

## JRA-VAN 競馬ソフト作成体験教室

### Lesson-6 : JV-Data の内容を読み出す 応用編 1 (ADO.NET の利用)

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。

前回までで、実際に JV-Data をダウンロードし、その内容を一部コード変換を行った上で表示するといった仕組みを実装しました。今回は、ADO.NET(\*)を用いてレース詳細と関連する馬毎レース情報の内容を表示してみましょう。

※ADO.NETは、.NET Framework クラスライブラリに含まれるデータアクセスを行うためのクラスです。  
詳細については、マイクロソフトの MSDN ライブラリなどをご参照ください。

#### 【 今回の目標 】

ADO.NET を用いて取得した JV-Data をデータセットに格納し、その内容から必要な情報を取り出して表示する。

具体的には、JV-Data のレース詳細と馬毎レース情報のすべてをデータセットに一旦格納します。その後、データセットのレース詳細の一件を取り出して表示エリアに表示します。さらに、その情報で馬毎レース情報から関連する情報を検索し、結果を表示エリアに表示します。

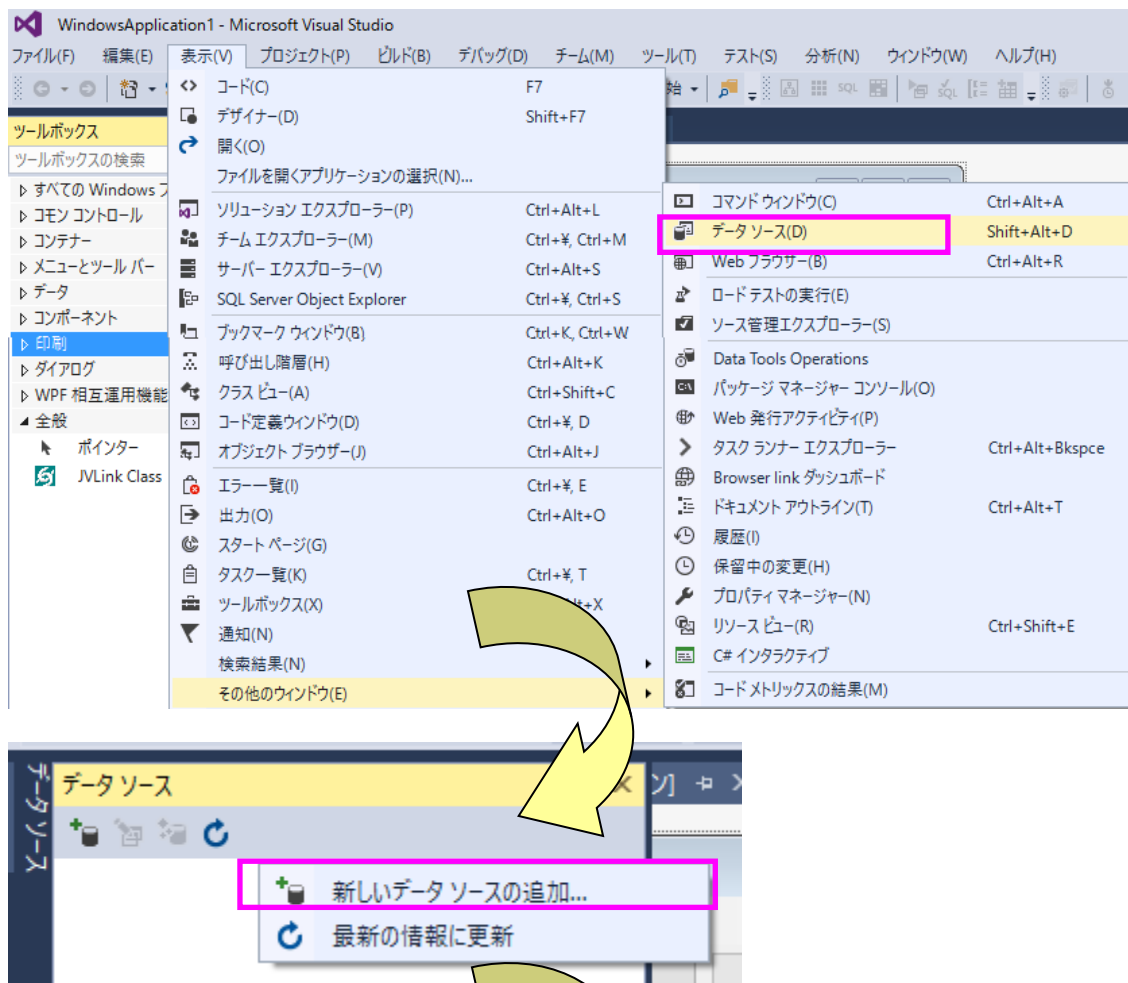
#### 【 やってみよう 】

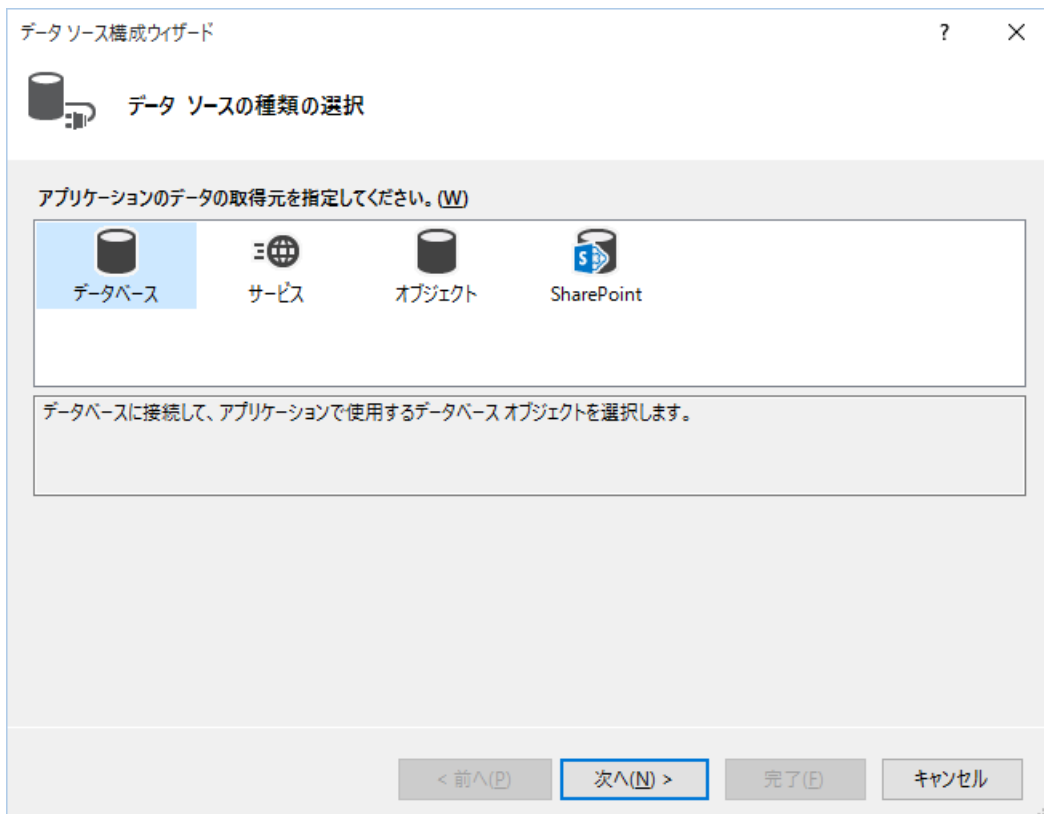
- ① JRA-VAN DataLab.サイトの競馬ソフト開発体験教室 Step.3 Lesson-6 の「レッスン用パーツ」をダウンロードし、解凍しておきます。
- ② Lesson-5 までを実装したフォーム(frmMain)を含むプロジェクトを開きます。  
(前回のレッスンの続きから開始するのであれば、この作業は必要ありません。)
- ③ ①で解凍したフォルダーの中から「CodeTable.csv」を作成中のプログラムフォルダにコピーします。(例では ¥Visual Studio Projects¥WindowsApplication1¥bin¥Debug フォルダにコピーしています。)
- ④ VB 2019 のデータソース構成ウィザードを利用し、プロジェクトにデータソースを追加します。  
以下でデータソースの追加手順を説明します。手順は多いですが、単純な設定ですので

一つ一つ進めていきましょう。

＜データソース追加手順＞

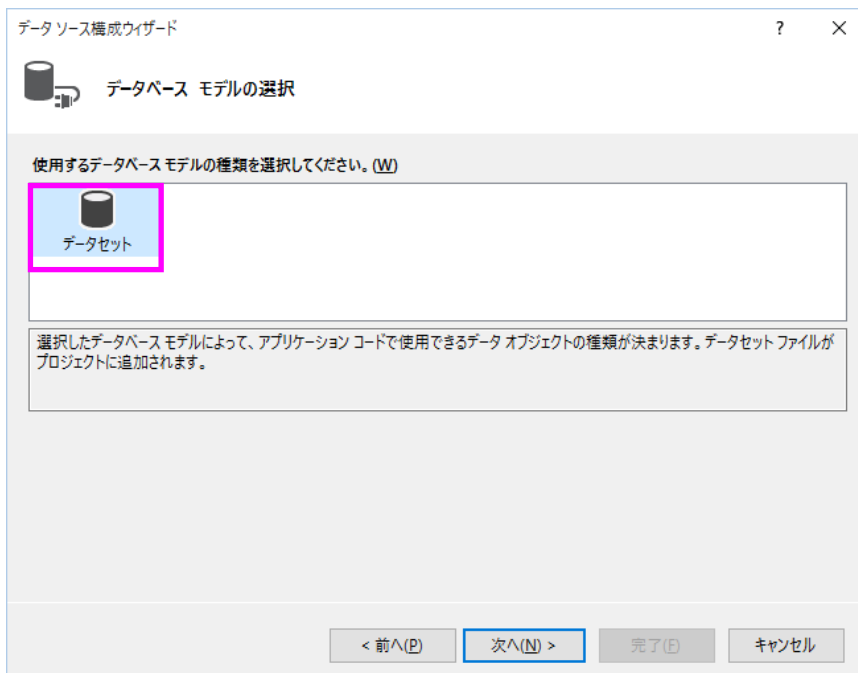
1. メニューバーの「表示(V)」→「その他のウィンドウ(E)」→「データソース(D)」をクリックしてデータソース画面を表示。データソース画面上で右クリックして「新しいデータソースの追加」を選択して「データソース構成ウィザード」を開きます。



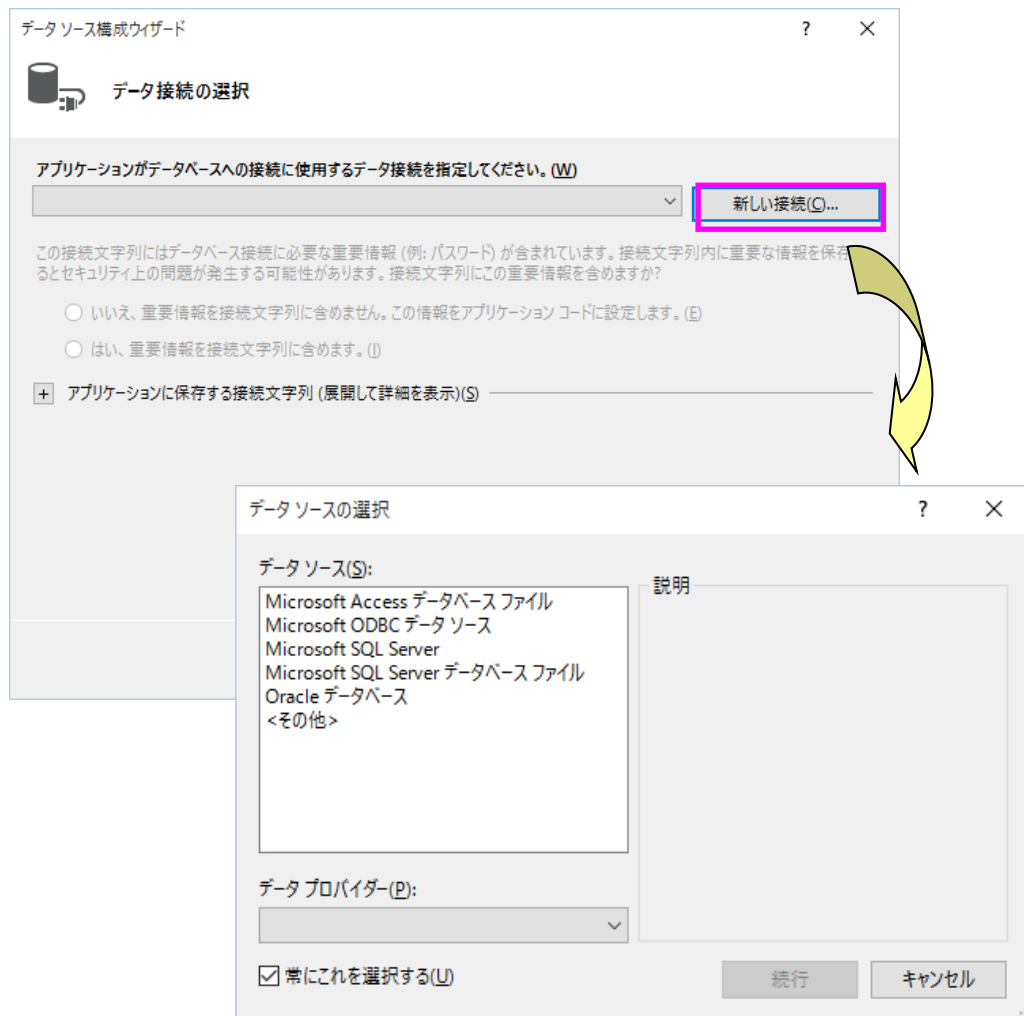


「データソースの種類を選択」で、「データベース」が選択されていることを確認し、「次へ(N)>」をクリックします。

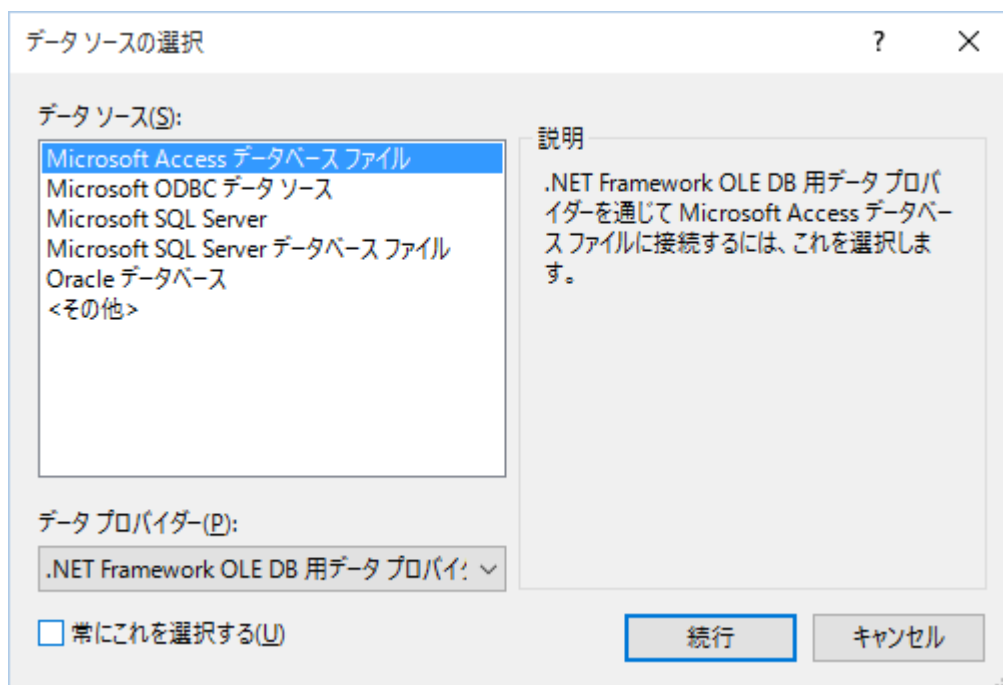
2. 「データベース モデルの選択」で「データセット」を選択します



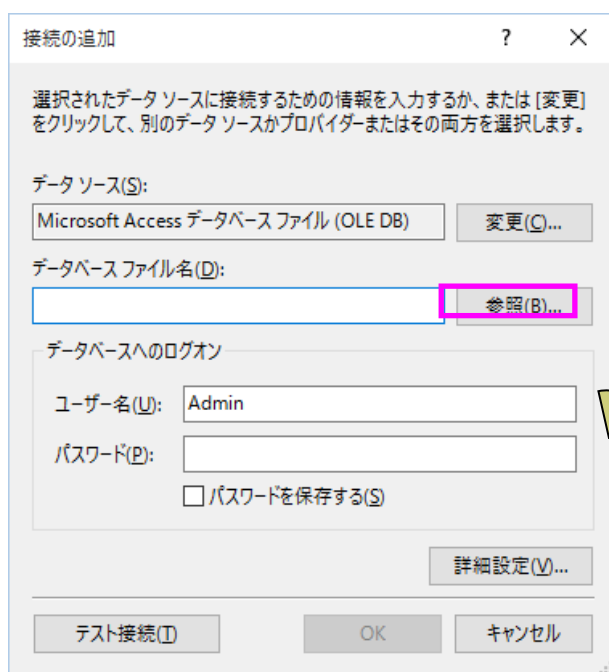
3. 「データ接続の選択」で、「新しい接続」をクリックし、「データソースの選択」ダイアログを開きます。

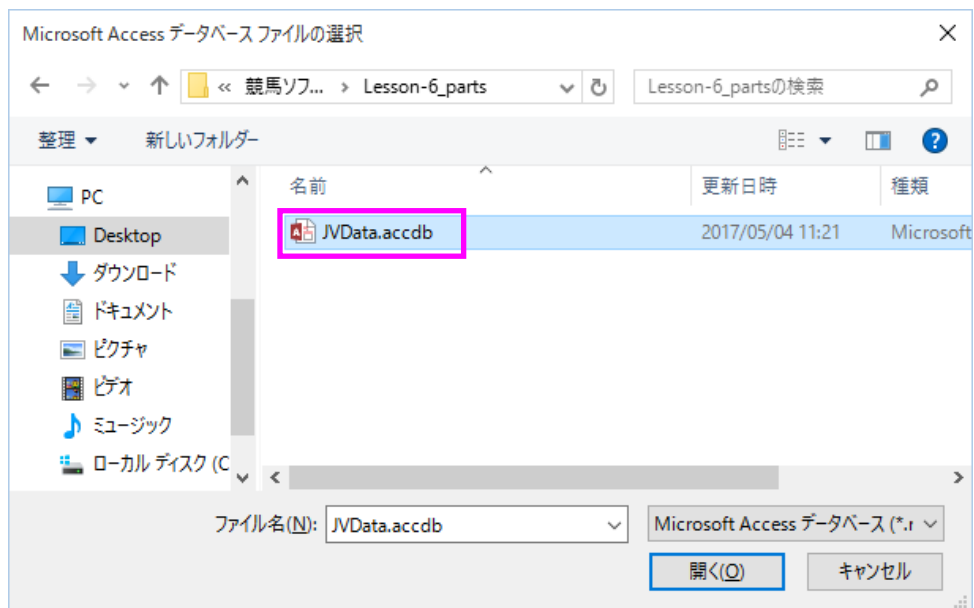


4. ここで「Microsoft Access データベースファイル」を選択し、「OK」をクリックしてください。

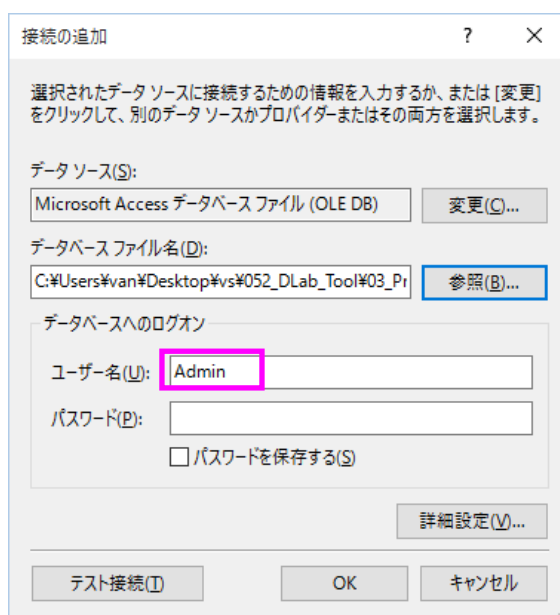


5. 次に「接続の追加」で、データベースファイル名を指定します。「参照(B)」をクリックし表示されるダイアログで、「レッスン用パーツ」の解凍後フォルダ内の「JVData.accdb」を選択して「開く(O)」をクリックします。





6. 「接続の追加」で、データソース、データベースファイル名の設定内容を確認し、「OK」をクリックしてください。

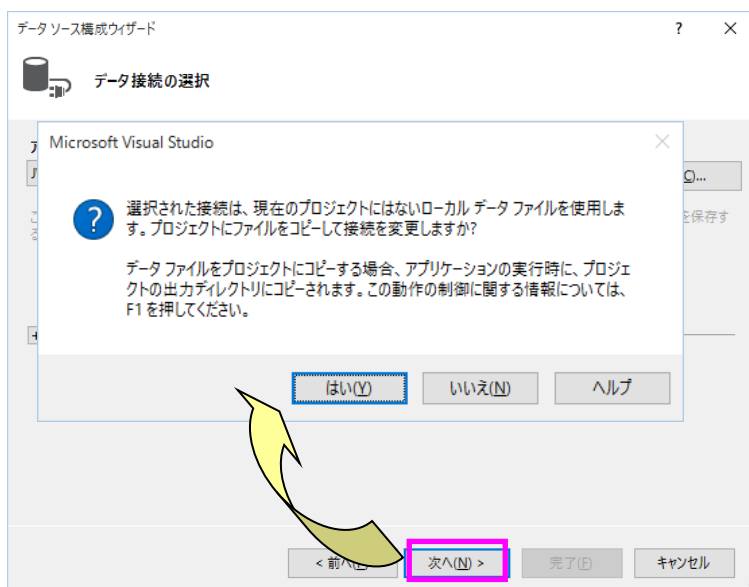


ここまでで、「データ接続の選択」で、アプリケーションがデータベース（MDB ファイル）への接続に使用するデータ接続(W)の追加が完了しました。

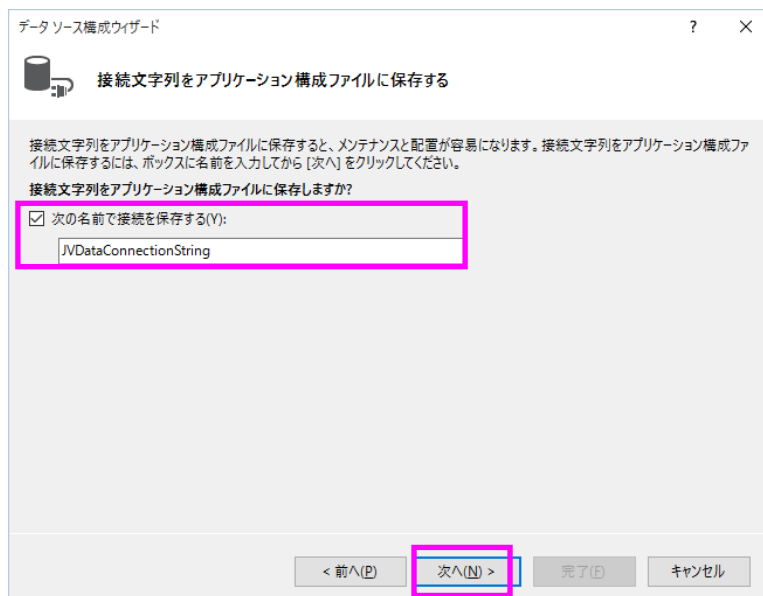
[ ワンポイントメモ ]

JVData.mdb では、ユーザー名、パスワードを使用しないため、「データベースにログオン」の設定は必要ありません。

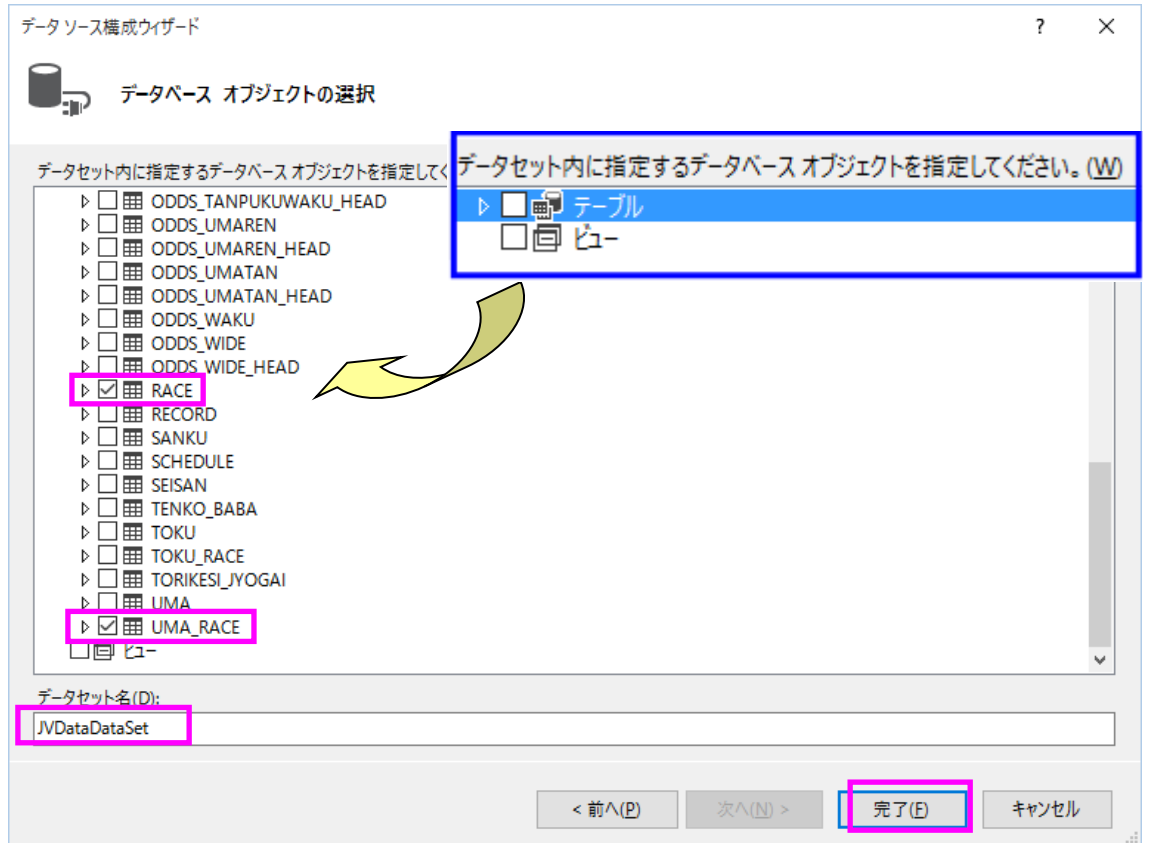
7. 続いて「データ接続の選択」で「次へ(N)>」をクリックします。JVData.mdb ファイルがプロジェクトフォルダに存在しないため、メッセージボックスが表示されます。「はい(Y)」をクリックして、プロジェクトフォルダにファイルをコピーし、次へ進みます。



8. 「接続文字列をアプリケーション構成ファイルに保存する」でチェックボックスがオンであること、接続文字列が「JVDataConnectionString」になっていることを確認し、「次へ(N)>」をクリックします。(通常、設定を変更する必要はありません)



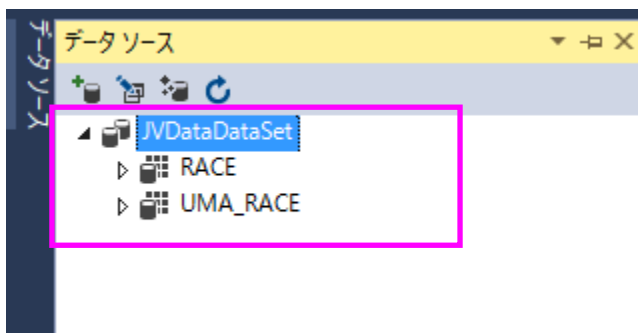
9. 「データオブジェクトの選択」で、「テーブル」の「▶」をクリックしてツリーを展開し、「RACE」と「UMA\_RACE」をチェックしてください。



「データセット名」が「JVDataDataSet」であることを確認し、「完了(F)」をクリックしてください。(通常、データセット名を変更する必要はありません。)



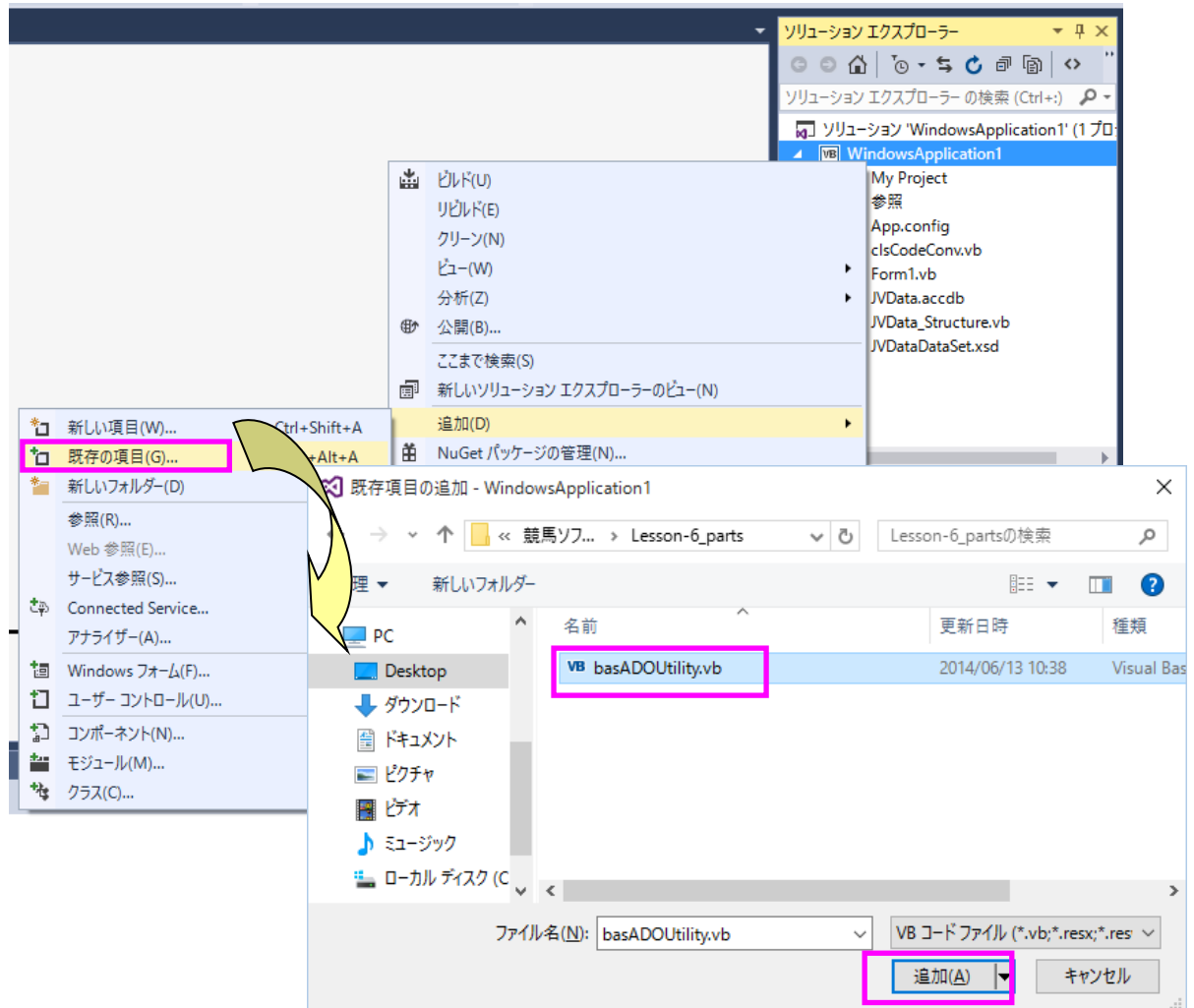
以上で、プロジェクトにデータソースが追加されました。  
データソースを確認するには、1 で表示したデータソース画面から可能です。



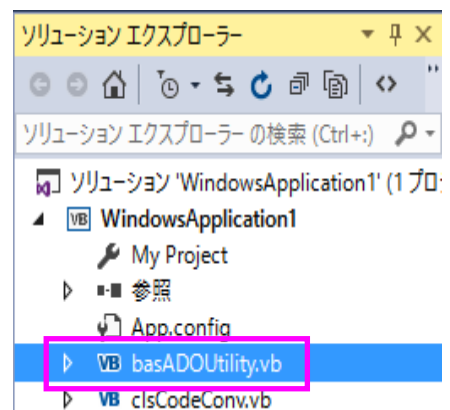
[ ワンポイントメモ ]

タブのクリックでソリューションエクス  
プローラーとデータソースを切り替える  
ことができます。

- ⑤ ソリューションエクスプローラーより右クリックにて「追加(D)」→「既存の項目(G)」を選択してダイアログを表示します。このダイアログで「レッスン用パーツ」の解凍後フォルダ内の「basADOUtility.vb」を選択して「追加(A)」をクリックします。



ソリューションエクスプローラーで「basADOUtility.vb」が追加されていることを確認してください。



- ⑥ Form1.vb をコードエディタウィンドウで開き、btnGetJVData\_Click メソッドに以下のコードを追加します。（灰色部分は前回コーディング済み）

```
Private Sub btnGetJVData_Click (ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetJVData.Click
    Dim lReturnCode As Long
    '' JV-Dataデータセット
    Dim jvdds As New JVDataDataSet

    Trv
        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 lOption As Long '' 引数 JVOpen:オプション
        Dim lReadCount As Long '' JVLink 戻り値
        Dim strLastFileTimestamp As String '' JVOpen: 最新ファイルのタイムスタンプ

        Const lBuffSize As Long = 110000 '' JVRead:データ格納バッファサイズ
        Const lNameSize As Integer = 256 '' JVRead:ファイル名サイズ
        Dim strBuff As String '' JVRead:データ格納バッファ
        Dim strFileName As String '' JVRead:ダウンロードファイル名
        Dim RaceInfo As JV_RA_RACE '' レース詳細情報構造体
        Dim RaceUmaInfo As JV_SE_RACE_UMA '' 馬毎レース情報構造体

        ' 表示ウィンドウの初期化
        rtbData.Clear ()

        ' 進捗表示初期設定
        tmrDownload.Enabled = False '' タイマー停止
        prgDownload.Value = 0 '' JVDataダウンロード進捗
        prgJVRead.Value = 0 '' JVData読み込み進捗

    (後省略)
```

Point1

[ソースコード 008-01]

・ Point1

JV-Data を格納するためのデータセットを定義します。

このデータセットに、レース詳細、及び馬毎レース情報を格納します。

馬毎レース情報の格納のため、馬毎レース情報構造体の定義を追加し、また、実行時に表示ウィンドウを初期化（内容をクリア）する処理も追加します。

- ⑦ 続いて JV-Data 読込時の処理に修正を加えます。(灰色部分は前回コーディング済み、薄い水色部分はコーディングの修正箇所)

```

(前省略)
Case Is > 0 ' 正常読み込み
' レコード種別IDの識別
If Mid(strBuff, 1, 2) = "RA" Then
' レース詳細のみ処理
' レース詳細構造体への展開
RaceInfo.SetData(strBuff)

' 読み込んだ情報をデータセットへ格納する
SetJVDataRaceDataSet(RaceInfo, jvdds) Point2

' データ表示
' rtbData.AppendText(
'   "年:" & RaceInfo.id.Year &
'   " 月日:" & RaceInfo.id.MonthDay &
'   " 場:" & obiCodeConv.GetCodeName("2001", RaceInfo.id.JyoCD, "3") &
'   " 回次:" & RaceInfo.id.Kai i i &
'   " 日次:" & RaceInfo.id.Nich i i &
'   " R:" & RaceInfo.id.RaceNum &
'   " レース名:" & RaceInfo.RaceInfo.Ryakusyo10 _
'   & vbCrLf)

ElseIf Mid(strBuff, 1, 2) = "SE" Then Point3
' 馬毎レース情報構造体への展開
RaceUmaInfo.SetData(strBuff)

' 読み込んだ情報をデータセットへ格納する
SetJVDataUmaRaceDataSet(RaceUmaInfo, jvdds)
Else
' レース詳細、馬毎レース情報以外は読み飛ばす
Call AxJVLink1.JVSkip()
End If
(後省略)

```

[ソースコード 008-02]

・ Point2

JV-Data 構造体の情報をデータセットに格納するため、basADOUtility の SetJVDataRaceDataSet を呼び出します。引数には、格納する JV-Data 構造体と、格納先のデータセットを指定します。

表示は JV-Data 構造体を格納し終えてから行うため、データ表示処理はコメントアウトします。(削除してしまってもかまいません)

・ Point3

馬毎レース情報の読み込みのため、レコード種別 ID が「SE」(馬毎レース情報)のレコードの処理を記述します。読み込んだ情報は、レース詳細と同様に SetJVDataUmaRaceDataSet を呼び出しデータセットに格納します。

また、Else の処理のコメントが、「レース詳細以外は読み飛ばす」になっていますので、こちらも修正しておきます。

- ⑧ データセットの内容を表示する処理を加えます。挿入位置にご注意ください。

(灰色部分は前回コーディング済み、薄い水色部分はコーディングの修正箇所)

(前省略)

```
' タイマー有効時は、無効化する
If tmrDownload.Enabled = True Then
    tmrDownload.Enabled = False
    prgDownload.Value = prgDownload.Maximum
End If
```

```
End If
```

```
' データセットに格納した各情報を画面に表示
```

```
' レース詳細の情報を一行表示
```

```
Dim raceIndex As Integer
Dim raceDRs As JVDDataDataSet.RACERow() '' レース詳細(複数行)格納用
```

```
' レース詳細の件数を取得
```

```
raceIndex = jvdds.RACE.Count()
```

Point4

```
' レース詳細が存在しない場合
```

```
If raceIndex = 0 Then
    MsgBox("レース情報が存在しません")
```

```
Else
```

```
' レース詳細の第10レースを検索(複数行)
```

```
raceDRs = jvdds.RACE.Select("RaceNum = '10' ")
```

Point5

```
If raceDRs.Length = 0 Then
    MsgBox("検索結果が0件です")
```

```
Exit Try
```

```
End If
```

```
' 取得結果の一件目を構造体にセット
```

```
SetJVDataRaceStructure(raceDRs(0), RaceInfo)
```

```
' 画面表示
```

```
rtbData.AppendText( _
    "年:" & RaceInfo.id.Year & _
    " 月日:" & RaceInfo.id.MonthDay & _
    " 場:" & objCodeConv.GetCodeName("2001", RaceInfo.id.JyoCD, "3") & _
    " 回次:" & RaceInfo.id.Kaiji & _
    " 日次:" & RaceInfo.id.Nichiji & _
    " R:" & RaceInfo.id.RaceNum & _
    " レース名:" & RaceInfo.RaceInfo.Ryakusyo10 _
    & vbCrLf)
```

Point6

```
' 表示したレース詳細の馬毎レース情報を全馬分表示
```

```
Dim raceUmaDRs As JVDDataDataSet.UMA_RACERow() '' 馬毎レース情報(複数行)格納用
```

```
Dim raceUmaDR As JVDDataDataSet.UMA_RACERow '' 馬毎レース情報(一行)格納用
```

```
' データセットからレース詳細のキーで馬毎レース情報の行(複数行)を検索
```

```
raceUmaDRs = jvdds.UMA_RACE.Select( _
    "Year = '" & RaceInfo.id.Year & "' AND " & _
    "MonthDay = '" & RaceInfo.id.MonthDay & "' AND " & _
    "JyoCD = '" & RaceInfo.id.JyoCD & "' AND " & _
    "Kaiji = '" & RaceInfo.id.Kaiji & "' AND " & _
    "Nichiji = '" & RaceInfo.id.Nichiji & "' AND " & _
    "RaceNum = '" & RaceInfo.id.RaceNum & "'")
```

```

' 一行ずつ取り出し、画面に表示
For Each raceUmaDR In raceUmaDRs
' 行の情報を構造体にセット
SetJVDataUmaRaceStructure(raceUmaDR, RaceUmaInfo)
' 画面表示
rtbData.AppendText( _
    " 枠:" & RaceUmaInfo.Wakuban & _
    " 馬番:" & RaceUmaInfo.Umaban & _
    " 馬名:" & RaceUmaInfo.Bamei & _
    " 騎手:" & RaceUmaInfo.KisyuRyakusyo _
    & vbCrLf)
Next
End If
Catch
' Debug.WriteLine(Err.Description)
MsgBox(Err.Description)
Exit Sub
Finally
' JVLink終了処理
IReturnCode = Me.AxJVLink1.JVClose()
If IReturnCode <> 0 Then
    MsgBox("JVCloseエラー:" & IReturnCode)
End If
' 後処理
jvdds.Dispose() ' JV-Dataデータセットのリソース解放
End Try
(後省略)

```

Point7

Point8

Point9

[ソースコード 008-03]

• Point4

データセットからレース詳細を取得するためのデータロックラスを定義します。

JVDataDataSet に、RACE 用のデータロックラス JVDataDataSet.RACERow が用意されていますので、今回はこちらを使用します。複数のレース詳細を取得するため配列で定義します。

• Point5

データセットからレース詳細の情報を取得します。データセットの RACE テーブル用 Select メソッドでフィルタ基準を指定し、複数のレース詳細を取得します。

さらに取得した複数のレース詳細の一件目を JV-Data 構造体に格納するため、basADOUtility の SetJVDataRaceStructure を呼び出します。引数には、格納するデータロックラス（一件）と、格納先の JV-Data 構造体を指定します。

JV-Data 構造体に格納した情報を使って内容を画面表示します。

• Point6

続いて、取得したレース詳細の情報に関連する馬毎レース情報を表示します。

まず、複数の馬毎レース情報を格納するためのデータロックラスの配列と、馬毎レース情報を一件ずつ処理するためのデータロックラスを定義します。いずれ

も、JVDataDataSet.UMA\_RACERow クラスです。

レース詳細と同様に馬毎レース情報をデータセットから取り出します。データセットの UMA\_RACE テーブル用 Select メソッドに指定するフィルタ基準は、レース詳細の主キーです。

・ Point7

For Each 文で馬毎レース情報を一件ずつ処理します。

SetJVDataUmaRaceStructure により、馬毎レース情報を JV-Data 構造体にセットし、一行ずつ画面に表示します。

[ ワンポイントメモ ]

出走馬名表状態（木曜日等）では、枠番、馬番は初期値です。このように、タイミングによって初期値となる項目は、実際の競馬ソフトを開発する際に考慮する必要があります。初期値については、JV-Data 仕様書をご覧ください。

・ Point8

今回の処理に合わせていくつか処理を修正します。まず、エラー発生時はメッセージボックスにてエラーメッセージを表示するようにします。続いてエラー発生時の処理を適切に行うため、Try~Catch に Finally を追加し、JVLink の終了処理を Finally 内に移します。

・ Point9

使用したクラスのリソースを解放します。

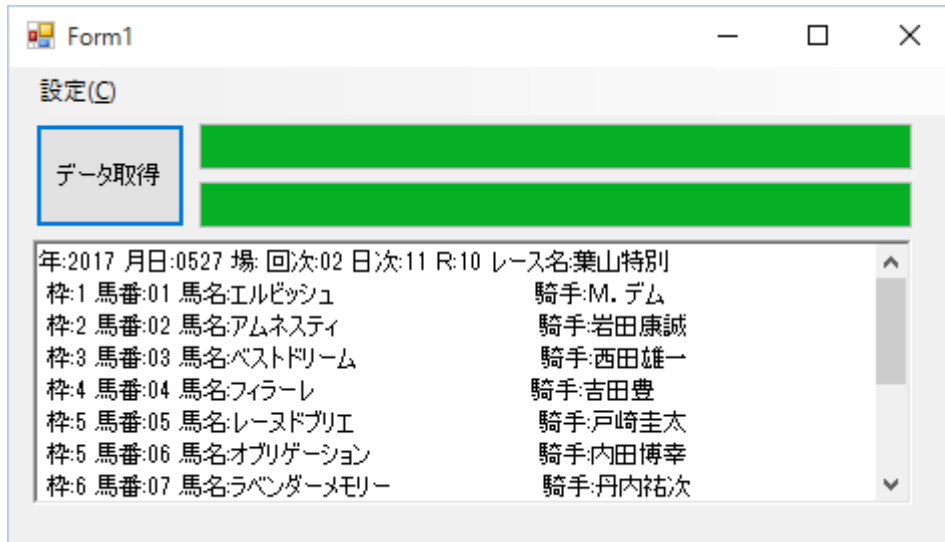
【 確認しよう 】

それでは、実際に動かしてみましょう。

- 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。
- ② JV-Link を使用するための「利用キー」を設定していない場合は、メインフォーム上のメニューから「設定」→「JV-Link の設定」を選択し、「利用キー」を設定します（詳細は Lesson-1 を参照）。
- ③ 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、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ます。これらの数値は、実行するタイミングによって異なります。詳しくは、Lesson-2 をご覧下さい。
- ④ メッセージボックスを「OK」で閉じると、JV-Data のレース詳細、馬毎レース情報の読み込みとデータセットへの格納が行われます。すべての情報が格納された後、レース詳細情報の「開催年、開催月日、競馬場名、開催回、開催日、レース番号、競走

名略称 10 文字」が 1 行表示され、続いて馬毎レース情報の「枠番、馬番、馬名、騎手名」が表示されます。

JV-Data のダウンロード進捗状況や読み込み進捗状況は上下のプログレスバーで確認することができます。



#### [ ワンポイントメモ ]

レース詳細の `Select` メソッドのフィルタ基準を変更し、他のレースの情報も表示してみましよう。(フィルタ基準の書式は「フィールド名 比較演算子 値」です。詳細についてはマイクロソフトの MSDN ライブラリなどをご参照ください。)

例)

```
raceDRs = jvdds.RACE.Select("RaceNum = '11' AND YoubiCD = '1'")
```

```
raceDRs = jvdds.RACE.Select("GradeCD In ('A','B','C','D','E')")
```