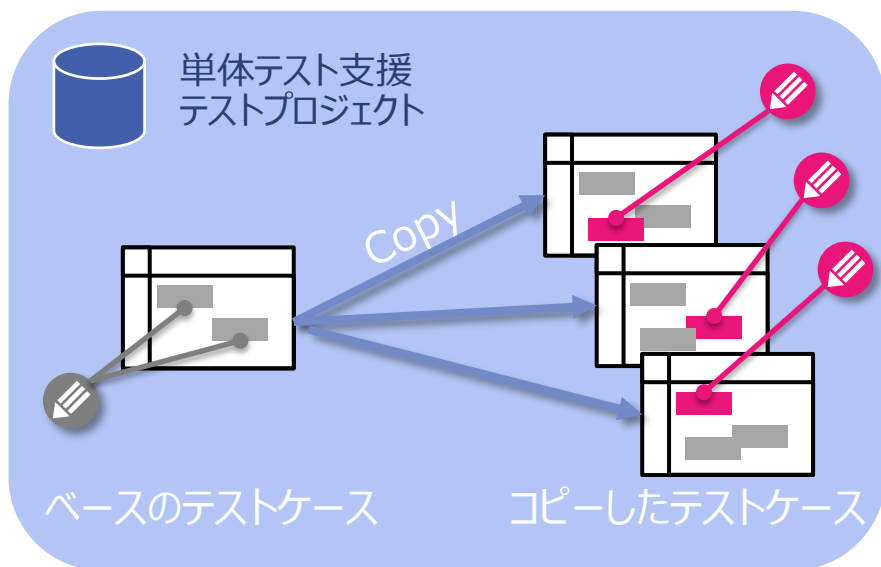


### 演習1テストケース3 [CLRのテスト]

商品ID'A1'に対して操作'CLR'を実行した時のテストです。  
在庫情報のStockを期待値として確認します。



演習の手順です。



準備

サンプルコードの準備

テストプロジェクトの作成

1つ目のテストケースを作成

2つ目のテストケースを作成

3つ目のテストケースを作成

テストの実行

テスト結果の確認

テストケースごとの結果確認

カバレッジの結果確認

Progress

準備

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

結果確認

### 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¥opstock.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。

## 2-6. テストプロジェクトの作成

Progress

準備

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

結果確認

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

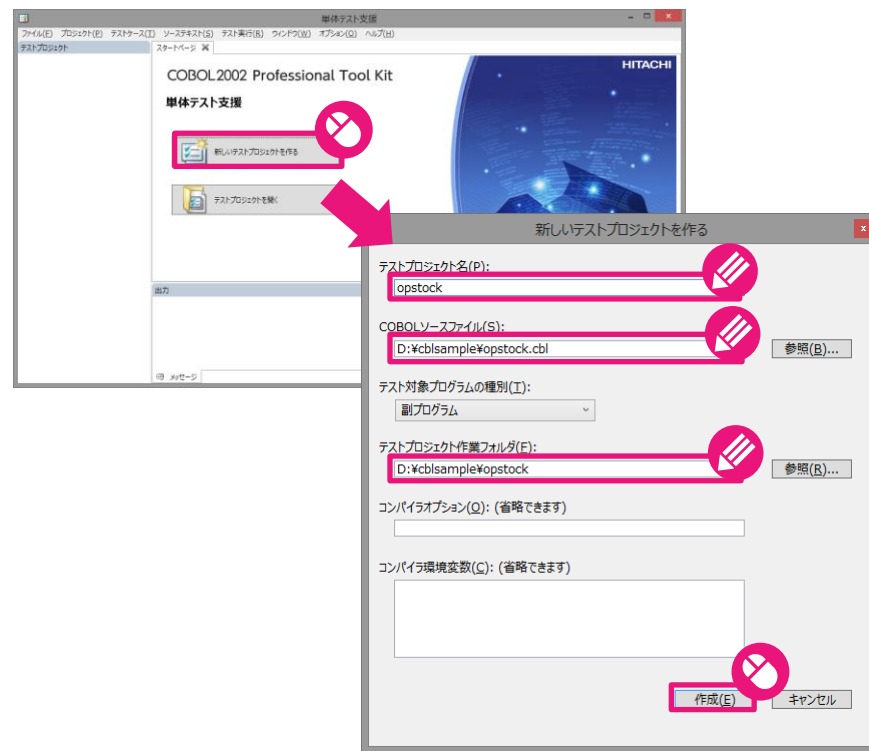
opstock

COBOLソースファイル

D:\¥cb1sample¥opstock.cbl

テストプロジェクト作業フォルダ

D:\¥cb1sample¥opstock



### 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。

## 2-7. 1つ目のテストケースを作成

Progress

準備

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

結果確認

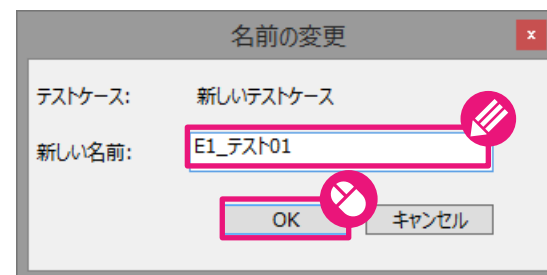
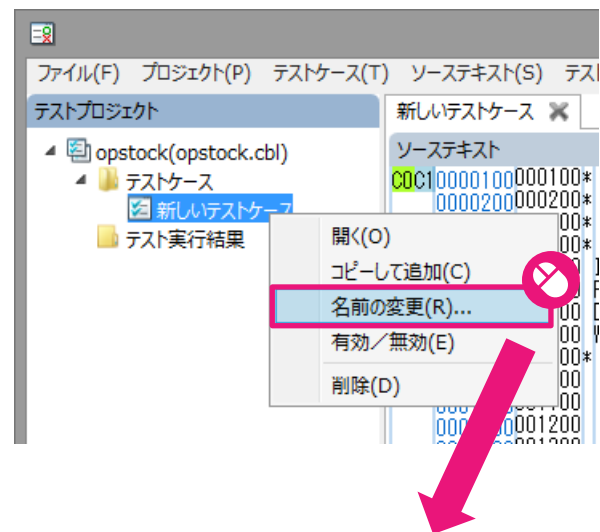
これからテストケースを3つ作成します。  
1つ目のテストケースでは初期値を含めて  
全てのテストデータを設定します。

画面左側の「テストプロジェクト」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

E1\_テスト01





## 2-8. 1つ目のテストケースを作成

Progress

準備

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

結果確認

次にこのテストケースにテストデータを設定します。  
このテストケースでは初期値の在庫情報および  
入力データ、期待値を設定します。

まず初期値の在庫情報のテストデータを設定しま  
す。

画面右側の「設定」で「プログラム開始時」タブが  
選ばれていることを確認します。

次にテストデータの「表示」のプルダウンメニューを開  
き、「手続き部で参照」を選択します。

データ項目のツリーから"STOCKITEM"を展開し、  
以下3つのデータ項目に値を設定します。

ITEMCOUNT

1

ITEMID OF ITEM(1)

'A1'

STOCK OF ITEM(1)

20

設定

プログラム開始時

プログラム名/入口名(E): OPSTOCK

テストデータ

プログラムを開始するとき

代入する値を指定してください。

表示(S): 引数と返却項目

検索(F): すべて

手続き部で参照

引数と返却項目

値設定

01 設定

プログラム開始時

代入する値を指定してください。

表示(S): 手続き部で参照

検索(E):

データ項目	値設定
01 STOCKITEMS	
05 ITEMCOUNT	1
05 ITEM	
05 ITEM(1)	
10 ITEMID	'A1'
10 STOCK	20
05 ITEM(2)	
05 ITEM(3)	

## 2-9. 1つ目のテストケースを作成

Progress

準備

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

結果確認

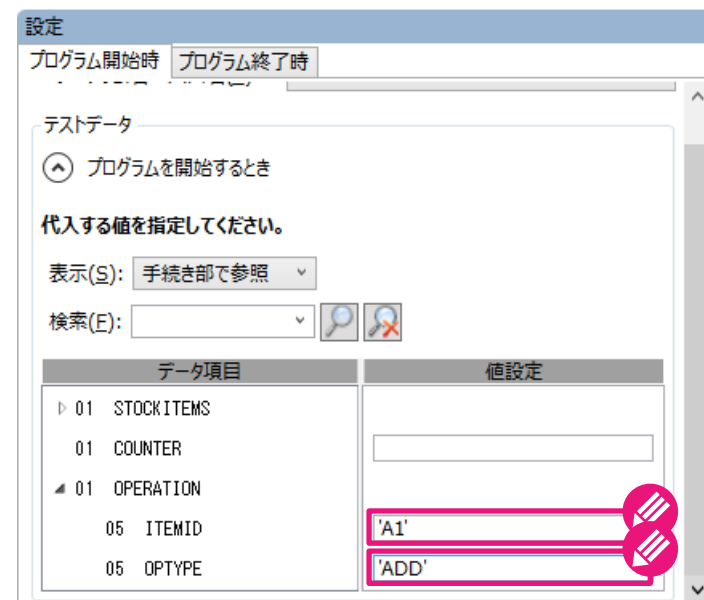
次にプログラム引数の入力データのテストデータを設定します。  
データ項目のツリーから"OPERATION"を展開し、  
以下2つのデータ項目に値を設定します。

ITEMID

'A1'

OPTYPE

'ADD'



## 2-10. 1つ目のテストケースを作成

Progress

準備

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

結果確認

期待値を設定します。  
画面右側の「設定」で「プログラム終了時」タブをクリックします。

データ項目のツリーから"STOCKITEMS"を展開し、以下のデータ項目に値を設定します。

STOCK OF ITEM(1)

21

これで1つ目のテストケースへのテストデータ設定は完了です。

2つ目、3つ目のテストケースはこの1つ目のテストケースをコピーして作成します。

設定

プログラム開始時 プログラム終了時

テストデータ

プログラムを終了するとき

値を表示するデータ項目の選択と、期待値の設定をしてください。

表示(S): 手続き部で参照

検索(E):

データ項目	値の表示	期待値
▷ 01 STOCKITEMS	<input type="checkbox"/>	
▷ 01 COUNTER	<input type="checkbox"/>	
▷ 01 TRATI	<input type="checkbox"/>	

設定

プログラム開始時 プログラム終了時

値を表示するデータ項目の選択と、期待値の設定をしてください。

表示(S): 手続き部で参照

検索(E):

データ項目	値の表示	期待値
◀ 01 STOCKITEMS	<input checked="" type="checkbox"/>	
05 ITEMCOUNT	<input type="checkbox"/>	
◀ 05 ITEM	<input checked="" type="checkbox"/>	
◀ 05 ITEM(1)	<input checked="" type="checkbox"/>	
10 ITEMID	<input type="checkbox"/>	
10 STOCK	<input checked="" type="checkbox"/>	21
▷ 05 ITEM(2)	<input type="checkbox"/>	
▷ 05 ITEM(3)	<input type="checkbox"/>	

## 2-11. 2つ目のテストケースを作成

Progress

準備

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

結果確認

2つ目のテストケースを作成します。  
1つ目のテストケースをコピーして、必要な部分のみテストデータの値を変更します。

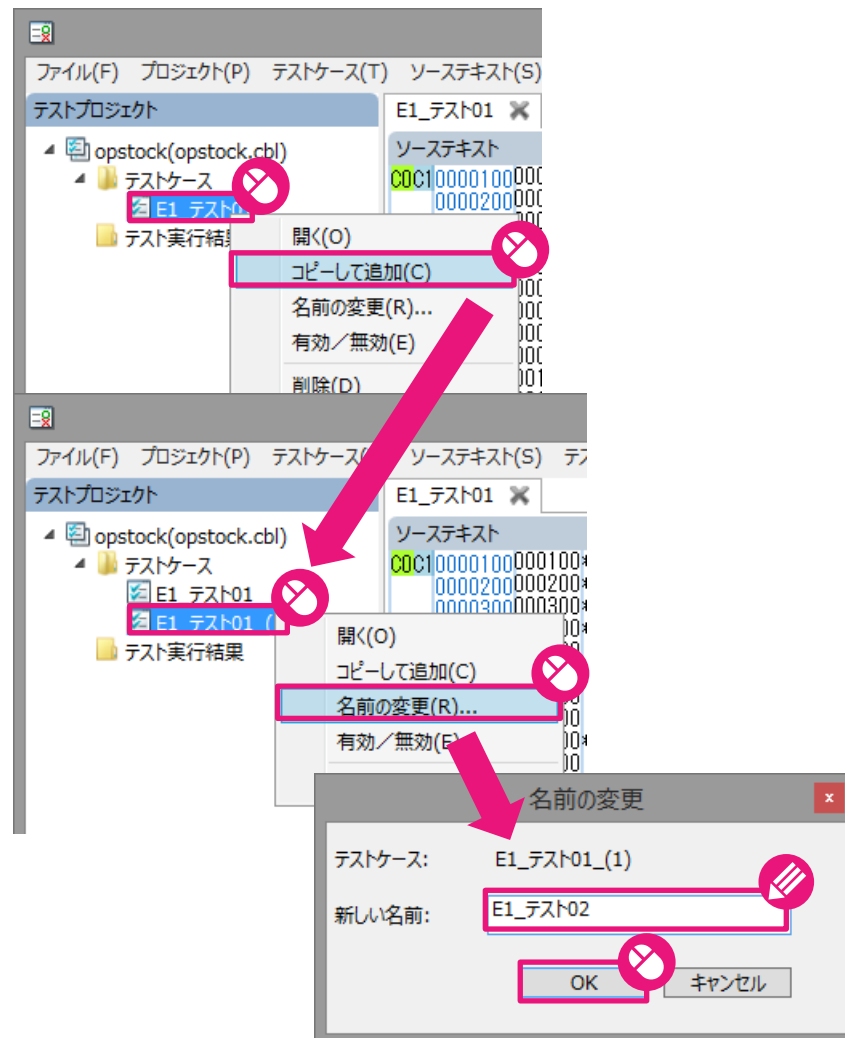
「テストプロジェクト」ツリーから"E1\_テスト01"を右クリックして、右クリックメニューから「コピーして追加」をクリックします。  
これで「テストプロジェクト」ツリーのテストケースの下に"E1\_テスト01\_(1)"が追加されます。

"E1\_テスト01\_(1)"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

E1\_テスト02



## 2-12. 2つ目のテストケースを作成

Progress

準備

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

結果確認

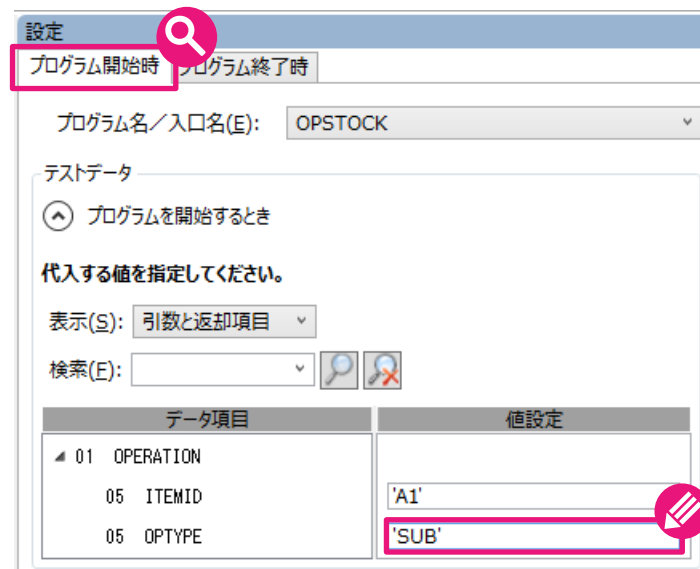
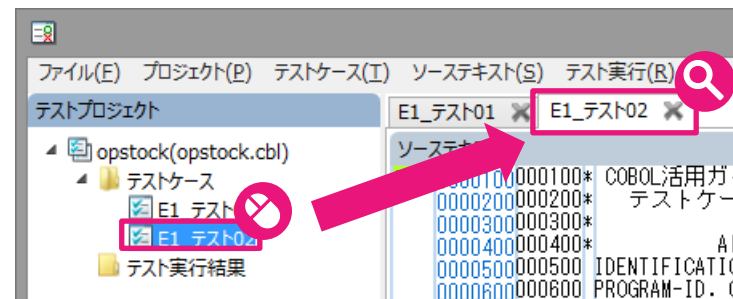
コピーして作成した2つ目のテストケースのテストデータの値を変更します。  
変更するテストデータは、プログラム引数の入力データと期待値です。

「テストプロジェクト」ツリーから"E1\_テスト02"をダブルクリックして"E1\_テスト02"タブを表示します。

プログラム引数の入力データの値を変更します。  
画面右側の「設定」の「プログラム開始時」タブが選ばれていることを確認し、データ項目のツリーから"OPERATION"を展開し、以下のデータ項目の値を変更します。

OPTYPE

'SUB'



## 2-13. 2つ目のテストケースを作成

Progress

準備

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

結果確認

次に期待値の値を変更します。  
画面右側の「設定」から「プログラム終了時」タブを  
選択し、データ項目のツリーから  
"STOCKITEMS"を展開し、以下のデータ項目  
の値を変更します。

ITEMID OF ITEM(1)

19

これで2つ目のテストケースのテストデータ設定は完  
了です。

設定

プログラム開始時 **プログラム終了時**

値を表示するデータ項目の選択と、期待値の設定をしてください。

表示(S): 手続き部で参照

検索(E):

データ項目	値の表示	期待値
▲ 01 STOCKITEMS	<input checked="" type="checkbox"/>	
05 ITEMCOUNT	<input type="checkbox"/>	
▲ 05 ITEM	<input checked="" type="checkbox"/>	
▲ 05 ITEM(1)	<input checked="" type="checkbox"/>	
10 ITEMID	<input type="checkbox"/>	
10 STOCK	<input checked="" type="checkbox"/>	19
▷ 05 ITEM(2)	<input type="checkbox"/>	
▷ 05 ITEM(3)	<input type="checkbox"/>	

## 2-14. 3つ目のテストケースを作成

Progress

準備

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

結果確認

3つ目のテストケースを作成します。  
1つ目のテストケースをコピーして、必要な部分のみテストデータの値を変更します。

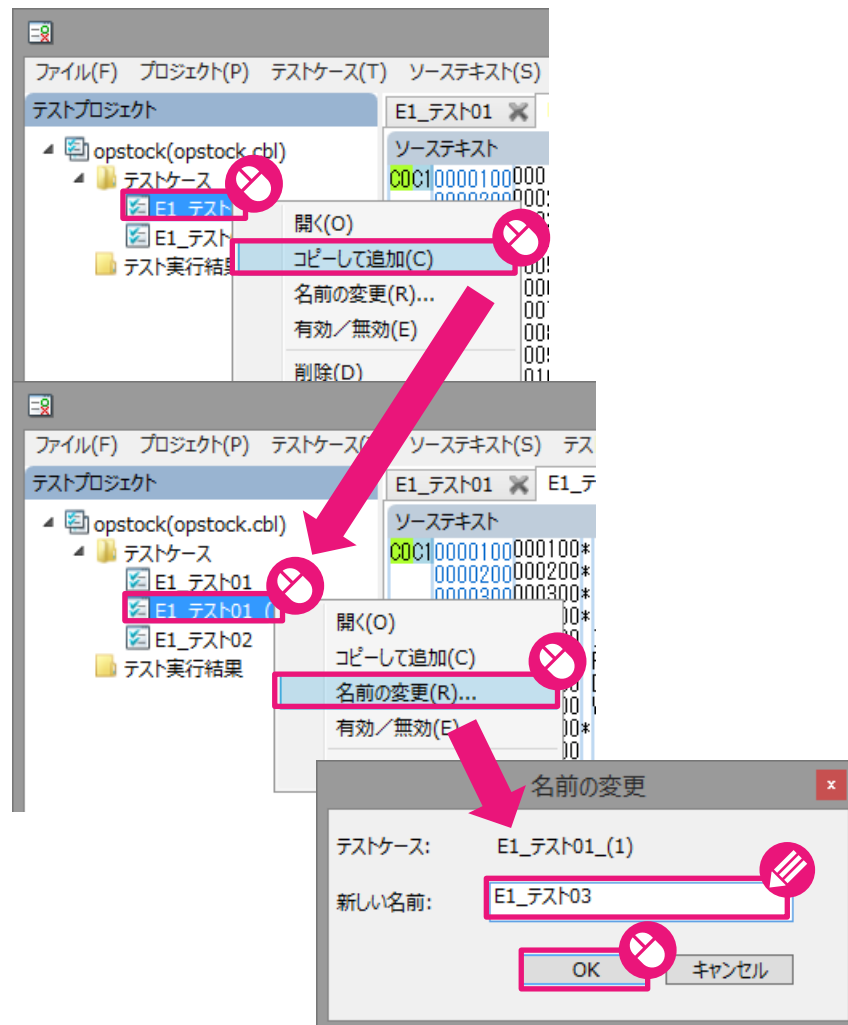
「テストプロジェクト」ツリーから"E1\_テスト01"を右クリックして、右クリックメニューから「コピーして追加」をクリックします。  
これで「テストプロジェクト」ツリーのテストケースの下に"E1\_テスト01\_(1)"が追加されます。

"E1\_テスト01\_(1)"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

E1\_テスト03



## 2-15. 3つ目のテストケースを作成

Progress

準備

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

結果確認

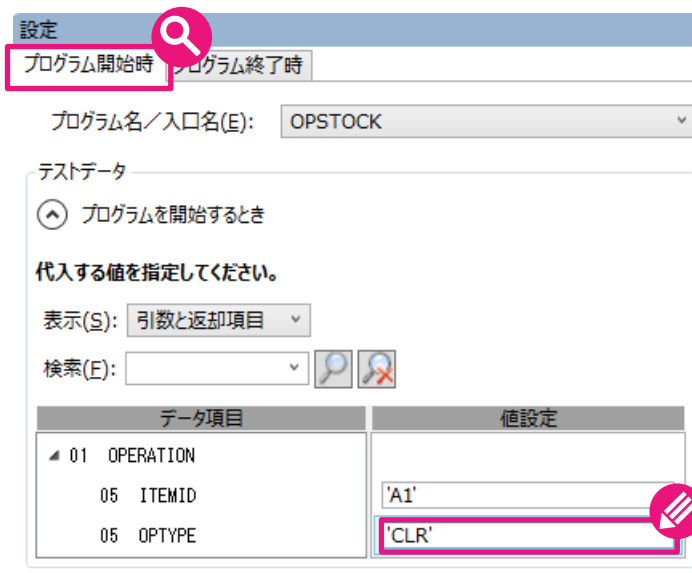
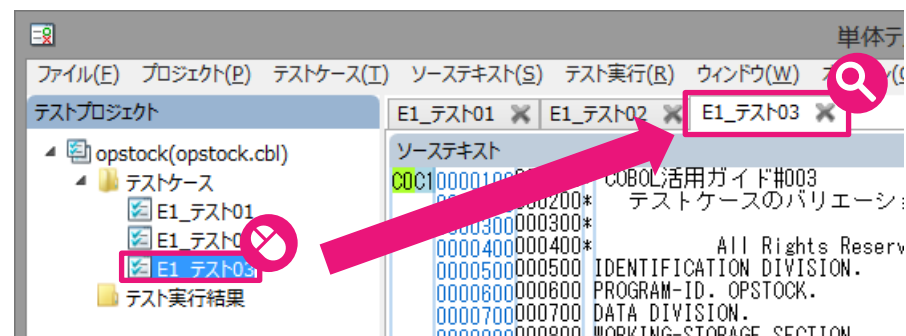
コピーして作成した3つ目のテストケースのテストデータの値を変更します。  
変更するテストデータは、プログラム引数の入力データと期待値です。

「テストプロジェクト」ツリーから"E1\_テスト03"をダブルクリックして"E1\_テスト03"タブを表示します。

プログラム引数の入力データの値を変更します。  
画面右側の「設定」の「プログラム開始時」タブが選ばれていることを確認し、データ項目のツリーから"OPERATION"を展開し、以下のデータ項目の値を変更します。

OPTYPE

'CLR'





## 2-16. 3つ目のテストケースを作成

Progress

準備

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

結果確認

次に期待値の値を変更します。  
画面右側の「設定」から「プログラム終了時」タブを  
選択し、データ項目のツリーから  
"STOCKITEMS"を展開し、以下のデータ項目  
の値を変更します。

ITEMID OF ITEM(1)

0

これで3つ目のテストケースのテストデータ設定は完了です。

設定

プログラム開始時 **プログラム終了時**

値を表示するデータ項目の選択と、期待値の設定をしてください。

表示(S): 手続き部で参照

検索(E):

データ項目	値の表示	期待値
01 STOCKITEMS	<input checked="" type="checkbox"/>	
05 ITEMCOUNT	<input type="checkbox"/>	
05 ITEM	<input checked="" type="checkbox"/>	
05 ITEM(1)	<input checked="" type="checkbox"/>	
10 ITEMID	<input type="checkbox"/>	
10 STOCK	<input checked="" type="checkbox"/>	0
05 ITEM(2)	<input type="checkbox"/>	
05 ITEM(3)	<input type="checkbox"/>	

## 2-17. テストの実行

Progress

準備

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

結果確認

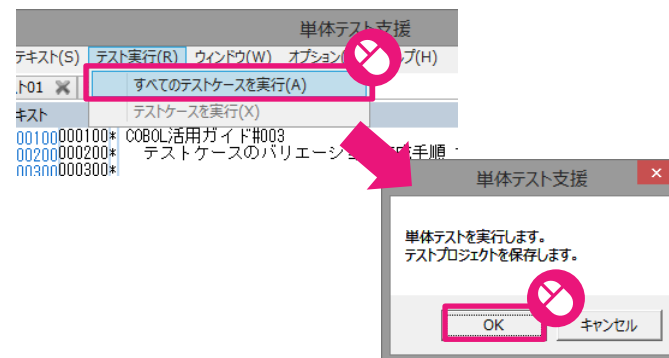
単体テストを実行します。  
メニューから「テスト実行」→「すべてのテストケース  
を実行」をクリックし、表示されるダイアログで  
「OK」をクリックします。

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が  
表示されます。  
このレポートには実行したテストケースの結果が一  
覧で表示されます。

判定結果がすべて"OK"となりテスト完了です。

テスト自体はこれで完了ですが、実行したテスト  
ケースの詳細結果を見てみましょう。



出力

### テストプロジェクト結果レポート

コンパイラ情報: COBOL2002(64) 03-04  
ソースファイル名: opstock.cbl  
開始時間: 2016/09/02 13:45:16  
終了時間: 2016/09/02 13:45:16  
経過時間: 00:00:00  
エビデンスファイル格納フォルダ: D:\%cblsample%\opstock%\Result%\Result\_20160902\_134515

テストケース 合計:3件  
OK 3件(100%) NG 0件(0%) 未判定 0件(0%) エラー 0件(0%)

#	テストケース	判定結果
1	E1_テスト01	OK
2	E1_テスト02	OK
3	E1_テスト03	OK

メッセージ Result\_20160902\_134515

## 2-18. テスト結果の詳細確認

Progress

準備

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

結果確認

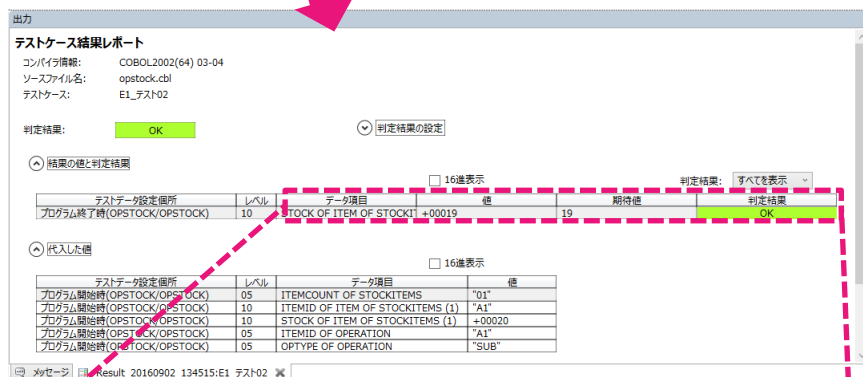
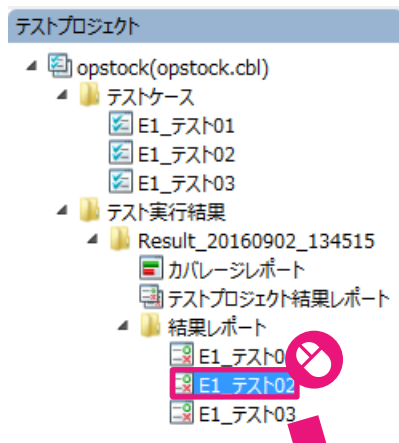
ここで詳細なテスト結果の確認としてテストケース"E1\_テスト02"とカバレッジの情報を確認します。

まずテストケース"E1\_テスト02"についての詳細結果を確認します。  
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て結果レポートの"E1\_テスト02"をダブルクリックします。

画面下部に"E1\_テスト\_02"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欄には、在庫から1つ減らした('SUB')結果, 19になってOKとなっていることがわかります。  
また、在庫情報などの初期値もテストケースのコピーによって設定されていることも確認してみてください。

### TIPS

ここでは"E1\_テスト02"について結果の確認を行いました。同様に"E1\_テスト01"や"E1\_テスト03"についても確認してみてください。



データ項目	値	期待値	判定結果
STOCK OF ITEM OF STOCKIT	+00019	19	OK

Progress

準備

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

結果確認

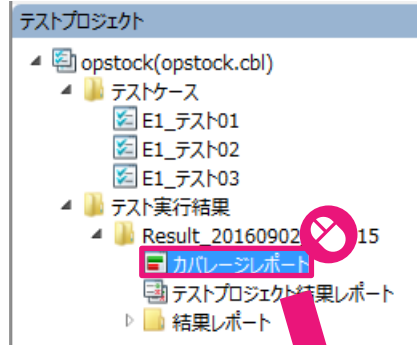
次にカバレッジの結果を確認します。  
画面左側の「テストプロジェクト」ツリーのテスト実行結果から、先ほど実行した結果のツリーを展開してカバレッジレポートをダブルクリックします。

画面下部にカバレッジレポートが表示されます。  
C0メジャーが100%で、C1メジャーが75%となっていることが確認できます。  
この演習ではC0を100%とするテストデータを設定しました。C1を100%とするテストデータの作成にもチャレンジしてみてください。

テスト完了!

## TIPS

C1は分岐網羅であり、100%にするには全ての分岐を通るテストデータが必要となります。  
サンプルコードのIF文とEVALUATE文では、演習のテストデータだけでは分岐を全て通っていないパターンが存在します。それらに対応するテストケースを追加することでC1も100%とすることが可能です。  
答えは「付録3. C1メジャー100%にするには」を参照してください。



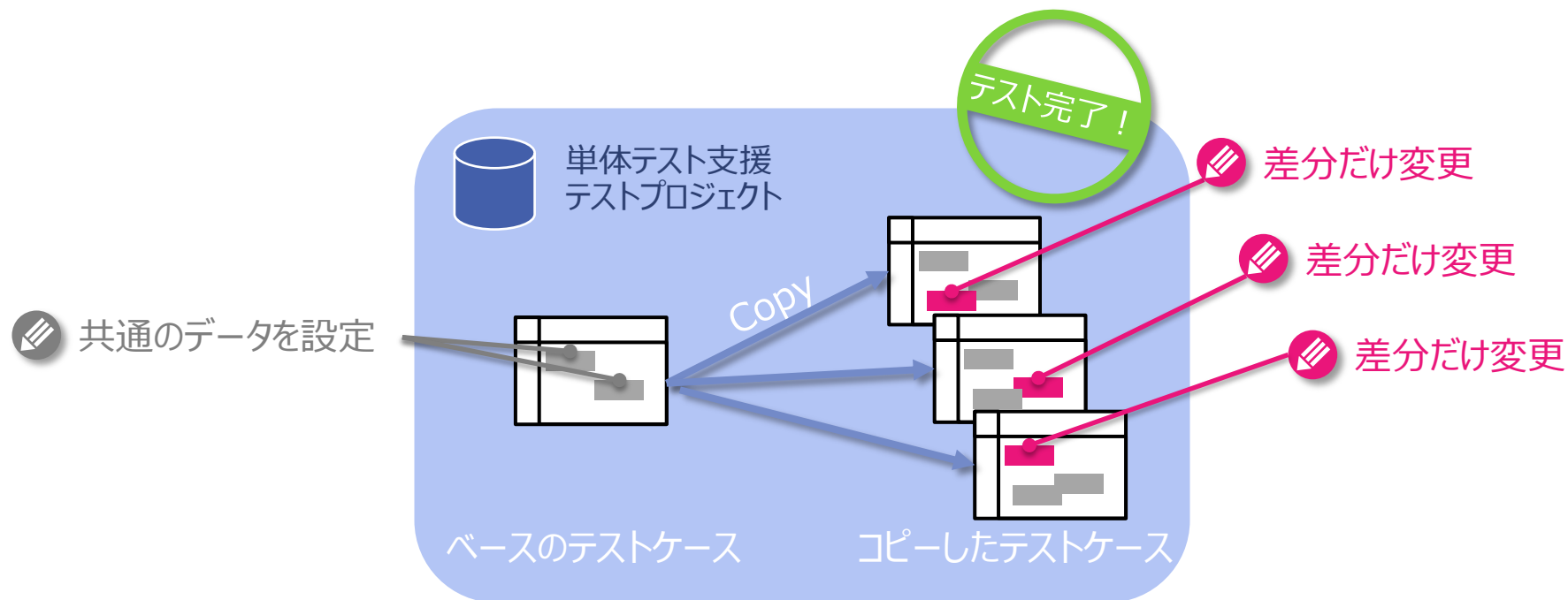
出力

### カバレッジレポート

コンパイラ情報: COBOL2002(64) 03-04  
ソースファイル名: opstock.cbl

種類	カバレッジ
C0メジャー	100.0%
C1メジャー	75.0%
S1メジャー	---

メッセージ Result\_20160902\_134515



単体テスト支援のGUIでベースのテストケース(E1\_テスト01)を作成し、それをコピーして差分だけを変更してテストケースを増やす(E1\_テスト02, E1\_テスト03)演習を行いました。

各テストケースで共通のテストデータを設定する場合に、テストケースのコピー機能を利用すると効率的に作業を進めることができます。

### 3. テストケースCSVを用いて テストパターンを増やしてみよう

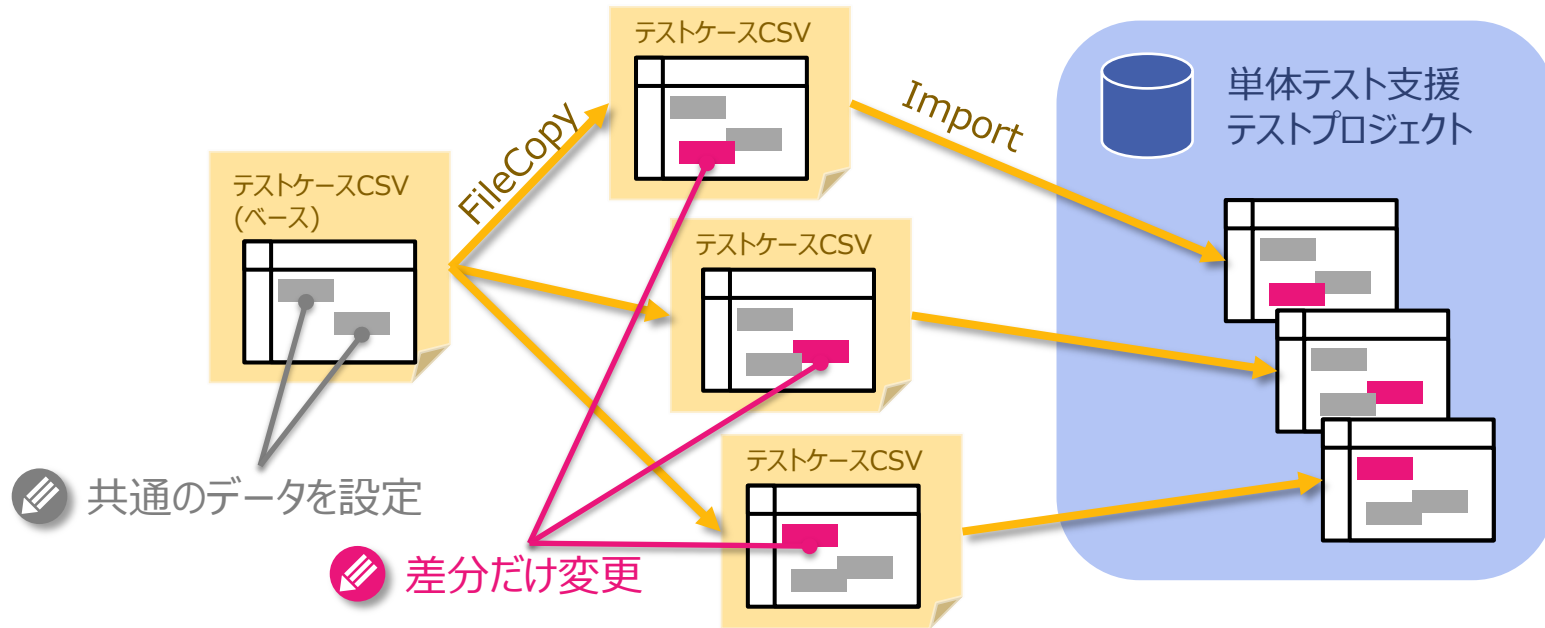


## 3-1. 演習する内容(演習2)

テストケースCSVを用いてテストパターンを増やす演習を行います。

演習2ではテストケースの情報を外部ファイルとして編集できるテストケースCSVを利用します。テストケースCSVで共通となるテストデータを設定し、そのファイルをコピーして一部のテストデータを編集&インポートすることでテストパターンを増やすことを行います。

演習2のゴールは「テストケースCSVを利用してテストパターンを増やし、カバレッジのC0メジャー(命令網羅)を100%とする単体テストを実施する」です。

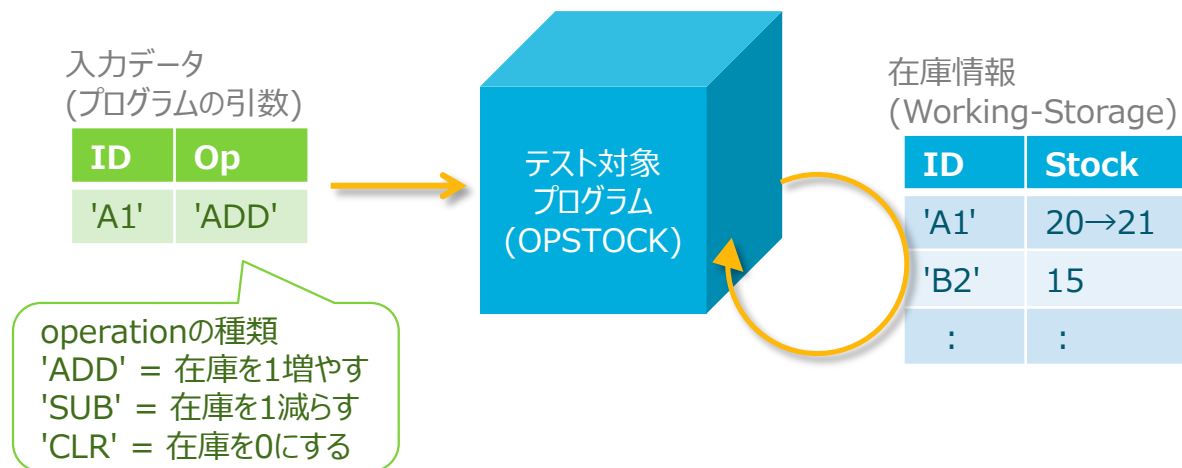


テスト対象のプログラム[OPSTOCK]について説明します。  
(演習1および演習2で共通です)

### 機能仕様 :

商品の在庫を操作するプログラムです。

プログラムでは在庫情報として商品IDとその個数を持ちます。  
プログラムの引数から入力データとして商品IDと操作の種類を受け取り、  
それに合わせて在庫情報を更新します。





## 3-3. 作成するテストケース

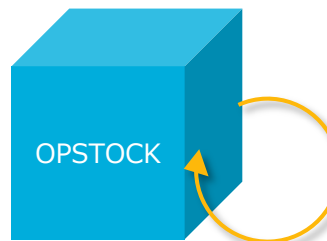
演習2では3つのテストケースを作成します。

### 演習2テストケース1 [ADDのテスト]

商品ID'C3'に対して操作'ADD'を実行した時のテストです。  
在庫情報のStockを期待値として確認します。

入力データ

ID	Op
'C3'	'ADD'



在庫情報

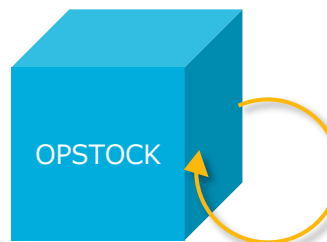
ID	Stock
'A1'	10
'B2'	20
'C3'	30→31
:	:

### 演習2テストケース2 [SUBのテスト]

商品ID'G7'に対して操作'SUB'を実行した時のテストです。  
在庫情報のStockを期待値として確認します。

入力データ

ID	Op
'G7'	'SUB'



在庫情報

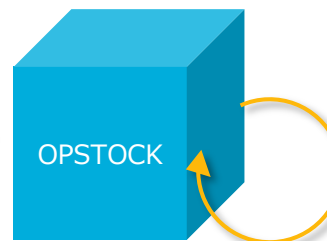
ID	Stock
:	:
'F6'	70
'G7'	80→79
:	:

### 演習2テストケース3 [CLRのテスト]

商品ID'J0'に対して操作'CLR'を実行した時のテストです。  
在庫情報のStockを期待値として確認します。

入力データ

ID	Op
'J0'	'CLR'

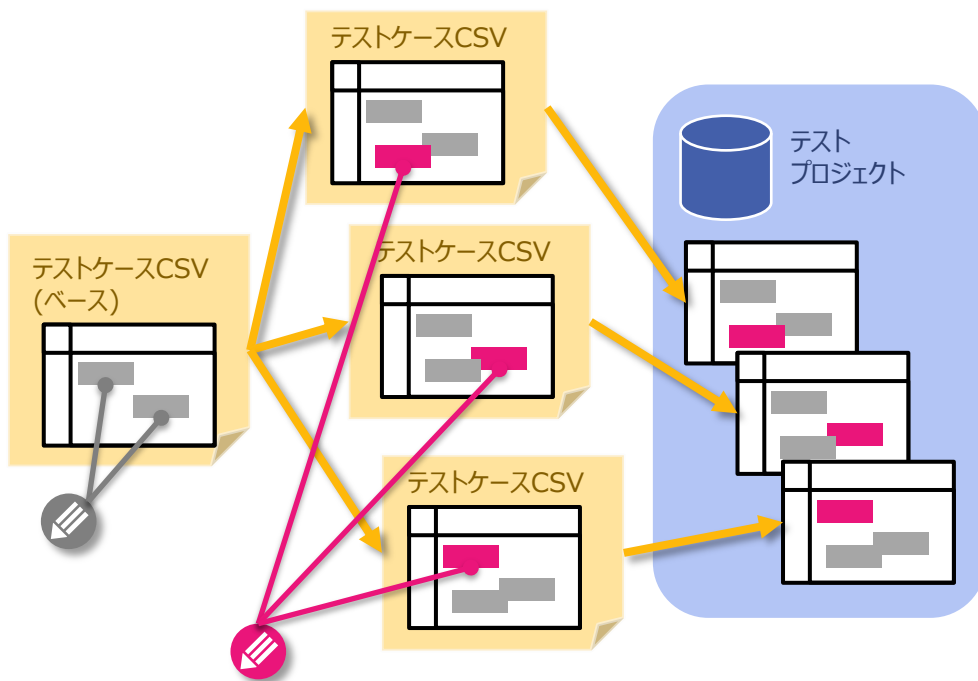


在庫情報

ID	Stock
:	:
'H8'	90
'I9'	100
'J0'	110→0

## 3-4. 演習2の流れ

演習の手順です。



準備

テストプロジェクトの準備

ひな形CSVのエクスポート

演習用CSVの準備

1つ目のテストケースの作成と編集

2つ目のテストケースの作成と編集

3つ目のテストケースの作成と編集

テストケースCSVのインポート

テストの実行

テスト結果の確認

テストケースごとの結果確認

カバレッジの結果確認

## 3-5. テストプロジェクトの準備

Progress

準備

テストケースCSV作成

インポート

実行

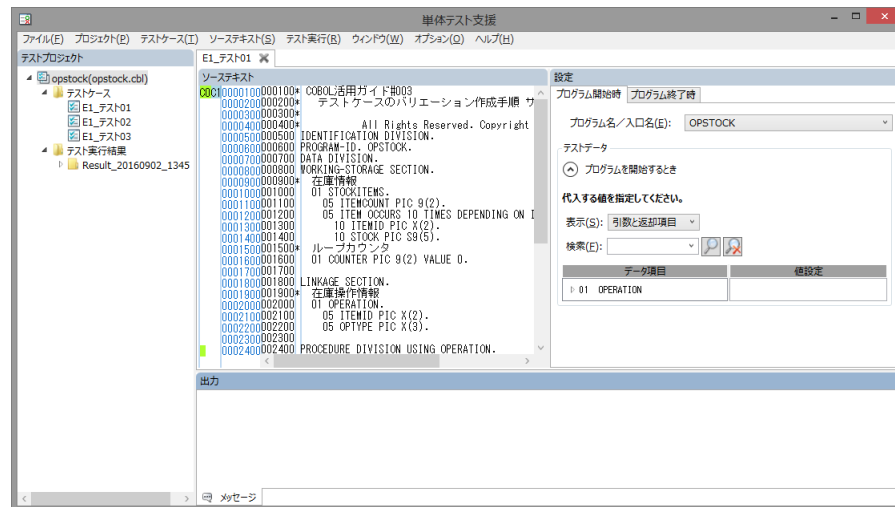
結果確認

演習2でも演習1で作成したテストプロジェクトを引き続き使います。

演習1を完了している場合は、そのままそのテストプロジェクトを操作してください。

演習1をスキップした場合は、演習1のテストプロジェクト作成までを実施してください。

演習1のテストケースやテスト実行結果については、作成する必要はありません。



## 3-6. ひな形テストケースCSVのエクスポート

Progress

準備

テストケースCSV作成

インポート

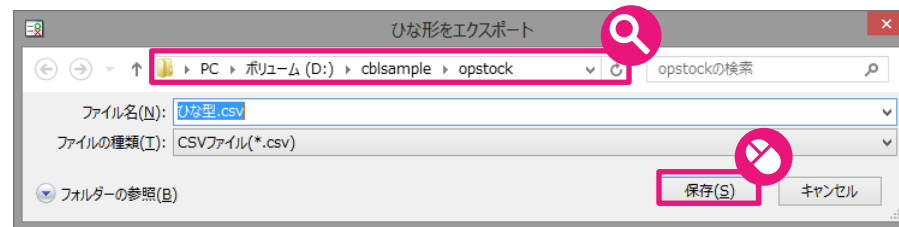
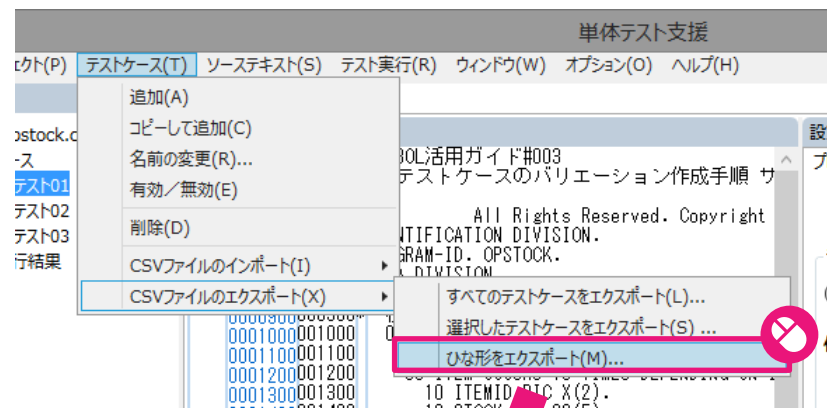
実行

結果確認

単体テスト支援ではテストケースをCSV形式でインポート/エクスポートできます。  
また、ひな形テストケースCSVは大量のテストデータを機械的に生成する場合などのベースとして主に利用します。  
ここではひな形テストケースCSVの出力を行います。

単体テスト支援のメニューから「テストケース」→「CSVファイルのエクスポート」→「ひな形をエクスポート」を選択します。

「ひな形エクスポート」ダイアログが表示されます。テストプロジェクトのあるopstockフォルダが選択されていることを確認して「保存」をクリックします。



## 3-7. ひな形テストケースCSVのエクスポート

Progress

準備

テストケースCSV作成

インポート

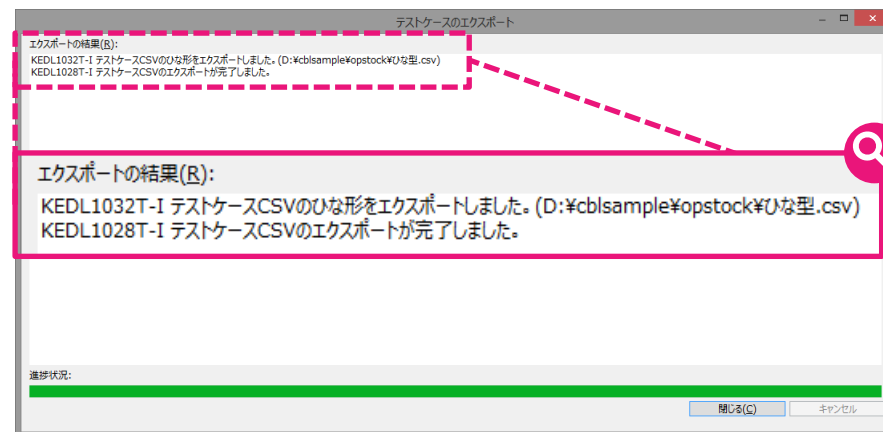
実行

結果確認

エクスポートを実行すると実行した結果のログ情報が表示されます。

ひな形テストケースCSVはテキストエディタなどのCSVを編集できるツールで内容を編集できます。複数のテストケースでの共通な初期値設定などは、このひな形テストケースCSVにあらかじめ設定しておき、これをコピーして個別のテストデータを設定する手順が効率的です。

今回の演習では、今エクスポートしたひな形テストケースCSVではなく、既に初期値となるテストデータを設定済みのテストケースCSVを元にテストケースを作成します。



## 3-8. 演習用テンプレートCSVの準備

Progress

準備

テストケースCSV作成

インポート

実行

結果確認

演習用の初期値設定済みのテストケースCSV(以降、演習用テンプレートCSV)を任意のフォルダに作成します。

「付録2. 演習用テンプレートCSV」に記載しているテキストをコピー&ペーストして、演習用テンプレートCSVのファイルを作成してください。以降は以下に作成したものとして説明します。

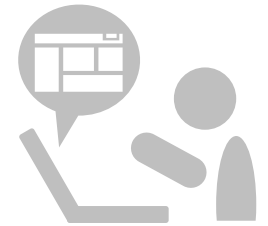
フォルダパス

```
D:¥cb1sample¥opstock¥template.csv
```

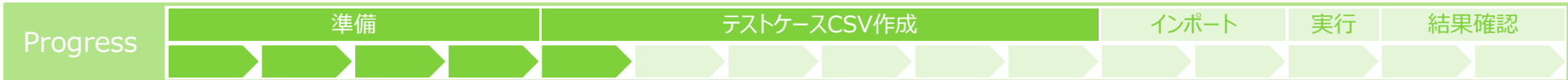
また、この演習用テンプレートCSVを元に、これから作成するテストケースCSVの格納フォルダを作成します。以降は以下に作成したものとして説明します。

フォルダパス

```
D:¥cb1sample¥opstock¥testcasecsv
```



# 3-9. 1つ目のテストケースの作成と編集



1つ目のテストケースを作成します。  
演習用テンプレートCSVファイルを以下のフォルダ  
にコピーしてファイル名を変更します。

コピー先  
D:\¥cblsample¥opstock¥testcasecsv¥  
E2\_テスト01.csv

コピーしたファイルを編集し、テストデータを設定し  
ます。  
CSVファイルを編集できるアプリケーションで開きま  
す。

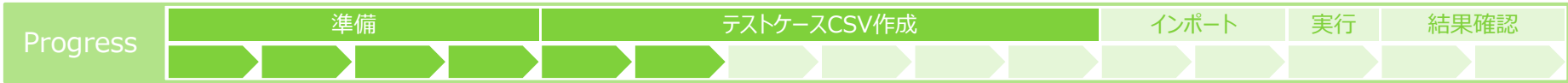
### TIPS

CSVファイルを編集できるアプリケーションはテキストエ  
ディタの他、MicrosoftのExcelなどがあります。  
演習の例では秀丸エディタのCSVモードを利用して  
います。

Excelで編集する場合には「付録4. CSV編集時の  
注意」も参照してください。

```
Version: 03-04
* テストプロジェクト名: opstock
* プログラム名: OPSTOCK
* CSVソースファイル: D:\¥cblsample¥opstock¥cbl
***** 【プログラム開始時】 画面の設定 *****
* <データ項<参照データ<データ項目名> <値の表示> <期待値> <代入値>
7 StartPoint, R, 5, ITEMCOUNT OF STOCKITEMS, FALSE, 10,
8 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (1), FALSE, A1,
9 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (1), FALSE, 20,
10 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (2), FALSE, B2,
11 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (2), FALSE, 30,
12 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (3), FALSE, C3,
13 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (3), FALSE, 40,
14 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (4), FALSE, D4,
15 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (4), FALSE, 50,
16 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (5), FALSE, E5,
17 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (5), FALSE, 60,
18 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (6), FALSE, F6,
19 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (6), FALSE, 70,
20 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (7), FALSE, G7,
21 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (7), FALSE, 80,
22 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (8), FALSE, H8,
23 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (8), FALSE, 90,
24 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (9), FALSE, I9,
25 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (9), FALSE, 100,
26 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (10), FALSE, J0,
27 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (10), FALSE, 110,
28 *** 【演習2】 在庫操作の情報を入力する部分 ***
29 StartPoint, R, 5, ITEMID OF OPERATION, FALSE, C3,
30 StartPoint, R, 5, OPTYPE OF OPERATION, FALSE, ADD,
31 ***** 【プログラム終了時】 画面の設定 *****
32 * <データ項<参照データ<データ項目名> <値の表示> <期待値> <代入値>
33 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (1), TRUE, A1,
34 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (1), TRUE, 20,
35 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (2), TRUE, B2,
36 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (2), TRUE, 30,
37 *** 【演習2】 テストケース01の期待値変更部分 ***
38 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (3), TRUE, C3,
39 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (3), TRUE, 41,
40 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (4), TRUE, D4,
41 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (4), TRUE, 50,
42 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (5), TRUE, E5,
43 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (5), TRUE, 60,
44 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (6), TRUE, F6,
45 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (6), TRUE, 70,
46 *** 【演習2】 テストケース02の期待値変更部分 ***
47 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (7), TRUE, G7,
48 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (7), TRUE, 80,
49 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (8), TRUE, H8,
50 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (8), TRUE, 90,
51 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (9), TRUE, I9,
52 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (9), TRUE, 100,
53 *** 【演習2】 テストケース03の期待値変更部分 ***
54 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (10), TRUE, J0,
55 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (10), TRUE, 110,
56 [EOF]
```

# 3-10. 1つ目のテストケースの作成と編集



2つの代入値と1つの期待値のテストデータを設定します。  
まず"\*\*\* [演習2] 在庫操作の情報を入力する部分 \*\*\*"の行を探します。  
その行に続く2行が代入値を設定する部分です。  
以下の値を<代入値>の列に入力します。

ITEMID OF OPERATION
'C3'
OPTYPE OF PERATION
'ADD'

次に"\*\*\* [演習2] テストケース01の期待値変更部分 \*\*\*"の行を探します。  
その次の行が期待値を設定する部分です。  
<期待値>の列を以下の値に変更します。

STOCK OF ITEM OF STOCKITEMS(3)
41

ファイルを上書き保存します。

```
1 # Version: 03-04
2 * テストプロジェクト名: opstock
3 * プログラム名: OPSTOCK
4 * CDBULソースファイル: D:\cblsample\opstock.cbl
5 ***** [プログラム開始時] 画面の設定 *****
6 * <データ項<参照データ<データ項目名> <値の表示> <期待値> <代入値>
7 StartPoint, R, 5, ITEMCOUNT OF STOCKITEMS, FALSE, '10',
8 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (1), FALSE, 'A1',
9 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (1), FALSE, '20',
10 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (2), FALSE, 'B2',
11 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (2), FALSE, '30',
12 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (3), FALSE, 'C3',
13 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (3), FALSE, '40',
14 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (4), FALSE, 'D4',
15 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (4), FALSE, '50',
16 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (5), FALSE, 'E5',
17 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (5), FALSE, '60',
18 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (6), FALSE, 'F6',
19 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (6), FALSE, '70',
20 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (7), FALSE, 'G7',
21 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (7), FALSE, '80',
22 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (8), FALSE, 'H8',
23 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (8), FALSE, '90',
24 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (9), FALSE, 'I9',
25 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (9), FALSE, '100',
26 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (10), FALSE, 'J0',
27 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (10), FALSE, '110',
28 *** [演習2] 在庫操作の情報を入力する部分 ***
29 StartPoint, R, 5, ITEMID OF OPERATION, FALSE, 'C3',
30 StartPoint, R, 6, OPTYPE OF OPERATION, FALSE, 'ADD',
31 ***** [プログラム終了時] 画面の設定 *****
32 * <データ項<参照データ<データ項目名> <値の表示> <期待値> <代入値>
33 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (1), TRUE, 'A1',
34 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (1), TRUE, '20',
35 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (2), TRUE, 'B2',
36 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (2), TRUE, '30',
37 *** [演習2] テストケース01の期待値変更部分 ***
38 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (3), TRUE, 'C3',
39 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (3), TRUE, '41',
40 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (4), TRUE, 'D4',
41 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (4), TRUE, '50',
42 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (5), TRUE, 'E5',
43 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (5), TRUE, '60',
44 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (6), TRUE, 'F6',
45 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (6), TRUE, '70',
46 *** [演習2] テストケース02の期待値変更部分 ***
47 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (7), TRUE, 'G7',
48 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (7), TRUE, '80',
49 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (8), TRUE, 'H8',
50 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (8), TRUE, '90',
51 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (9), TRUE, 'I9',
52 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (9), TRUE, '100',
53 *** [演習2] テストケース03の期待値変更部分 ***
54 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (10), TRUE, 'J0',
55 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (10), TRUE, '110',
56 [EOF]
```



# 3-11. 2つ目のテストケースの作成と編集



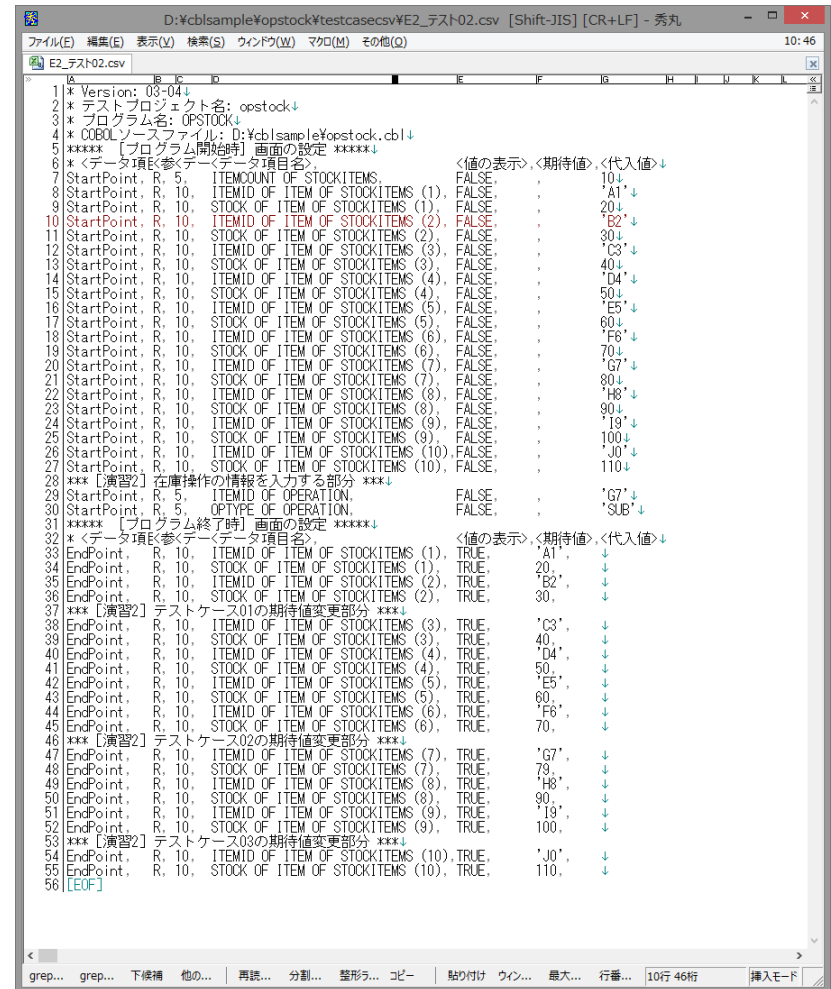
2つ目のテストケースを作成します。  
演習用テンプレートCSVファイルを以下のフォルダ  
にコピーしてファイル名を変更します。

コピー先  
D:¥cblsample¥opstock¥testcasecsv¥  
E2\_テスト02.csv

コピーしたファイルを編集し、テストデータを設定し  
ます。  
CSVファイルを編集できるアプリケーションで開きま  
す。

### TIPS

CSVファイルを編集できるアプリケーションはテキストエ  
ディタの他、MicrosoftのExcelなどがあります。  
演習の例では秀丸エディタのCSVモードを利用して  
います。  
Excelで編集する場合には「付録4. CSV編集時の  
注意」も参照してください。



# 3-12. 2つ目のテストケースの作成と編集



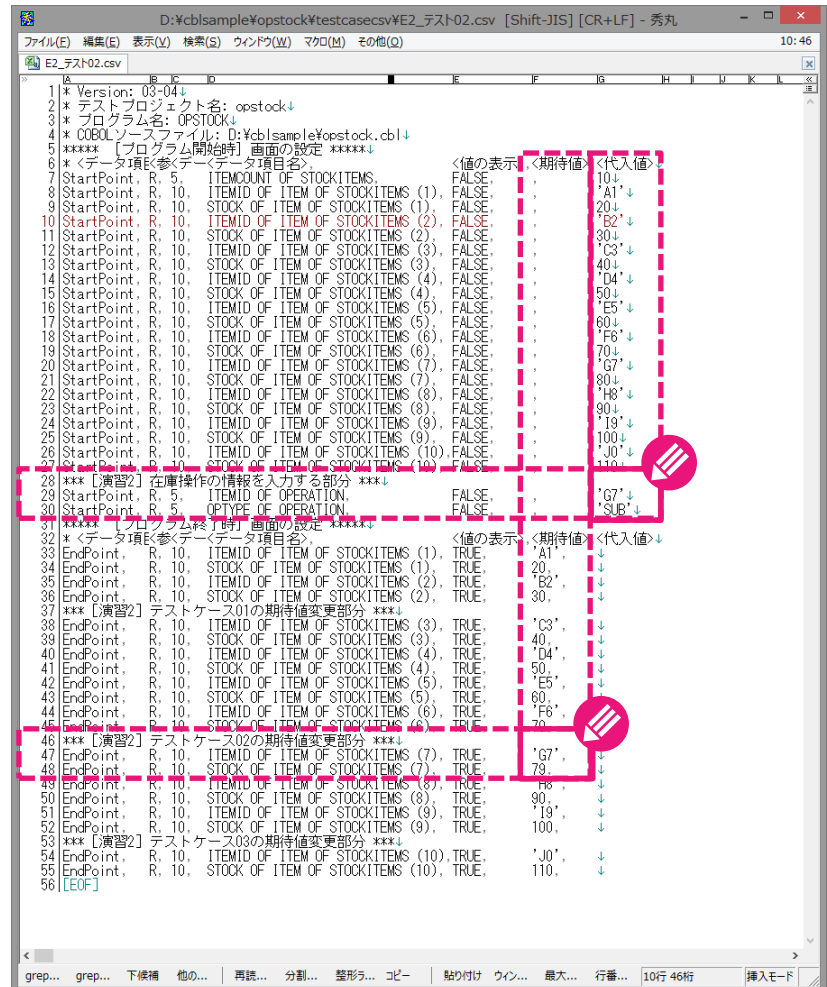
2つの代入値と1つの期待値のテストデータを設定します。  
まず"\*\*\* [演習2] 在庫操作の情報を入力する部分 \*\*\*"の行を探します。  
その行に続く2行が代入値を設定する部分です。  
以下の値を<代入値>の列に入力します。

ITEMID OF OPERATION
'G7'
OPTYPE OF OPERATION
'SUB'

次に"\*\*\* [演習2] テストケース02の期待値変更部分 \*\*\*"の行を探します。  
その次の行が期待値を設定する部分です。  
<期待値>の列を以下の値に変更します。

STOCK OF ITEM OF STOCKITEMS(7)
79

ファイルを上書き保存します。



# 3-13. 3つ目のテストケースの作成と編集

Progress

準備

テストケースCSV作成

インポート

実行

結果確認

3つ目のテストケースを作成します。  
演習用テンプレートCSVファイルを以下のフォルダ  
にコピーしてファイル名を変更します。

コピー先

D:\¥cblsample¥opstock¥testcasecsv¥  
E2\_テスト03.csv

コピーしたファイルを編集し、テストデータを設定し  
ます。  
CSVファイルを編集できるアプリケーションで開きま  
す。

## TIPS

CSVファイルを編集できるアプリケーションはテキストエ  
ディタの他、MicrosoftのExcelなどがあります。  
演習の例では秀丸エディタのCSVモードを利用して  
います。

Excelで編集する場合には「付録4. CSV編集時の  
注意」も参照してください。

```
D:\¥cblsample¥opstock¥testcasecsv¥E2_テスト03.csv [Shift-JIS] [CR+LF] - 秀丸
ファイル(E) 編集(E) 表示(V) 検索(S) ウィンドウ(W) マクロ(M) その他(O) 1:1
E2_テスト03.csv
1 * Version: 03-04↓
2 * テストプロジェクト名: opstock↓
3 * プログラム名: OPSTOCK↓
4 * ODBLソースファイル: D:\¥cblsample¥opstock.cb↓
5 ***** [プログラム開始時] 画面の設定 *****
6 * <データ項<参くデータ項目名> <値の表示>、<期待値>、<代入値>↓
7 StartPoint, R, 5, ITEMOUNT OF STOCKITEMS, FALSE, , 10↓
8 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (1), FALSE, , 'A1'↓
9 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (1), FALSE, , 20↓
10 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (2), FALSE, , 'B2'↓
11 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (2), FALSE, , 30↓
12 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (3), FALSE, , 'C3'↓
13 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (3), FALSE, , 40↓
14 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (4), FALSE, , 'D4'↓
15 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (4), FALSE, , 50↓
16 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (5), FALSE, , 'E5'↓
17 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (5), FALSE, , 60↓
18 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (6), FALSE, , 'F6'↓
19 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (6), FALSE, , 70↓
20 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (7), FALSE, , 'G7'↓
21 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (7), FALSE, , 80↓
22 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (8), FALSE, , 'H8'↓
23 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (8), FALSE, , 90↓
24 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (9), FALSE, , 'I9'↓
25 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (9), FALSE, , 100↓
26 StartPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (10), FALSE, , 'J0'↓
27 StartPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (10), FALSE, , 110↓
28 *** [演習2] 在庫操作の情報を入力する部分 ***↓
29 StartPoint, R, 5, ITEM OF OPERATION, FALSE, , 'J0'↓
30 StartPoint, R, 5, OPTYPE OF OPERATION, FALSE, , 'CLR'↓
31 ***** [プログラム終了時] 画面の設定 *****
32 * <データ項<参くデータ項目名> <値の表示>、<期待値>、<代入値>↓
33 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (1), TRUE, , 'A1'↓
34 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (1), TRUE, , 20↓
35 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (2), TRUE, , 'B2'↓
36 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (2), TRUE, , 30↓
37 *** [演習2] テストケース01の期待値変更部分 ***↓
38 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (3), TRUE, , 'C3'↓
39 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (3), TRUE, , 40↓
40 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (4), TRUE, , 'D4'↓
41 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (4), TRUE, , 50↓
42 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (5), TRUE, , 'E5'↓
43 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (5), TRUE, , 60↓
44 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (6), TRUE, , 'F6'↓
45 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (6), TRUE, , 70↓
46 *** [演習2] テストケース02の期待値変更部分 ***↓
47 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (7), TRUE, , 'G7'↓
48 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (7), TRUE, , 80↓
49 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (8), TRUE, , 'H8'↓
50 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (8), TRUE, , 90↓
51 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (9), TRUE, , 'I9'↓
52 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (9), TRUE, , 100↓
53 *** [演習2] テストケース03の期待値変更部分 ***↓
54 EndPoint, R, 10, ITEMID OF ITEM OF STOCKITEMS (10), TRUE, , 'J0'↓
55 EndPoint, R, 10, STOCK OF ITEM OF STOCKITEMS (10), TRUE, , 0↓
56 [EOF]
```

# 3-14. 3つ目のテストケースの作成と編集



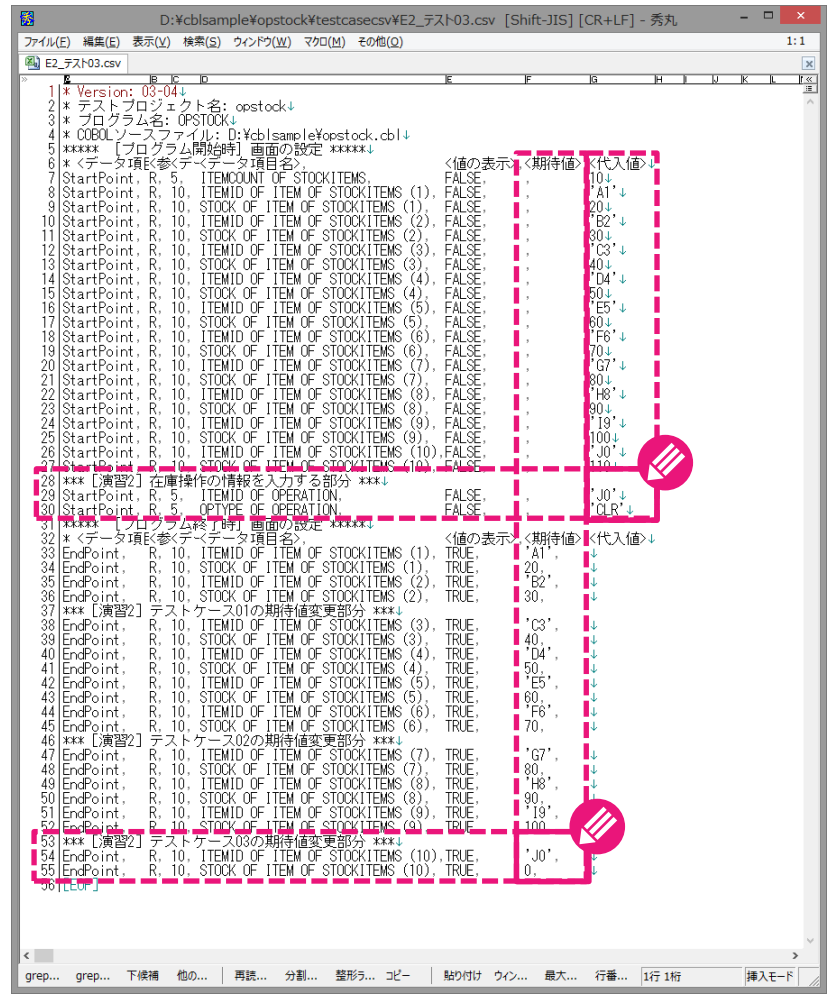
2つの代入値と1つの期待値のテストデータを設定します。  
まず"\*\*\* [演習2] 在庫操作の情報を入力する部分 \*\*\*"の行を探します。  
その行に続く2行が代入値を設定する部分です。  
以下の値を<代入値>の列に入力します。

ITEMID OF OPERATION
'J0'
OPTYPE OF PERATION
'CLR'

次に"\*\*\* [演習2] テストケース03の期待値変更部分 \*\*\*"の行を探します。  
その次の行が期待値を設定する部分です。  
<期待値>の列を以下の値に変更します。

STOCK OF ITEM OF STOCKITEMS(10)
0

ファイルを上書き保存します。



# 3-15. テストケースCSVのインポート

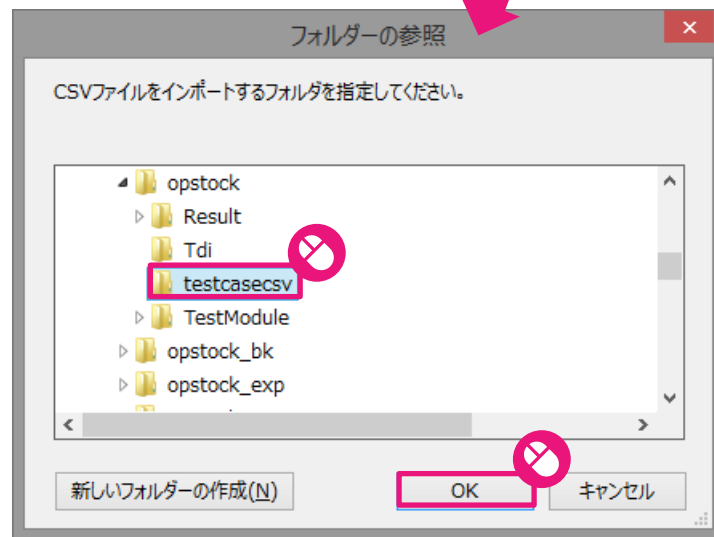
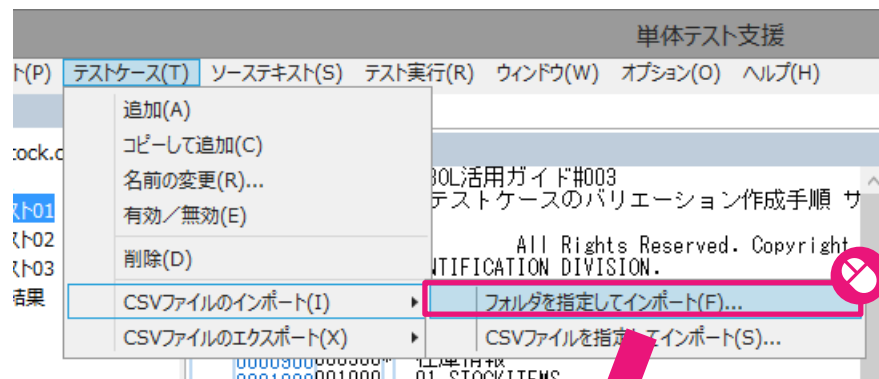


作成したテストケースCSVをテストプロジェクトにインポートします。  
メニューから「テストケース」→「CSVファイルのインポート」→「フォルダを指定してインポート」をクリックします。

「フォルダーの参照」ダイアログで、作成したテストケースCSVを格納したフォルダを選択し「OK」をクリックします。

選択するフォルダ

D:¥cblsample¥opstock¥testcasecsv

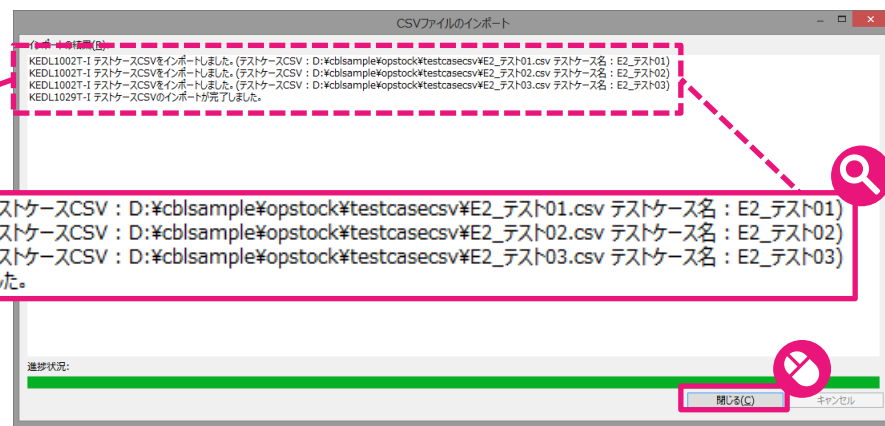


# 3-16. テストケースCSVのインポート

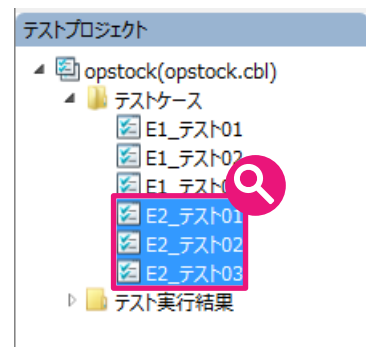


「CSVファイルのインポート」ダイアログが表示され、作成したテストケースCSVがインポートされます。インポート処理が完了したら「閉じる」をクリックしてダイアログを閉じます。

KEDL1002T-I テストケースCSVをインポートしました。(テストケースCSV : D:\%cblsample%\opstock%\testcasescsv\E2\_テスト01.csv テストケース名 : E2\_テスト01)  
KEDL1002T-I テストケースCSVをインポートしました。(テストケースCSV : D:\%cblsample%\opstock%\testcasescsv\E2\_テスト02.csv テストケース名 : E2\_テスト02)  
KEDL1002T-I テストケースCSVをインポートしました。(テストケースCSV : D:\%cblsample%\opstock%\testcasescsv\E2\_テスト03.csv テストケース名 : E2\_テスト03)  
KEDL1029T-I テストケースCSVのインポートが完了しました。



これで作成したテストケースCSVのテストデータがテストプロジェクトに取り込まれ、テストケースとなりました。



# 3-17. テストの実行

Progress

準備

テストケースCSV作成

インポート

実行

結果確認

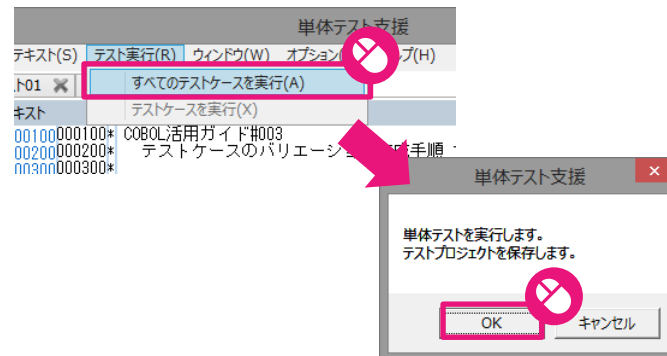
単体テストを実行します。  
メニューから「テスト実行」→「すべてのテストケース  
を実行」をクリックし、表示されるダイアログで  
「OK」をクリックします。

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が  
表示されます。  
このレポートには実行したテストケースの結果が一  
覧で表示されます。

判定結果がすべて"OK"となりテスト完了です。

テスト自体はこれで完了ですが、実行したテスト  
ケースの詳細結果を見てみましょう。



出力

### テストプロジェクト結果レポート

コンパイラ情報: COBOL2002(64) 03-04  
ソースファイル名: opstock.cbl  
開始時間: 2016/09/05 17:57:59  
終了時間: 2016/09/05 17:57:59  
経過時間: 00:00:00  
エビデンスファイル格納フォルダ: D:\%cblsample%opstock%Result%Result\_20160905\_175759

テストケース 合計:6件  
OK 6件(100%) NG 0件(0%) 未判定 0件(0%) エラ 0件(0%)

#	テストケース	判定結果
1	E1_テスト01	OK
2	E1_テスト02	OK
3	E1_テスト03	OK
4	E2_テスト01	OK
5	E2_テスト02	OK
6	E2_テスト03	OK

メッセージ Result\_20160905\_175759

# 3-18. テスト結果の詳細確認



ここで詳細なテスト結果の確認としてテストケース "E2\_テスト01"とカバレッジの情報を確認します。

まずテストケース"E2\_テスト01"についての詳細結果を確認します。  
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て結果レポートの"E2\_テスト01"をダブルクリックします。

画面下部に"E2\_テスト\_01"のテストケース結果レポートが表示されます。  
このテストケースでは'C3'の在庫を1つ増やした処理をテストしています。  
結果の値と判定結果のSTOCK OF ITEM OF STOCKITEMS(3)の値が41になってOKとなっていることがわかります。

**TIPS**  
ここでは"E2\_テスト01"について結果の確認を行いました。同様に"E2\_テスト02"や"E2\_テスト03"についても確認してみてください。

テストプロジェクト

- opstock(opstock.cbl)
  - テストケース
    - E1\_テスト01
    - E2\_テスト01
    - E2\_テスト02
    - E2\_テスト03
  - テスト実行結果
    - Result\_20160902\_134515
    - Result\_20160906\_101532
      - カバレッジレポート
      - テストプロジェクト結果レポート
      - 結果レポート
        - E1\_テスト01
        - E1\_テスト02
        - E1\_テスト03
        - E2\_テスト01**
        - E2\_テスト02
        - E2\_テスト03

出力

テストケース結果レポート

コンパイル情報: COBOL2002(64) 03-04  
ソースファイル名: opstock.cbl  
テストケース: E2\_テスト01

判定結果: OK

結果の値と判定結果

テストデータ設定箇所	レベル	データ項目	値	期待値	判定結果
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"A1"	"A1"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00020	20	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"B2"	"B2"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00030	30	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"C3"	"C3"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00041	41	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"E5"	"E5"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00050	50	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"F5"	"F5"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00070	70	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"G7"	"G7"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00080	80	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"H8"	"H8"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00090	90	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"I9"	"I9"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00100	100	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	ITEMID OF ITEM OF STOCKI	"J0"	"J0"	OK
プログラム終了時(OPSTOCK/OPSTOCK)	10	STOCK OF ITEM OF STOCKI	+00110	110	OK



Progress

準備

テストケースCSV作成

インポート

実行

結果確認

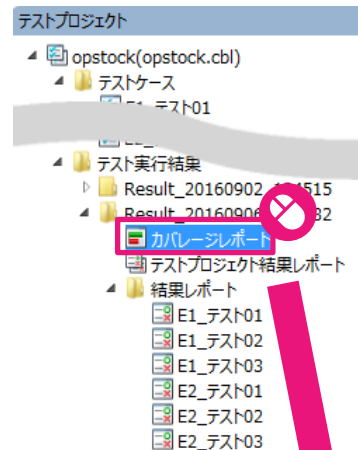
次にカバレッジの結果を確認します。  
画面左側の「テストプロジェクト」ツリーのテスト実行結果から、先ほど実行した結果のツリーを展開してカバレッジレポートをダブルクリックします。

画面下部にカバレッジレポートが表示されます。  
C0メジャーが100%で、C1メジャーが87.5%となっていることが確認できます。  
この演習ではC0を100%とするテストデータを設定しました。C1を100%とするテストデータの作成にもチャレンジしてみてください。

テスト完了!

## TIPS

C1は分岐網羅であり、100%にするには全ての分岐を通るテストデータが必要となります。  
サンプルコードのIF文とEVALUATE文では、演習のテストデータだけでは分岐を全て通っていないパターンが存在します。それらに対応するテストケースを追加することでC1も100%とすることが可能です。  
答えは「付録3. C1メジャー100%にするには」を参照してください。



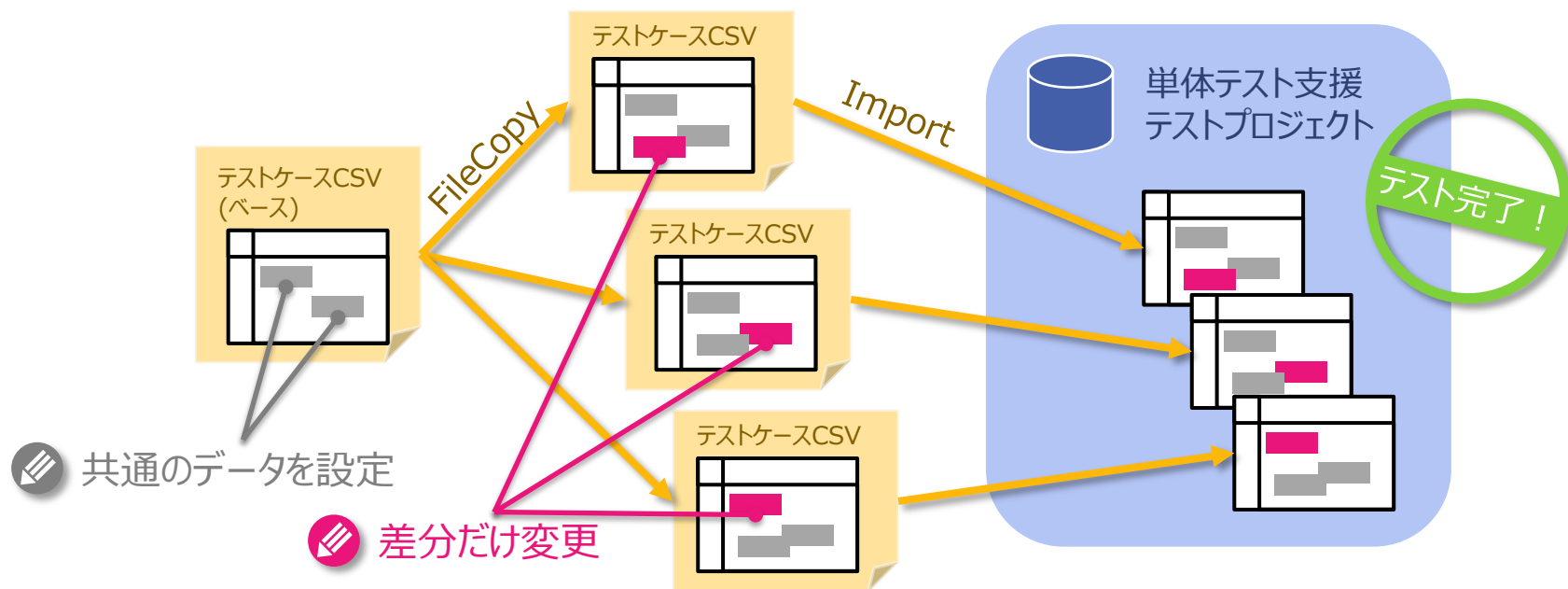
出力

**カバレッジレポート**

コンパイラ情報: COBOL2002(64) 03-04  
ソースファイル名: opstock.cbl

種類	カバレッジ
C0メジャー	100.0%
C1メジャー	87.5%
S1メジャー	---

メッセージ Result\_20160906\_101532:E2\_テスト01 Result\_20160906\_101532



テストケースで共通のテストデータを設定したテンプレートのテストケースCSVを元に、一部のデータを変更してテストケースCSV(E2\_テスト01, E2\_テスト02, E2\_テスト03)を作成し、それをテストプロジェクトにインポートしてテストケースを増やす演習を行いました。

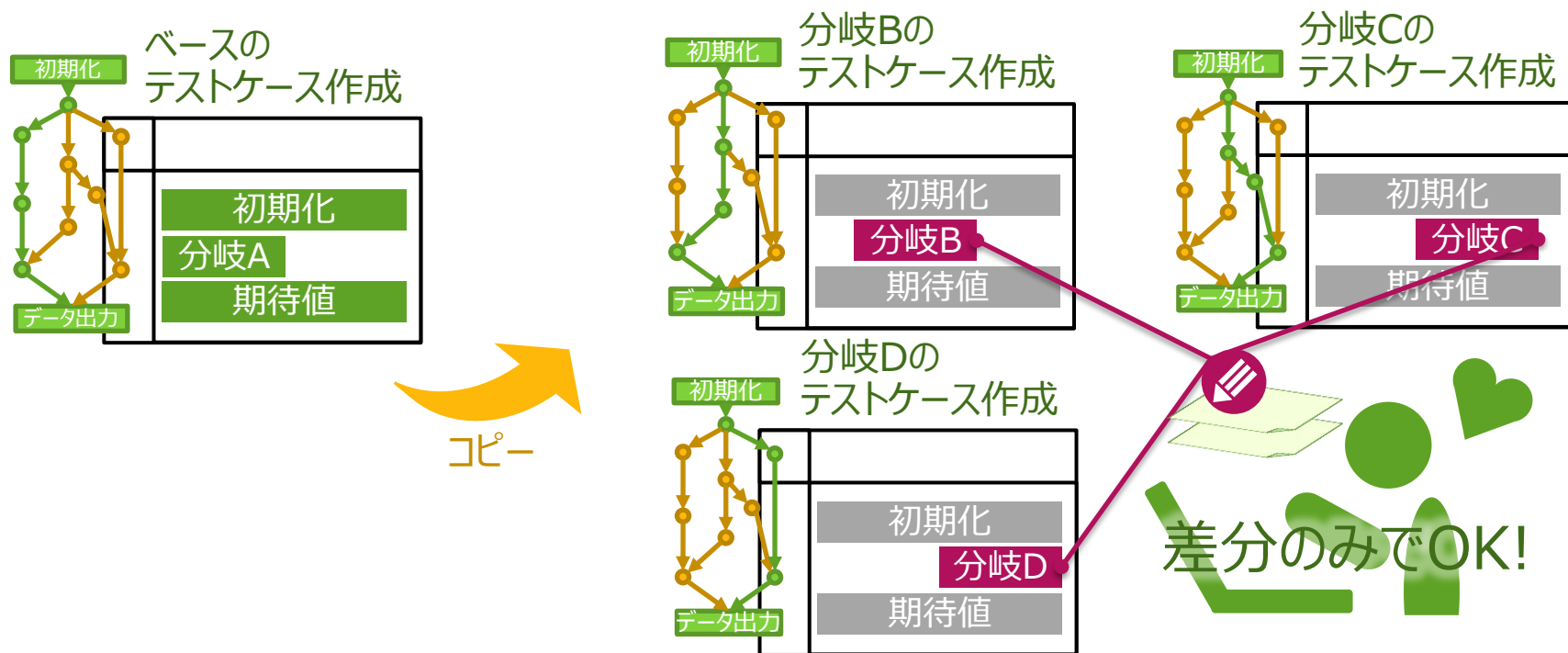
テストケースCSVにより、テストデータをツールなどを用いて機械的に一括生成するなど可能であり、より効率よくテストケースを作成することができます。  
なお、演習2ではベースとなるテストケースCSVは作成済みの演習用テンプレートを用いましたが、実際に適用する場合には、このテンプレートをひな形テストケースCSVを元に作成する必要があります。

---

## 4. まとめ



単体テスト支援のテストケースのコピー機能を活用することで、  
テストデータ設定の手間を削減でき、開発効率の向上に役立ちます。



テストケースのコピー機能以外にも、単体テスト支援は単体テストに有用な機能を取り揃えています。それらについてはCOBOL活用ガイドなどで紹介していますので、ご参照ください。

- 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/>

## opstock.cbl

```
000100* COBOL活用ガイド#003
000200* テストケースのコピーを利用したテストデータ作成手順 サンプルコード
000300*
000400* All Rights Reserved. Copyright (C) 2016, Hitachi, Ltd.
000500 IDENTIFICATION DIVISION.
000600 PROGRAM-ID. OPSTOCK.
000700 DATA DIVISION.
000800 WORKING-STORAGE SECTION.
000900* 在庫情報
001000 01 STOCKITEMS.
001100 05 ITEMCOUNT PIC 9(2).
001200 05 ITEM OCCURS 10 TIMES DEPENDING ON ITEMCOUNT.
001300 10 ITEMID PIC X(2).
001400 10 STOCK PIC S9(5).
001500* ループカウンタ
001600 01 COUNTER PIC 9(2) VALUE 0.
001700
001800 LINKAGE SECTION.
001900* 在庫操作情報
002000 01 OPERATION.
002100 05 ITEMID PIC X(2).
002200 05 OPTYPE PIC X(3).
002300
002400 PROCEDURE DIVISION USING OPERATION.
002500
002600 PERFORM VARYING COUNTER FROM 1 BY 1 UNTIL ITEMCOUNT < COUNTER
002700 IF ITEMID OF ITEM(COUNTER) = ITEMID OF OPERATION THEN
002800 EVALUATE OPTYPE OF OPERATION
002900 WHEN 'ADD'
003000 ADD 1 TO STOCK OF ITEM(COUNTER)
003100 WHEN 'SUB'
003200 ADD -1 TO STOCK OF ITEM(COUNTER)
003300 WHEN 'CLR'
003400 MOVE 0 TO STOCK OF ITEM(COUNTER)
003500 END-EVALUATE
003600 END-IF
003700 END-PERFORM.
003800
003900 EXIT PROGRAM.
004000 END PROGRAM OPSTOCK.
```

## template.csv

```
* Version: 03-04
* テストプロジェクト名: opstock
* プログラム名: OPSTOCK
* COBOLソースファイル: D:\%cbldsample%\opstock.cbl
***** [プログラム開始時] 画面の設定 *****
* <データ項目の設定個所の情報>, <参照フラグ>, <データ項目のレベル番号>, <データ項目名>, <値の表示>, <期待値>, <代入値>
StartPoint,R,5,ITEMCOUNT OF STOCKITEMS,FALSE,,10
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (1),FALSE,,A1'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (1),FALSE,,20
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (2),FALSE,,B2'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (2),FALSE,,30
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (3),FALSE,,C3'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (3),FALSE,,40
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (4),FALSE,,D4'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (4),FALSE,,50
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (5),FALSE,,E5'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (5),FALSE,,60
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (6),FALSE,,F6'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (6),FALSE,,70
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (7),FALSE,,G7'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (7),FALSE,,80
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (8),FALSE,,H8'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (8),FALSE,,90
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (9),FALSE,,I9'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (9),FALSE,,100
StartPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (10),FALSE,,J0'
StartPoint,R,10,STOCK OF ITEM OF STOCKITEMS (10),FALSE,,110
*** [演習2] 在庫操作の情報を入力する部分 ***
StartPoint,R,5,ITEMID OF OPERATION,FALSE,,
StartPoint,R,5,OPTYPE OF OPERATION,FALSE,,
***** [プログラム終了時] 画面の設定 *****
* <データ項目の設定個所の情報>, <参照フラグ>, <データ項目のレベル番号>, <データ項目名>, <値の表示>, <期待値>, <代入値>
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (1),TRUE,A1',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (1),TRUE,20,
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (2),TRUE,B2',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (2),TRUE,30,
*** [演習2] テストケース01の期待値変更部分 ***
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (3),TRUE,C3',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (3),TRUE,40,
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (4),TRUE,D4',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (4),TRUE,50,
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (5),TRUE,E5',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (5),TRUE,60,
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (6),TRUE,F6',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (6),TRUE,70,
*** [演習2] テストケース02の期待値変更部分 ***
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (7),TRUE,G7',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (7),TRUE,80,
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (8),TRUE,H8',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (8),TRUE,90,
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (9),TRUE,I9',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (9),TRUE,100,
*** [演習2] テストケース03の期待値変更部分 ***
EndPoint,R,10,ITEMID OF ITEM OF STOCKITEMS (10),TRUE,J0',
EndPoint,R,10,STOCK OF ITEM OF STOCKITEMS (10),TRUE,110,
```

演習1および2ではC0メジャー(命令網羅)を100%とするテストケースを作成しました。  
ただしC1メジャー(分岐網羅)は100%とはなっていません。

以下は演習1のテストプロジェクトを例に説明します。

サンプルコードにはIF文とEVALUATE文の処理が存在します。

サンプルコードでは明示的には書かれていませんが、

IF文には暗黙的に定義されたELSE節

EVALUATE文には暗黙的に定義されたWHEN OTHER節

がそれぞれ存在します。

C1メジャーを100%とする場合には、これらの暗黙的に定義された部分についてもテストを行う必要があります。

なので、

IF文には暗黙的に定義されたELSE節

→ 在庫情報に設定されていないITEMIDを入力データに指定

EVALUATE文には暗黙的に定義されたWHEN OTHER節

→ 在庫の操作として定義されていないOPTYPE文字列を入力データに設定

のテストが必要となります。

例えば以下のようなテストケースを追加することでC1メジャー100%になります。

テストケースEx1：入力データのITEMIDに'X1'

テストケースEx2：入力データのOPTYPEに'DMY'



Microsoft ExcelでテストケースCSVを編集するときの注意点を以下に記載します。

- 文字列を代入値や期待値に入力時の'(クォート)や"(ダブルクォート)  
代入値や期待値として文字列を入力する場合、'(クォート)や"(ダブルクォート)で文字列の前後をくくする必要があります。  
セルにこれらの文字を入力した場合、Excelの設定によってCSVファイルに出力したときの入力値が異なることがあります。  
例えば、セルの始めの文字に'(クォート)を入力すると文字列セル扱いとなり、先頭の'(クォート)が省略されます。このままだと単体テスト支援では文字列という扱いとならないため、先頭の'(クォート)を2個連続で入力してください。

eq. セルに'ABC'を入力する場合 → ["ABC"]と入力します。

正常に入力されているかを確認するには、ExcelでCSVファイルを保存した後に、テキストエディタでCSVファイルを読み込み、'(クォート)や"(ダブルクォート)が意図した文字列となっているかどうかを確認してください。

- セル内部で改行  
テストケースCSVではセルの内部で改行する入力には対応していません。  
セル内部で改行すると単体テスト支援で正常にインポートできませんので改行を入れないでください。

## ■ 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。

MicrosoftおよびExcelは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。  
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。