

演習 [トライキャッチ (Try Catch)] を使用したビジネス例外処理

実践演習

次の処理を実行するワークフローを作成してみましょう

[トライキャッチ (Try Catch)] を使用し、Excel ファイルの列 A または列 B に空白が入っていた場合に、Catch ブロックで `NullReferenceException` として、例外があったことを実行結果として列 C のセルに書き込み、処理を続行するようなワークフローを作成しましょう

※ ダウンロードした zip ファイルにある「合計.xlsx」を使用したワークフローを作成します

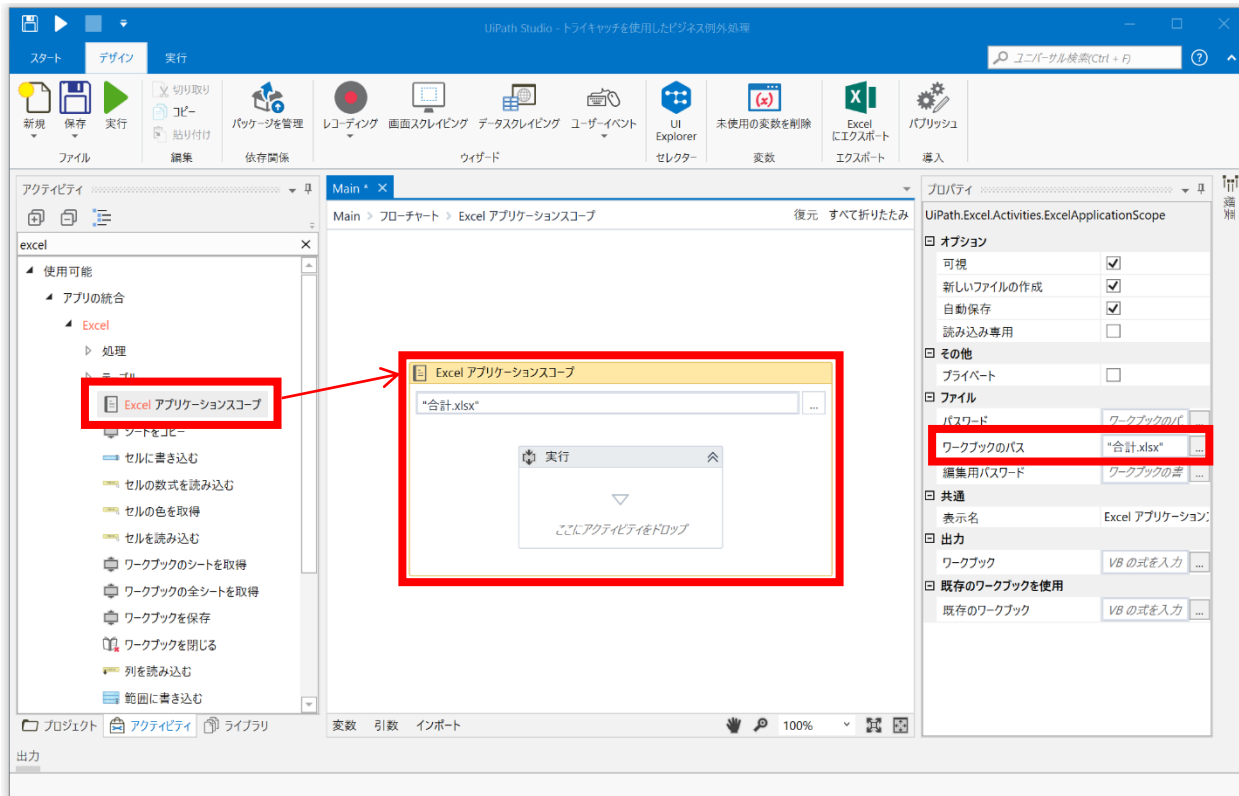
	A	B	C	D	E
1	159	36			
2	3359	35			
3	9072	8996			
4	623	56			
5	11	324			
6	6643	88942			
7		6			
8	3213	4324			
9	7732	43			
10	952	13			
11	3577	91			
12					
13					

- ① 列 A のセルの値 + 列 B のセルの値 を列 C に書き込む
- ② この時、空セルがあれば、[トライキャッチ (Try Catch)] を使用し、「空のセルがありました」と列 C に記録する
- ③ 次の行の処理へ進む（空セルの有無にかかわらず、Excel にデータがある最終行まで列 C に結果を書き込むワークフローにする）

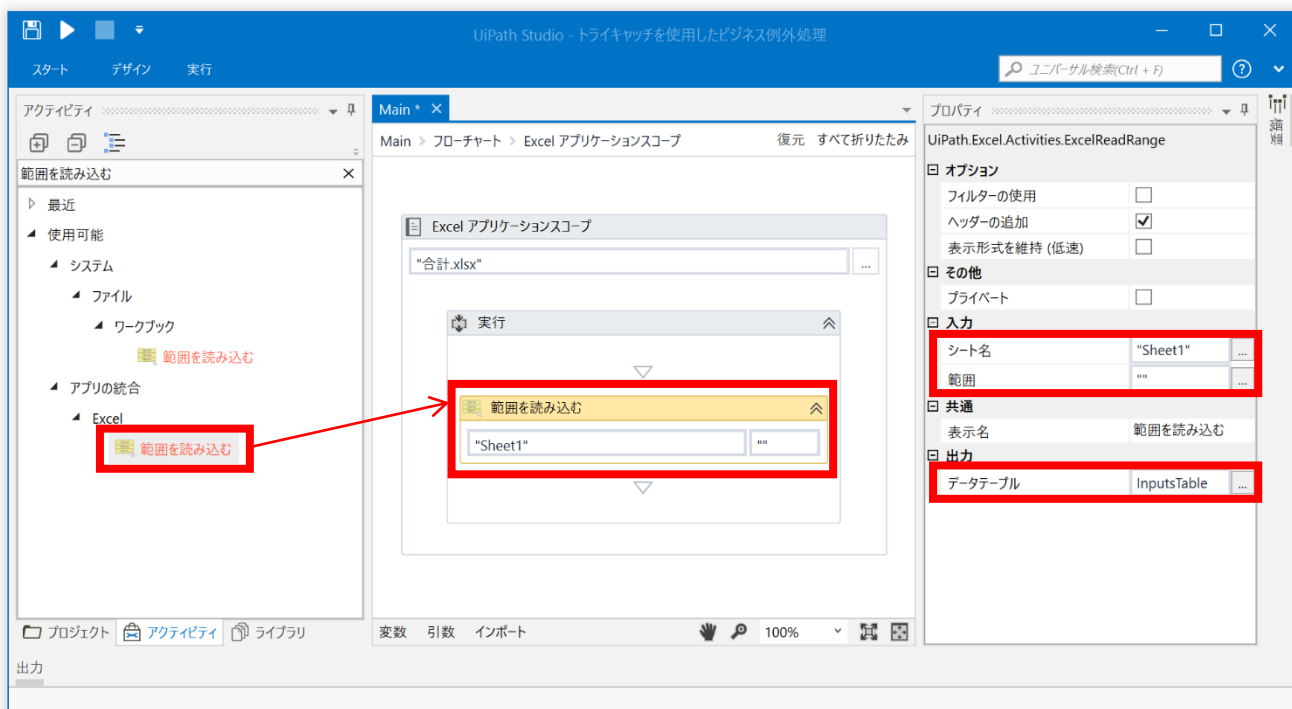
解答例

以下の手順でワークフローを作成してみましょう

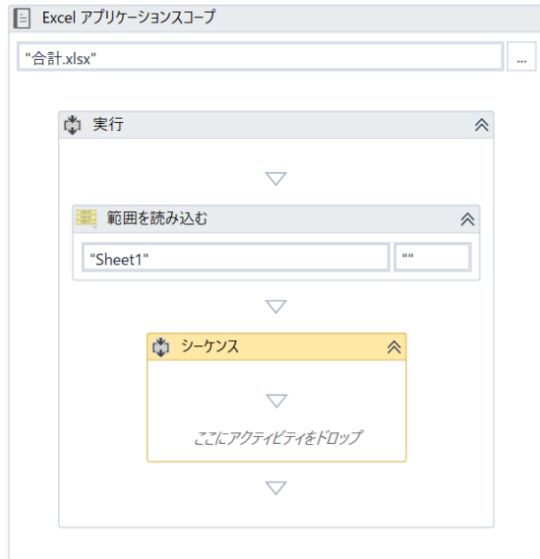
1. アクティビティパネルから [Excel アプリケーションスコープ (Excel Application Scope)] をデザイナーパネルにドラッグ&ドロップし、参照する Excel ファイル「**合計.xlsx**」を指定します



2. 「**範囲を読み込む (Read Range)**」を[Excel アプリケーションスコープ (Excel Application Scope)]の中にドラッグ&ドロップし、シート名に「**"Sheet1"**」、範囲に「**"**」を指定、出力データテーブルには、「**InputsTable**」という変数名を作成します



3. [範囲を読み込む (Read Range)] の下に [シーケンス (Sequence)] を配置します

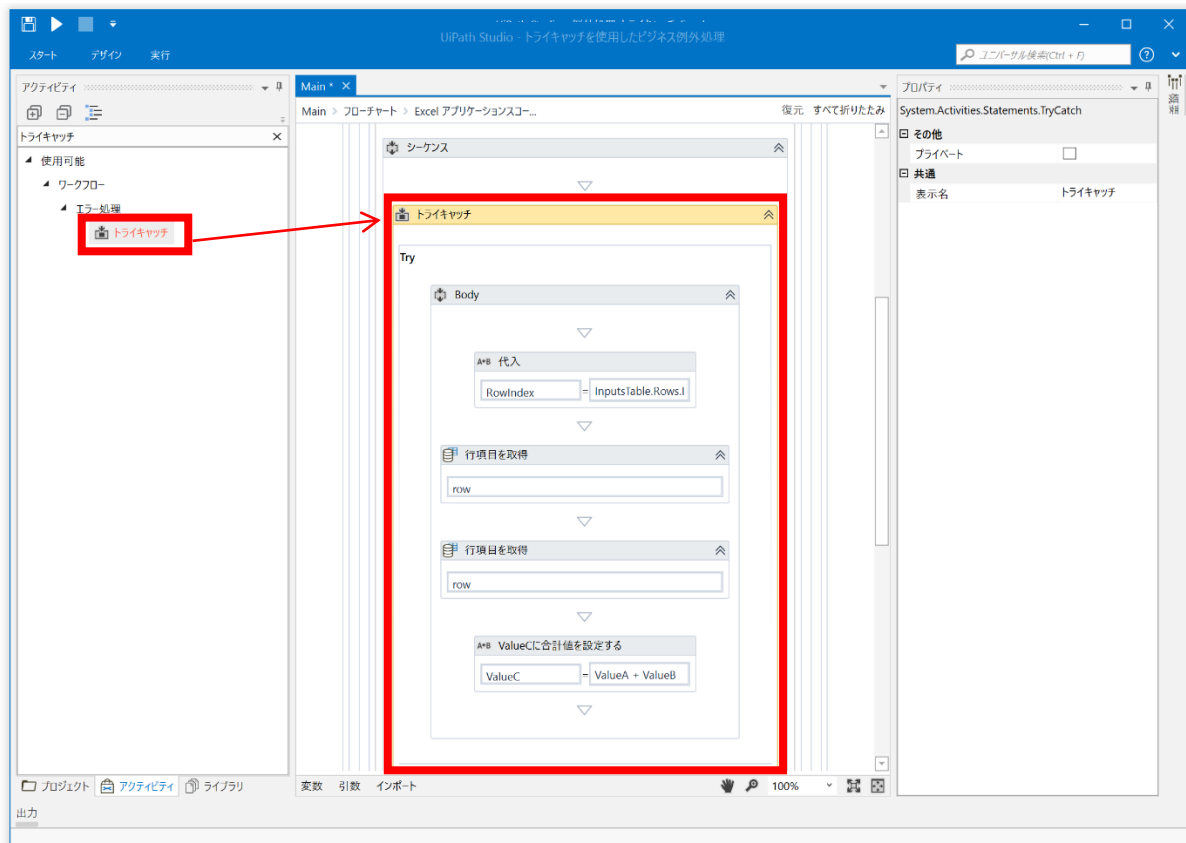


4. [トライキャッチ (Try Catch)] を検索し、[シーケンス (Sequence)] の中に配置します

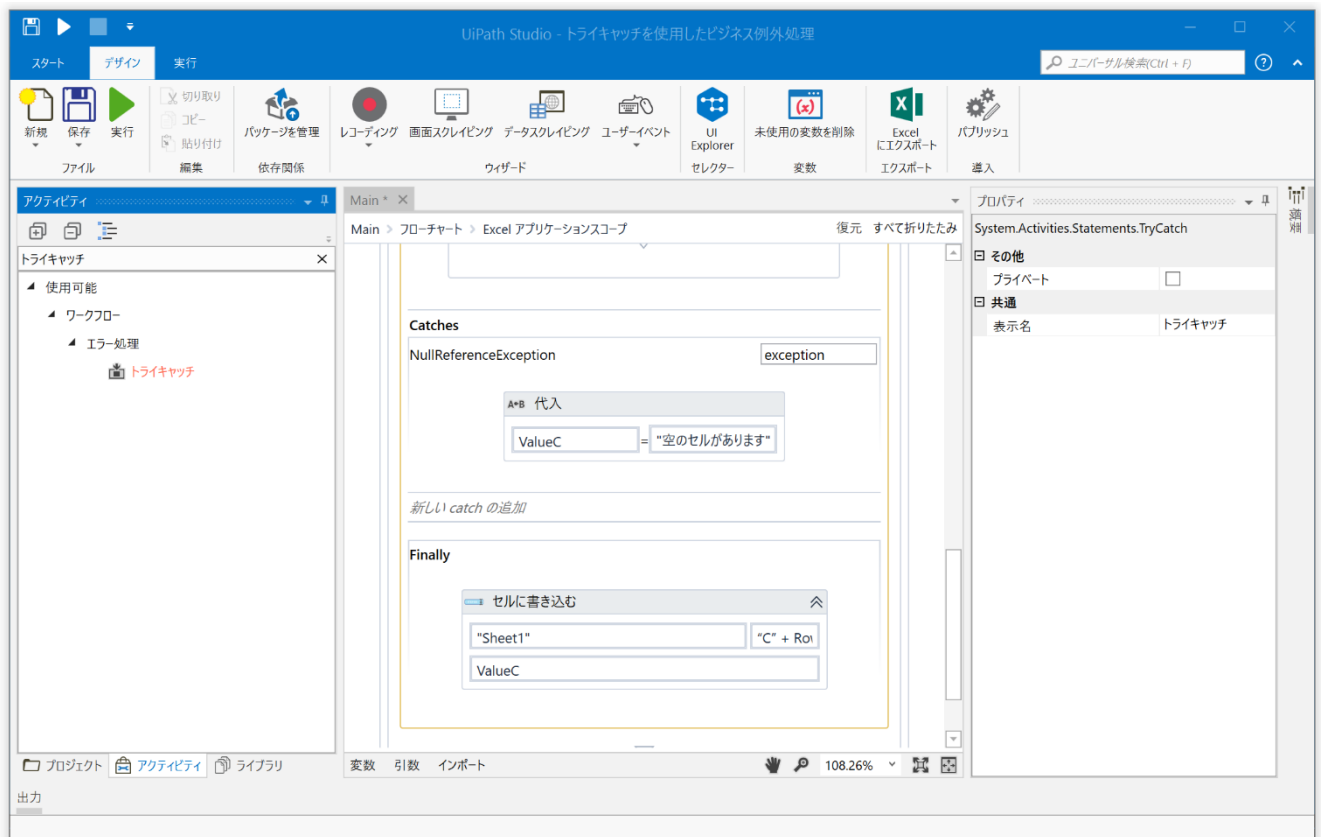
5. トライキャッチ (Try Catch)] の「Try」ブロックの中に、[代入 (Assign)] を追加して、右辺に「**InputsTable.Rows.IndexOf(row) + 1**」と入力し、左辺には、「**RowIndex**」という変数を作成します

6. [代入 (Assign)] の下に、[行項目を取得 (Get Row Item)] を2つ配置して、プロパティの設定で、列インデックス「**0**」の出力値には「**ValueA**」という変数名を作成し、列インデックス「**1**」の出力値には「**ValueB**」という変数名を作成します

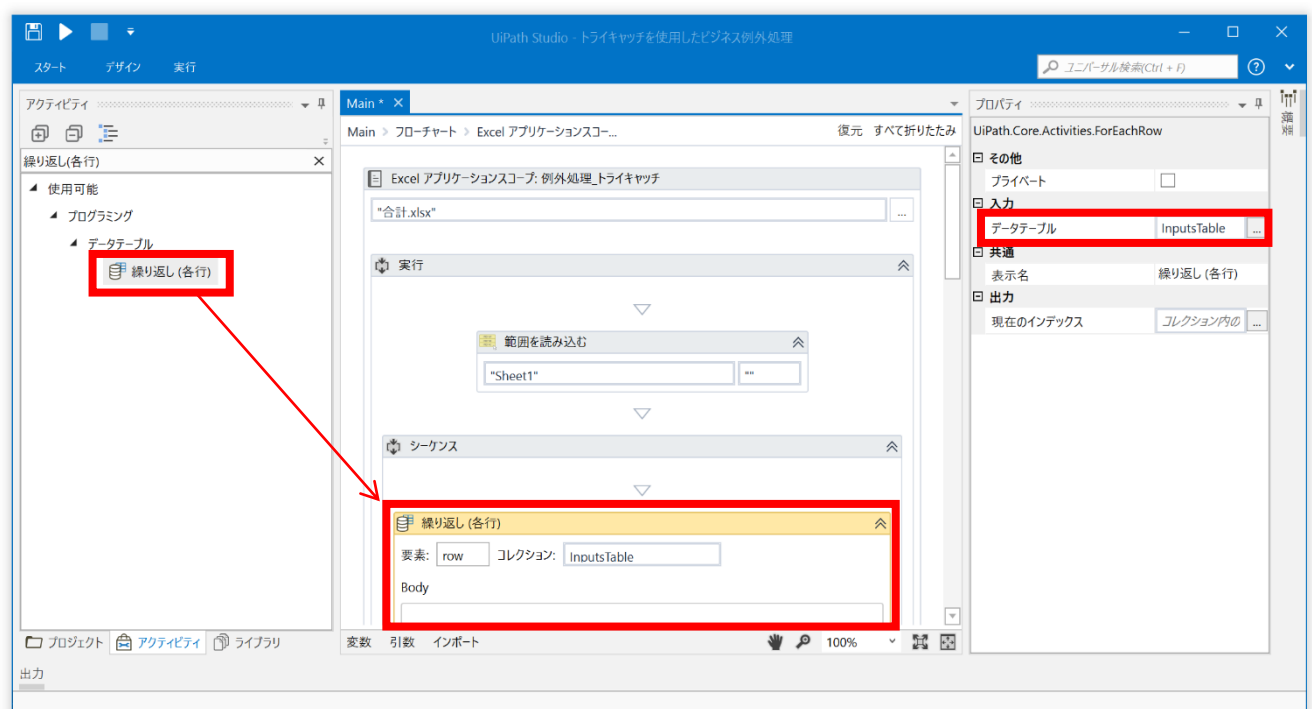
7. [行項目を取得 (Get Row Item)] の下に [代入 (Assign)] を追加し、右辺に「**ValueA + ValueB**」、左辺に「**ValueC**」という変数を作成します



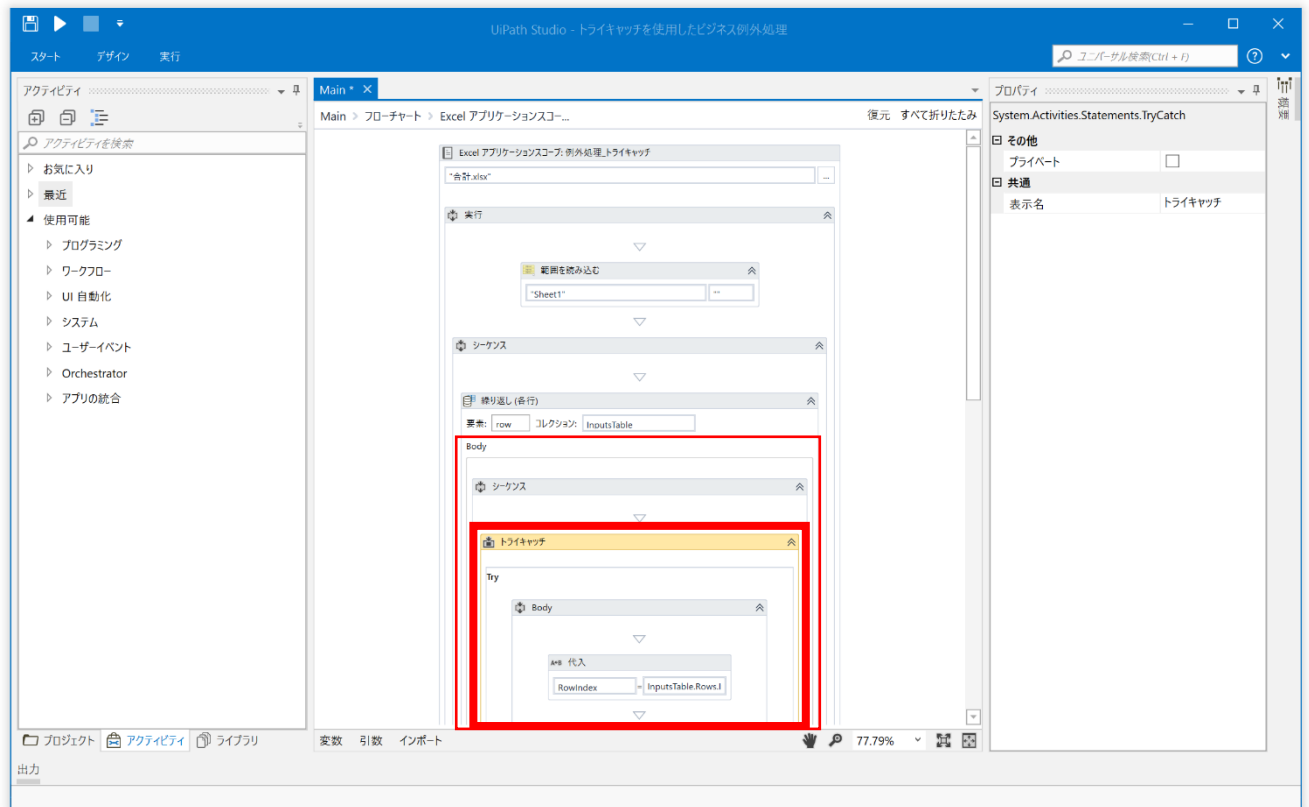
8. [トライキャッチ (Try Catch)] の「Catches」ブロックに、[代入 (Assign)] を配置し、右辺に「**空のセルがあります**」とし、左辺に変数「**ValueC**」を設定します
9. 「Finally」ブロックには、[セルに書き込む (Write Cell)] を配置し、シート名に「**Sheet 1**」、入力値に「**ValueC**」、範囲に「**C** + **RowIndex.ToString**」と設定します



10. [繰り返し(各行) (For Each Row)] を「3.」で配置した [シーケンス (Sequence)] の中に追加し、コレクションには、「**InputsTable**」を設定します



11. [繰り返し(各行) (For Each Row)] の「Body」内に「4.」～「9.」で設定した [トライキャッチ (Try Catch)] をドラッグ&ドロップします



12. 完成したワークフローは以下のようになります

