

Visual Basic 第2回

Visual Basicの条件分岐1

テキスト P.127 ~136

If文

基本形

If 条件式 **Then**

処理

Else If 条件式 **Then**

処理

Else

処理

End If

Javaにはないキーワード 忘れないように注意

基本はJavaと同じ。

当然、入れ子も可能。

論理演算子

And :かつ

Or :または

AndAlso :かつ※

OrElse :または※

※は前の式の時点で判定ができる場合は後ろの式を判定しない

Visual Basicの条件分岐2

テキスト P.148 ~152

Select文

基本はJavaのSwitch文と同じ

基本形

: (コロン)やbreakキーワードは不要

Select Case 判定対象

Case 値1

処理

Case 値2

処理

・

・

・

End Select

応用

Case文の後ろの判定する値のところには、範囲の指定をすることが可能。

EX)

Case 4 To 6

処理

Case 7 To 9

処理

EX2)

Case Is < 60

処理

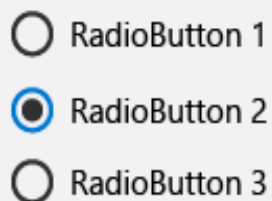
Case Is < 70

処理

ラジオボタンと文字列のフォーマット

ラジオボタン

画像



どれか一つしか選択できない選択肢を選ばせるときに使うコントロール。

他のを選択すると、前に選んでいたチェックは外れる。

文字列のフォーマット

通常、プログラミング言語では書式を指定して、文字列に変換することができる。

VisualBasicで整数を3桁のカンマ区切り、円を最後に着けて文字列にする場合は、以下のように記述する。

`IntegerData.ToString("#,##0円")`

参考 <https://dobon.net/vb/dotnet/string/inttostring.html>

HINT

- ・ イベントハンドラーの宣言文の最後の部分、どのイベントが起こった時に実行するかの部分にカンマ区切りで連続して記述することで、複数のイベントに対応するようになる。
- ・ ラジオボタンのチェックを変えたときというイベントは、
「ラジオボタン名.CheckedChanged」
- ・ 画面が読み込みされたとき（表示されたとき）というイベントは、
「MyBase.Load」

EX.エラーメッセージを表示する

通常、画面上で間違った入力や必須項目の選択をしていなかった場合、赤字で教えてくれますよね？あれをエラーメッセージといいます。

今回作った画面では、ユーザーがテキストを入力する箇所はなかったですが、前回のConvertTimeではどうでしょう？

数字(正確には整数)以外の文字がTextBoxに入力されているとエラーが起きて、フリーズしてしまいますよね。

Javaで、エラーハンドリングとして、try…Catchの構文をやったと思いますが、実はIF文でも入力チェックぐらいならできるんです。(むしろ入力チェックはtry…Catchではなく、IF文を使うことが多いです。)

画像例

アカウント作成

UserID

Usernameを入力してください

Email

Emailを入力してください

Password

Passwordを入力してください

Password Confirmation

登録

EX.エラーメッセージを表示する

せっかく画面と処理と両方作っているので、入力チェックとエラーメッセージ表示にチャレンジしてみましょう。

課題

前回作った**ConvertTime**のプロジェクトを改良して、整数以外が入力されていた場合にポップアップでエラーメッセージを表示されるようにしてみましょう。

HINT

- ・入力された文字列が整数かどうかの判別には**Integer.TryParse()**メソッドを使用する。
(詳細はググってください。)
- ・ポップアップの表示は以下のメソッドを使用する。

`MessageBox.Show(ポップアップ本文,ポップアップのタイトル,MessageBoxButton.OK,MessageBoxIcon.Error)`

追加演習

演習

テキストP. 167の練習問題Aにチャレンジしてください。終わった人は、練習問題Aに入力チェックを追加してみましょう。

それも終わった人は、なぜtry…Catchじゃなく、IF文で入力チェックをすることが多いか考えてみてください。