

## JRA-VAN 競馬ソフト作成体験教室

### Lesson-2 : JV-Data をダウンロードする

当コーナーでは、Microsoft Visual Basic 2019 Professional Edition(以下 VB 2019 と省略)で「JRA-VAN Data Lab.」サービス対応の競馬ソフトを作成していく過程をステップアップ形式で解説していきます。

今回から、実際に JV-Data をダウンロードして内容を表示する仕組みを実装していきます。今回は、細かいことは気にせず、とにかく JV-Data をダウンロードしてみます。

JRA-VAN Data Lab.では競馬ソフトを、**過去の全てのデータを保持する「蓄積系ソフト」**と**該当週のレース予想に十分なデータのみを保持する「非蓄積系ソフト」**の2つに大きく分類しています。この分類は JRA-VAN Data Lab.対応の競馬ソフトを作成するに上で非常に重要な点ですので、事前に JRA-VAN SDK に同梱されている「JRA-VAN Data Lab.仕様書」の「3.3 JV-Data 取得方法の概念」の項で確認しておいてください。ちなみに今回から数回にわたる実装では、「非蓄積系ソフト」のイメージで作成していきます。

#### 【 今回の目標 】

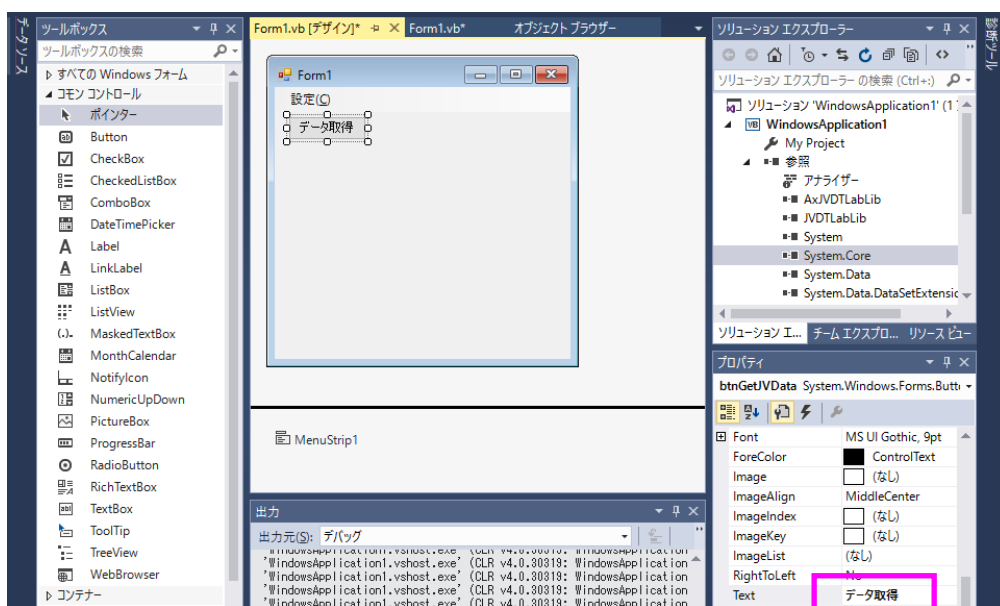
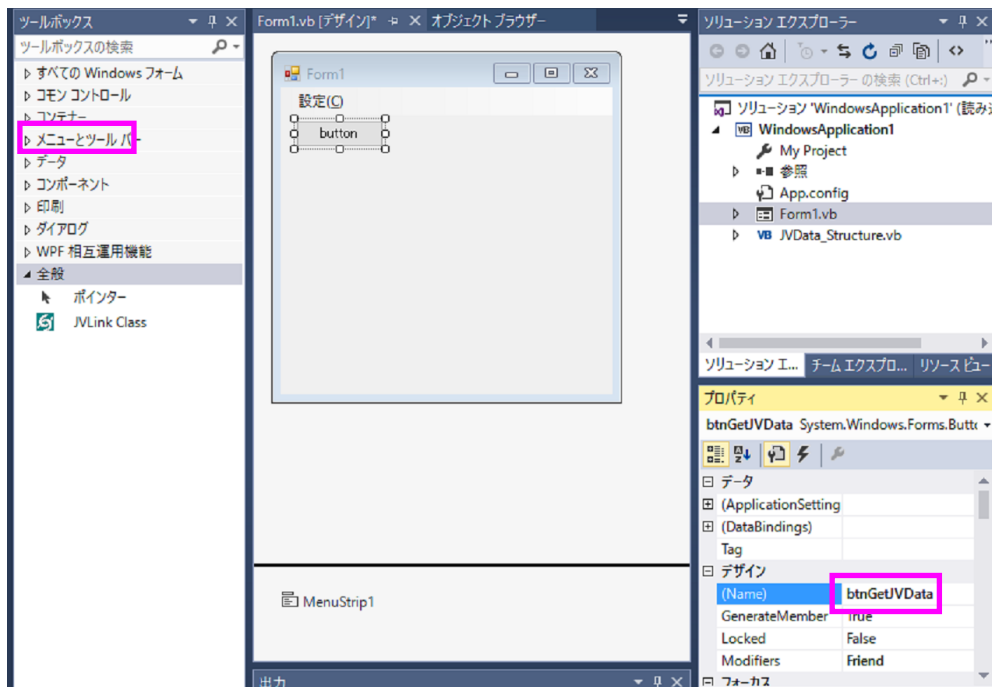
JV-Data をダウンロードしてみる。

JV-Link に実装されている以下のメソッドの使い方を理解する。

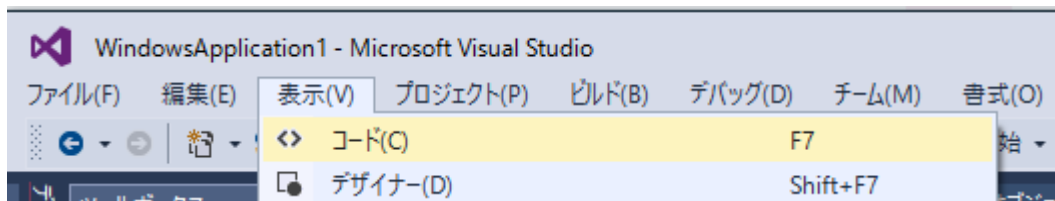
- JVInit メソッドで JV-Link の初期化を実行
- JVOpen メソッド JV-Data の取得を要求
- JVClose メソッド JV-Data の読み込み処理を終了

【 やってみよう 】

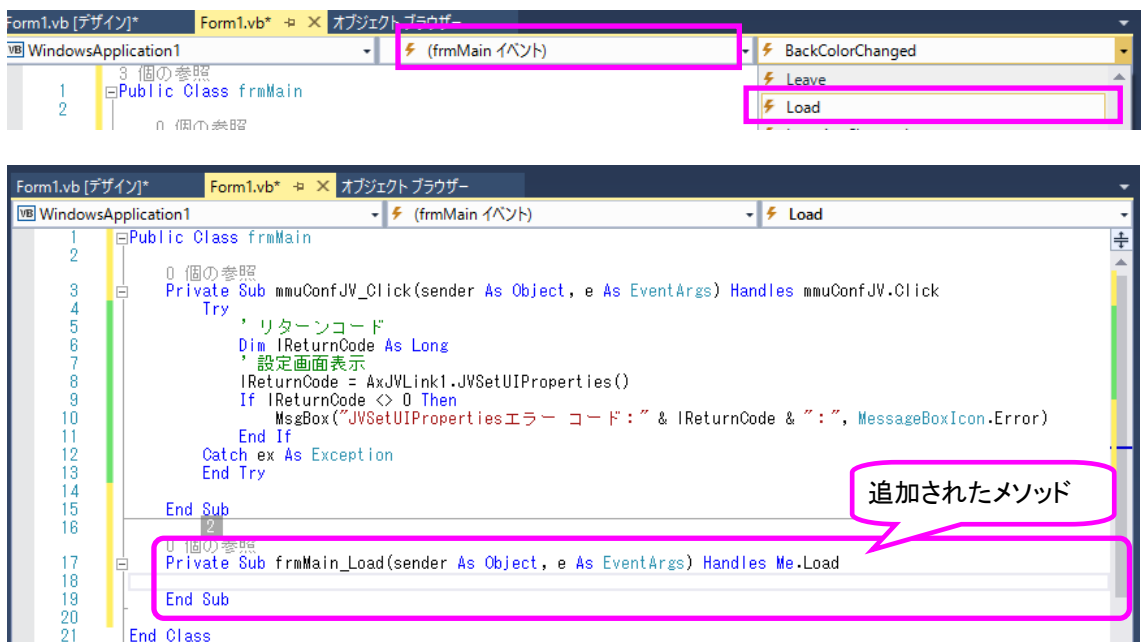
- ① Lesson-1 までを実装したフォーム(frmMain)を含むプロジェクトを開く。  
(前回のレッスンの続きから開始するのであれば、この作業は必要ありません。)
- ② コモンコントロールからフォームに「データ取得」ボタンを追加します。  
プロパティウィンドウで「(Name)」を「btnGetJVData」に、「Text」を「データ取得」にそれぞれ変更します。



- ③ JV-Link を用いて競馬データ (JV-Data) を取得するためには、JVInit メソッドによる JV-Link の初期化が必要です。それでは、JVInit メソッドをフォーム起動時に実行するようにコーディングしてみましょう。まずは、メニューから「表示」→「コード」を選択します。



- ④ コードエディタウィンドウが表示されるので、ウィンドウ上部の左のリストボックスから「(frmMain イベント)」を、右のリストボックスから「Load」を選択します。すると、コードエディタウィンドウに frmMenu フォームがロードされたときに自動的に呼び出されるメソッド「frmMenu\_Load」が追加されます。



- ⑤ このメソッドの中に、次のソースコードを追加します。 [ソースコード 004-01]

```
Private Sub frmMenu_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles MyBase.Load  
    Dim sid As String  
    Dim IReturnCode As Long  
  
    ' 引数設定  
    sid = "Test"  
  
    ' JVLink 初期化  
    IReturnCode = Me.AxJVLink1.JVInit( sid )  
  
    ' エラー判定  
    If IReturnCode <> 0 Then  
        MsgBox("JVInit エラー コード : " & IReturnCode & " : ", MessageBoxIcon.Error)  
        Me.Cursor = System.Windows.Forms.Cursors.Default  
        Exit Sub  
    End If  
  
End Sub
```

JVInit メソッドを呼び出す際には、sid という文字列を指定します。これは、呼び出し元の競馬ソフトを特定するためのもので、今回は "Test"を指定しています。

[ ワンポイントメモ ]

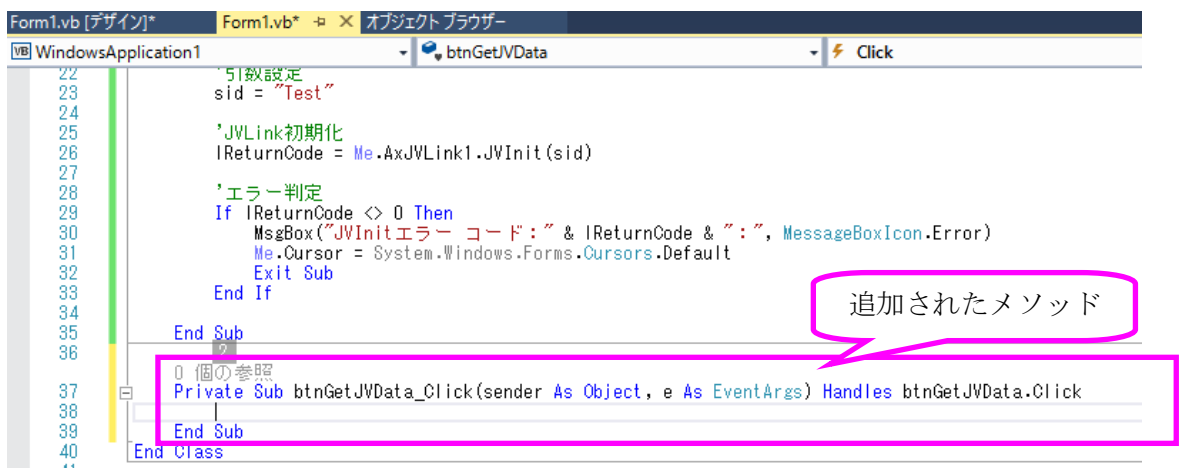
VB 2019 においてコントロールへのオブジェクト名の付け方は、

- ・ 名前として使える文字は、アルファベット、数字、アンダースコア( )
- ・ 名前の先頭文字はアルファベット、アンダースコア( )

という規則さえ守っていれば、どんな名前をつけようが自由です。

しかし、むやみに名前を付けてしまうと、(大きなプログラムでは特に) どんなコントロールなのかわけがわからなくなってしまいます。これに対して、コーディング全体を通して何らかの一貫性をもたせることによって、コーディングの保守性を向上させることを図ります。VB 2019 では通常、名前の先頭にコントロールの種類がわかるように3文字の識別文字(プリフィックス)を付けることによって、この一貫性を維持します。今までに出てきた「frmMain」の「frm(フォーム)」や「mnuConfJV」の「mnu(メニュー)」や「btnGetJVData」の「btn(ボタン)」などがこの識別文字に該当します。

- ⑥ さて、いよいよ「データ取得」ボタンクリック時のコーディングです。「データ取得」ボタンをダブルクリックすると、コードエディタウィンドウにこのボタンがクリックされた時に実行されるメソッド「btnGetJVData\_Click」が追加されます。



```
Form1.vb [デザイン]*  Form1.vb*  オブジェクトブラウザー
WindowsApplication1  btnGetJVData  Click
22  '51数設定
23  sid = "Test"
24
25  'JVLink初期化
26  IReturnCode = Me.AxJVLink1.JVInit(sid)
27
28  'エラー判定
29  If IReturnCode <> 0 Then
30  MsgBox("JVInitエラー コード:" & IReturnCode & ":", MessageBoxIcon.Error)
31  Me.Cursor = System.Windows.Forms.Cursors.Default
32  Exit Sub
33  End If
34
35  End Sub
36
37  0 個の参照
38  Private Sub btnGetJVData_Click(sender As Object, e As EventArgs) Handles btnGetJVData.Click
39  End Sub
40
41  End Class
```

追加されたメソッド

- ⑦ このメソッドの中に、次のソースコードを追加します。[ソースコード 004-02]

```
Private Sub btnGetJVData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles btnGetJVData.Click

    Dim IReturnCode As Long

    Try
        Dim strDataSpec As String          '' 引数 JVOpen:ファイル識別子
        Dim strFromTime As String          '' 引数 JVOpen:データ提供日付
        Dim IOption As Long                '' 引数 JVOpen:オプション
        Dim IReadCount As Long             '' JVLink 戻り値
        Dim IDownloadCount As Long         '' JVOpen:総ダウンロードファイル数
        Dim strLastFileTimestamp As String '' JVOpen:最新ファイルのタイムスタンプ

        ' 引数設定
        strDataSpec = "RACE"
        strFromTime = "20050301000000"
        IOption = "2"

        ' JVLink ダウンロード処理
        IReturnCode = Me.AxJVLink1.JVOpen(strDataSpec, strFromTime, IOption, _
            IReadCount, IDownloadCount, strLastFileTimestamp)

        ' エラー判定
        If IReturnCode <> 0 Then
            MsgBox("JVOpen エラー：" & IReturnCode)
        Else
            MsgBox("戻り値：" & IReturnCode & vbCrLf & _
                "読み込みファイル数：" & IReadCount & vbCrLf & _
                "ダウンロードファイル数：" & IDownloadCount & vbCrLf & _
                "タイムスタンプ：" & strLastFileTimestamp)
        End If

    Catch
        Debug.WriteLine(Err.Description)
        Exit Sub
    End Try

    ' JVLink 終了処理
    IReturnCode = Me.AxJVLink1.JVClose()
    If IReturnCode <> 0 Then
        MsgBox("JVClose エラー：" & IReturnCode)
    End If

End Sub
```

最終形ではデータの読み込みまで行なうのですが、今回はデータのダウンロードまでとどめておきます。データ種別、データ読み出しポイント（年月日時分秒）とオプションを指定して、JVOpen メソッドを呼び出すと、該当ファイル数およびダウンロードする必要があるファイル数、該当ファイルのうち最新ファイルのタイムスタンプが返ります。この際、ファイルをダウンロードする必要がある場合は、ダウンロードが開始されます。

今週開催されるレースに関する情報（JV-Data）を取得することを想定すると、指定

するファイル種別は「RACE」（=レース情報）、オプションは「2」（=今週データ）となります（詳細は「JV-Data 仕様書」および「JV-Link インターフェイス仕様書」を参照）。データ読み出しポイントとしては、とりあえず「00000000000000」を指定しておきます。

### 【 確認しよう 】

それでは、実際に動かしてみましよう。

- ① メニューから「デバッグ」→「デバッグ開始」を選択し、プログラムを実行します。  
エラーが発生した場合は、タスク一覧にエラーメッセージが表示されるので、正しく修正してエラーが無くなるまでビルドします。
- ② フォーム上のメニューから「設定」→「JV-Link の設定」を選択します。
- ③ 「JV-Link 設定」ダイアログが表示されるので JRA-VAN より取得した「利用キー」を設定します。すでに他の Data Lab.対応ソフトを使用している場合は、修正不可の属性で「利用キー」が表示されます。

今回はダウンロードしたファイルを直接チェックするので、データの保存設定で「データを保存する」にチェックを入れて、保存するフォルダのパスも設定しておきます。「データ保存場所」には、テスト用に C:¥ProgramData¥JRA-VAN¥Data Lab を設定しておきます。



JV-Link 設定

利用キー設定

-  -  -  -

※現在の利用キーと異なるキーを使用する場合は、  
チェックをつけるとキーを再入力できるようになります。

[DataLab.利用キーの確認・再発行](#)

データの保存設定

データを保存する(推奨)  
 データを保存しない

データ保存場所:  ...

動画再生ブラウザ

Microsoft Edge  
実行ファイル格納場所:  ...

Google Chrome  
実行ファイル格納場所:  ...

Internet Explorer

**① Microsoft Edge、またはGoogle Chromeでのご利用を推奨いたします。**

高度な設定

高度な設定を行う

JV-LinkAgentポート番号(初期値 6531):

その他の設定

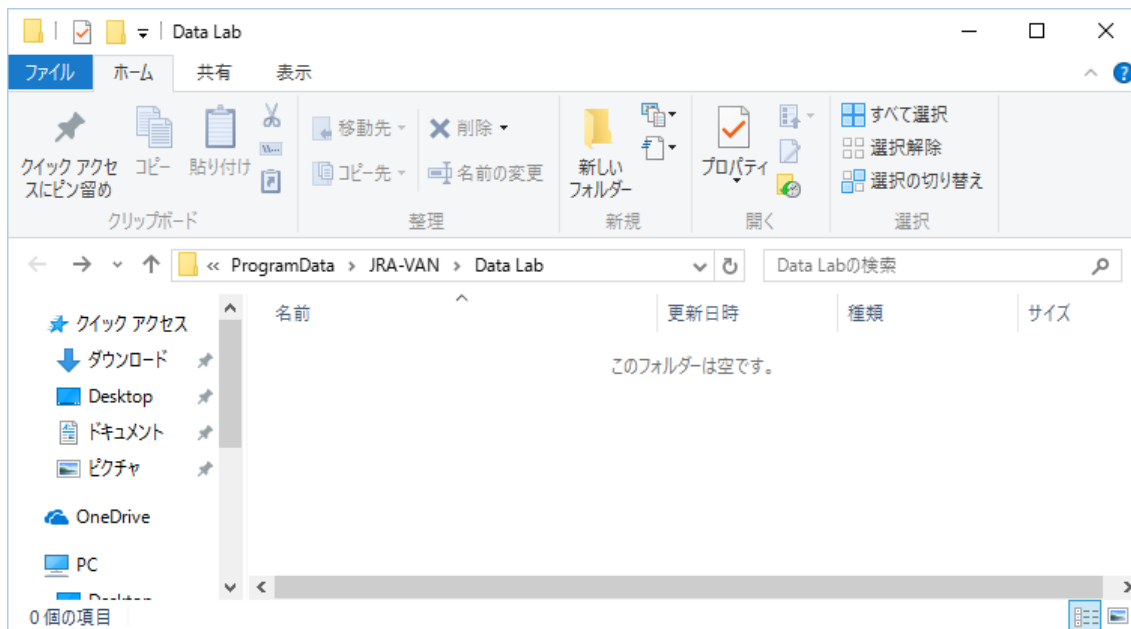
JRA-VANからのお知らせを表示する  
 払戻速報を表示する

状態取得

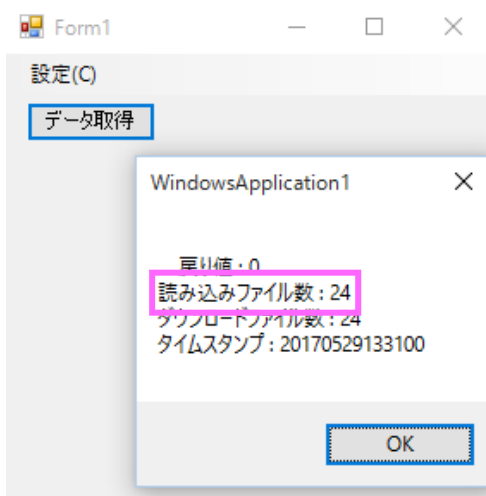
JVLink Version : 4.6.0

JRA-VAN より取得した  
利用キーを入力

- ④ 上記③で確認したデータ保存先フォルダの下にある「cache」フォルダ内にファイルが存在する場合は削除してください。なお、「cache」フォルダ自体が存在しない場合は、JV-Data を最初取得したときにフォルダが作成されます。

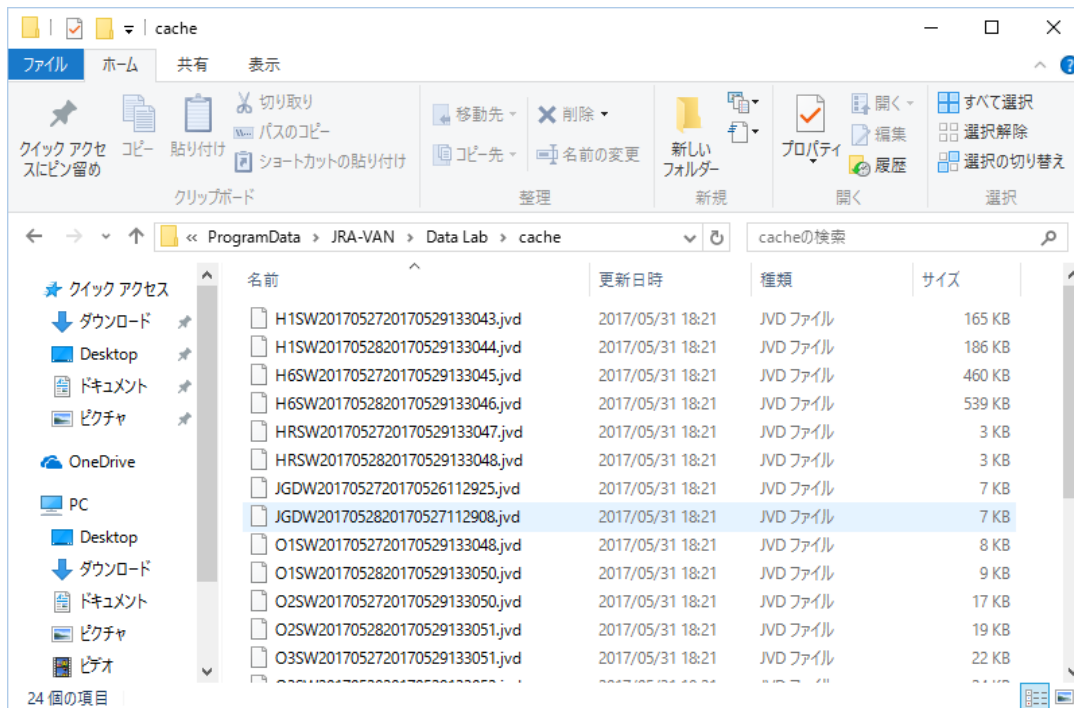


- ⑤ フォームの「データ取得」ボタンをクリックすると、メッセージボックスがポップアップします。メッセージボックスには、JVOpen メソッドの戻り値を表示するようにコーディングしてありますので、リターンコードや読み込みファイル数、ダウンロードファイル数、最新読み込みファイルのタイムスタンプが表示されるはずです。各戻り値の詳細についてはJV-Link インターフェース仕様書に記載されていますので、そちらを参照してください。



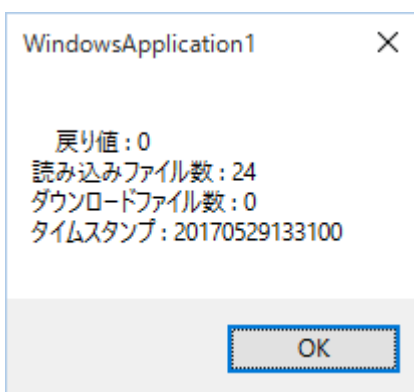
図の例では、読み込むべきファイルが 24 ファイルあり、そのうち実際にダウンロードするファイルが 24 ファイルあることを表しています。

④で確認した空の「cache」フォルダにファイルがダウンロードされていることも確認しておきましょう。メッセージボックスの「ダウンロードファイル数」に表示されている数値分のファイルが「cache」フォルダにダウンロードされます。



- \*1)ダウンロードされたファイル名の命名規則などは、特に意識する必要はありません。
- \*2)ダウンロードされたファイルの内容は暗号化されているため、テキストエディタなどで開いても、解読することはできません。データの内容は、次回以降に解説する「JVRead」メソッドを使って取得します。

⑥ メッセージボックスを「OK」で閉じた後、確認のためもう一度「データ取得」ボタンをクリックしてみましょう。



⑤と同様にメッセージボックスがポップアップしますが、今度は「ダウンロードファイル数」が0になっているはずですが、読み込むべきファイルが既にダウンロード済みなので、ダウンロードする必要があるファイル数は0ということになります。

では「cache」フォルダのファイルをどれか1つ削除して「データ取得」ボタンをクリックするとどうなるでしょうか？

もうお解りですね。ダウンロードファイル数は1となり、削除したファイルがダウンロードされます。

[ ワンポイントメモ ]

**JV-Link** は必要なファイルがローカルに存在するかどうかをチェックし、存在しなければダウンロードを実行します。このため、**JRA-VAN Data Lab.**サービスに対応した競馬ソフトを複数使っている場合でも、ファイルのダウンロードは1回で済みます。