

変数とローカル変数の使い分け方

株式会社NTTデータ

目次

- ① ローカル変数とそれに関連する言葉について
- ② 変数とローカル変数の違い
- ③ ローカル変数を設定する方法
- ④ ローカル変数を使用したシナリオ例の説明①
- ⑤ ローカル変数を使用したシナリオ例の説明②
- ⑥ ローカル変数を使用したシナリオ例の説明③
- ⑦ 変数とローカル変数の使い分け方①
- ⑧ 変数とローカル変数の使い分け方②

ローカル変数とそれに関連する言葉について

ローカル変数とはサブルーチンに対して設定するものです。
ローカル変数を設定すると、サブルーチン終了時にローカル変数の値をサブルーチン開始前の値に戻してくれるという効果があります。

また、ローカル変数そのものではありませんが、ローカル変数の使い勝手に関わる言葉について2点説明します。

①引数

特定のローカル変数に呼び出し元から値を渡すことを「引数を渡す」と言います。
ローカル変数の使い方の一つとして呼び出し元から呼び出す際にその場面に応じた引数を渡すことで呼び出し先の動作を変更するといったことがあります。

②返り値

呼び出し先のサブルーチンを終了する際にサブルーチン先での動作結果などを指定した変数に返すことを「返り値を返す」と言います。
これによってその場面に応じた変数に結果を返し、呼び出し元で処理することができます。

変数とローカル変数の違い

通常、変数はシナリオのどこかで値が変更されたらシナリオ全体を通してその変更が適用されます。

一方、ローカル変数はサブルーチンごとに設定することでそのサブルーチン内で値が変更されてもそのサブルーチン(ローカル)内でのみ変更が適用され、サブルーチンが終了すると同時にサブルーチン実行前の値に戻るという違いがあります。

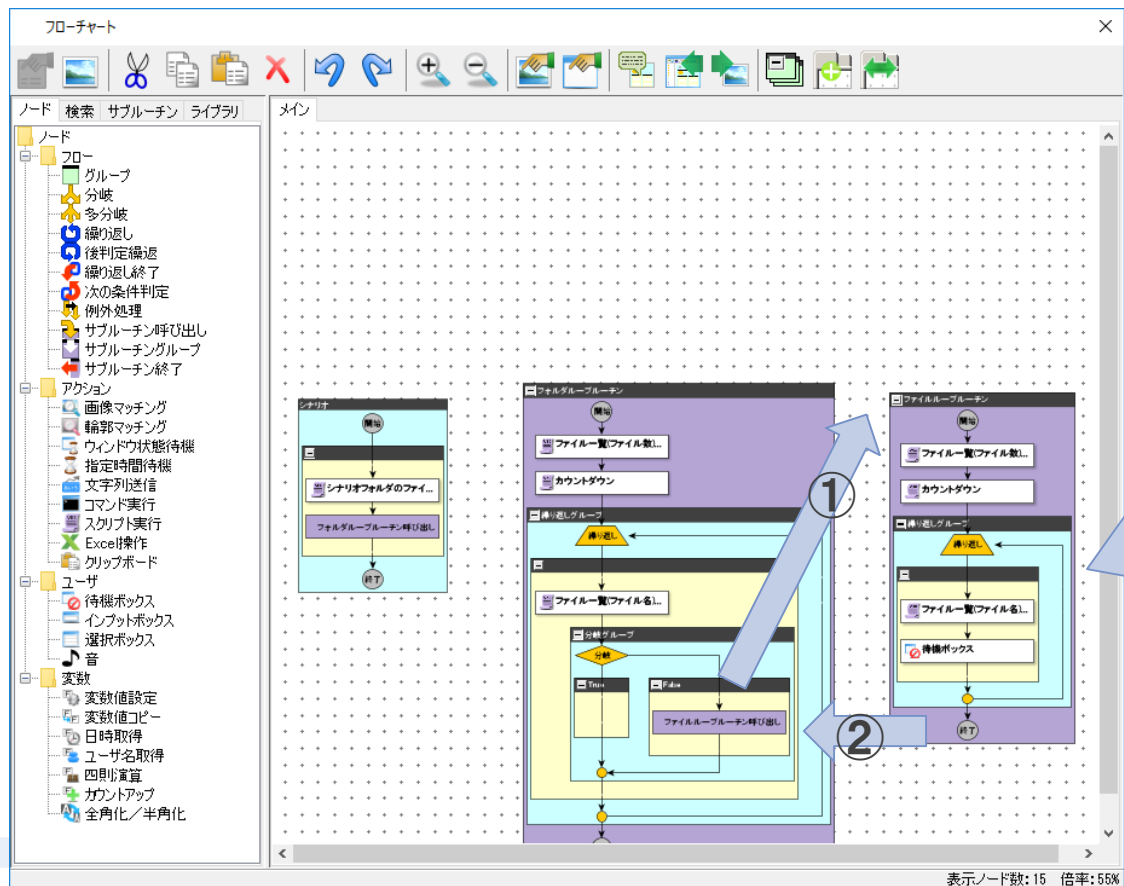
また、前のページで紹介した「引数」「返り値」が使用できるという点も違いとなります。

ローカル変数を設定する方法

- ① サブルーチングループのプロパティに対して変数の追加を行う
- ② 値を渡したい場合はサブルーチン呼び出しの際にローカル変数に渡したい値または変数を設定する(引数を渡す方法)
- ③ サブルーチン終了を使用することで値またはローカル変数を指定した変数に返すことができる(返り値を戻す方法)

ローカル変数を使用したシナリオ例の説明①

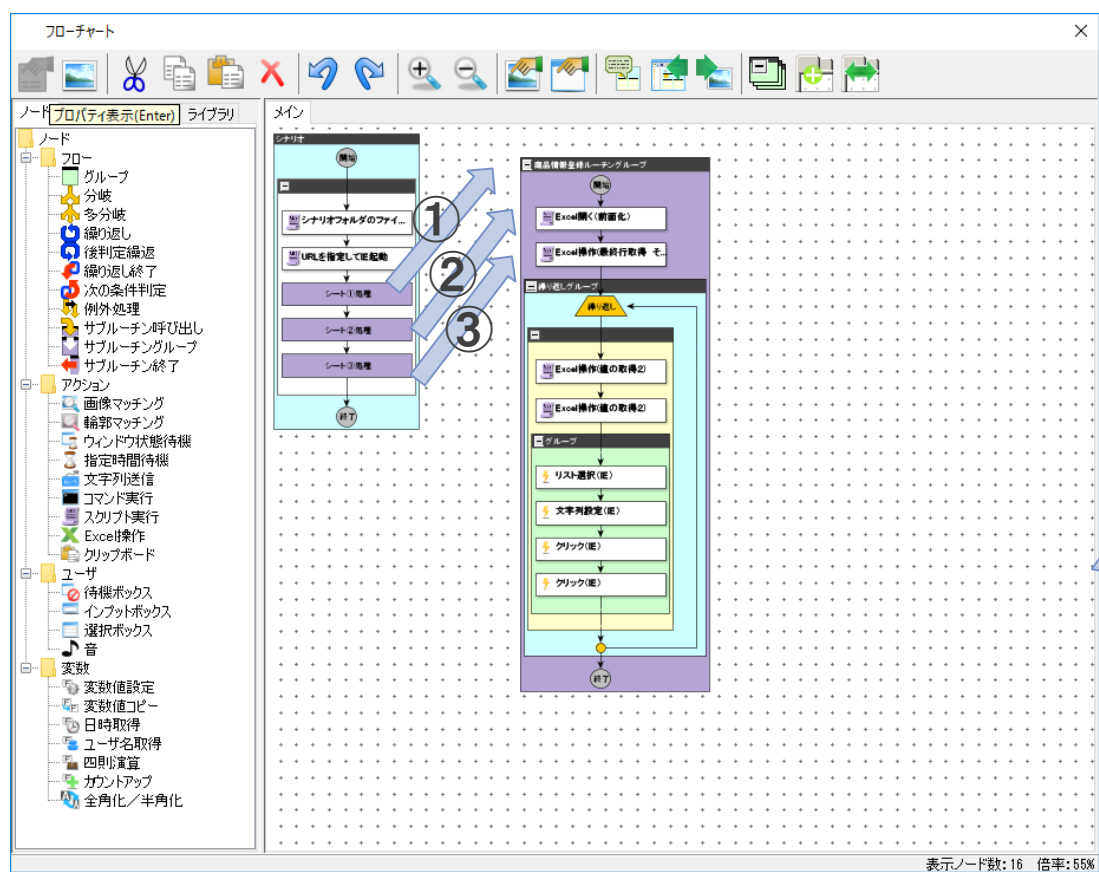
ローカル変数を利用することにより変数「カウンタ」、「ファイル数」を再利用している。
サブルーチン内で変数の値が変更されるが、その変更が呼び出し元に影響を与えないことがわかります。



①でサブルーチンを呼び出したのち「カウンタ」と「ファイル数」の値が変更されるが、
②でサブルーチンが終了した際には①の時点の変数の値に自動的に戻る。

ローカル変数を使用したシナリオ例の説明②

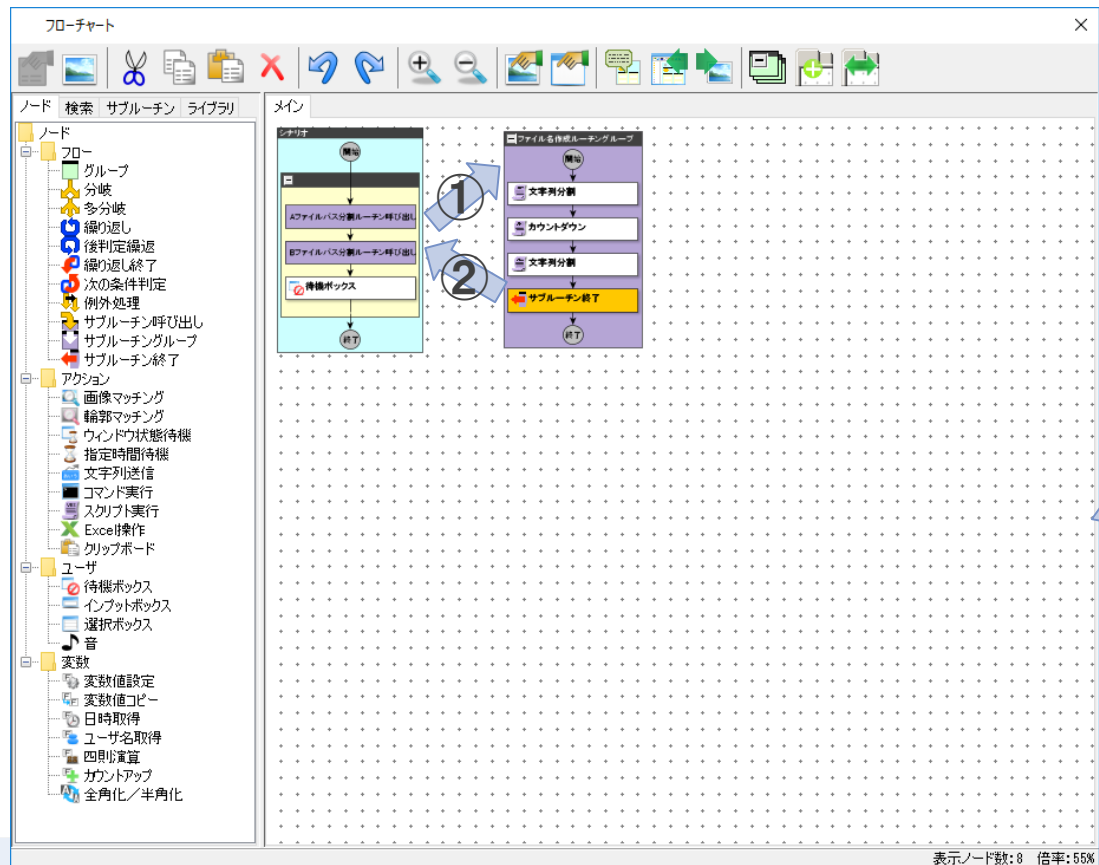
ローカル変数を利用し、引数を渡すことでフォーマットが異なる
エクセルシートから取得したい情報を狙って柔軟に取得できるようになっています。



①②③でそれぞれ別の値を
引数として渡すことで
同じサブルーチンでも別の動作を
できるようになっています。

ローカル変数を使用したシナリオ例の説明③

ローカル変数を利用し、呼び出し元から渡された引数のファイルパスに対してファイル名を取得できるようにしている。
また、結果として取得できたファイル名を呼び出し元で指定した変数に返すようにしている。



①で引数を渡すことでサブルーチンに渡すファイルパスを操作し、
②で「サブルーチン終了」を使用することでそれぞれ別の変数に返り値を返すことができます。

変数とローカル変数の使い分け方①

変数をローカル変数として設定すると以下のメリットがあります。

- ・サブルーチン内で変数の値が変化した場合サブルーチン終了時に呼び出し前の値に戻る
⇒変数の使いまわしがしやすくなる。
- ・サブルーチン呼び出し時にローカル変数に値を受け渡すことができる。
⇒サブルーチンの汎用性を高くすることができる。
- ・そのサブルーチンが必要とする変数を明確にできる。
⇒例えばそのサブルーチンが動作するためには、
エクセルのファイルパスが必要であるなどといったことが
明確にできるため、可読性を上げることができる。

変数とローカル変数の使い分け方②

ローカル変数とはサブルーチンの中で値を変更してもシナリオ全体に影響を与えないものです。

そうでない変数については値を変更したらシナリオ全体に影響を与えてしまいます。

そのため、基本的な変数はローカル変数にする必要はありませんが、特定のサブルーチン内でのみ変数の値を自由に変更したい場合、または特定のサブルーチンの動作を柔軟に変更したい場合等にローカル変数を活用していきましょう。

