（別添資料）条文調査フォームのソースコード

```vb
Imports System.Net.Http
Imports System.Xml

Class MainWindow
    Dim CurrentLawName As String
    Dim CurrentLawNumber As String
    Dim CurrentArticleNumber As String
    Dim TotalArticlesNumber As Long
    Dim ProcessUnits As Long = 1
    Dim ParallelLevel As Long = 1
    Const MaximumParallelLevel As Long = 1500
    Public LastLawNum(MaximumParallelLevel) As String
    Public LastArticleNum(MaximumParallelLevel) As String
    Public LastArticleXML(MaximumParallelLevel) As String
    Public 調査結果(MaximumParallelLevel) As String

    Private Sub LawNumberReady(sender As Object, e As RoutedEventArgs) Handles 法令番号設定完了ボタン.Click
        CurrentLawName = Me.法令名.Text
        CurrentLawNumber = GetLawNumberAsync(Me.法令名.Text).Result
        Me.法令番号.Text = CurrentLawNumber
        ProcessUnits = Long.Parse(Me.処理単位.Text)
        TotalArticlesNumber = GetNumberOfArticles()
        ParallelLevel = (TotalArticlesNumber - 1) ¥ ProcessUnits + 1
        If ParallelLevel > MaximumParallelLevel Then
            ProcessUnits = TotalArticlesNumber ¥ MaximumParallelLevel + 1
            ParallelLevel = (TotalArticlesNumber - 1) ¥ ProcessUnits + 1
        End If
        Me.条文数.Text = CStr(TotalArticlesNumber)
        Beep()
    End Sub

    Private Sub StartSequential(sender As Object, e As RoutedEventArgs) Handles 開始ボタン1.Click
        Call GetAllArticleNumbers1()
        Beep()
    End Sub

    Private Sub StartParallel(sender As Object, e As RoutedEventArgs) Handles 開始ボタン2.Click
        If (Me.法令名.Text = CurrentLawName) And (Long.Parse(Me.処理単位.Text) <> ProcessUnits) Then
            ProcessUnits = Long.Parse(Me.処理単位.Text)
            ParallelLevel = (TotalArticlesNumber - 1) ¥ ProcessUnits + 1
        End If
        Call GetAllArticleNumbers2()
        Beep()
    End Sub

    Private Sub StartParallelAuto(sender As Object, e As RoutedEventArgs) Handles 開始ボタン3.Click
        Dim CurrentProcessUnits As Long = TotalArticlesNumber
        Do
            CurrentProcessUnits = (CurrentProcessUnits + 1) ¥ 2
            Debug.Print(CurrentProcessUnits)
            ProcessUnits = CurrentProcessUnits
            ParallelLevel = (TotalArticlesNumber - 1) ¥ ProcessUnits + 1
            Me.処理単位.Text = CStr(CurrentProcessUnits)
            Call GetAllArticleNumbers2()
        Loop Until CurrentProcessUnits <= 1
```

```
56          Beep()
57      End Sub
58
59      Public Async Function GetLawNumberAsync(ByVal LawName As String) As Task(Of
          String)
60          Using LawList As New HttpClient()
61              Dim responseText As String
62              Dim URL As String = "https://elaws.e-gov.go.jp/api/1/lawlists/1"
63              Try
64                  responseText = Await LawList.GetStringAsync(URL).ConfigureAwait
                    (False)
65              Catch ex As Exception
66                  Return "サーバー交信誤り"
67              End Try
68              Dim xdoc As XmlDocument = New XmlDocument()
69              xdoc.LoadXml(responseText)
70              Dim xmlResultCode As XmlNode = xdoc.SelectSingleNode("//DataRoot/
                  Result/Code")
71              If xmlResultCode.InnerText <> "0" Then
72                  Return "サーバー処理誤り"
73              End If
74              Dim ThisXpath As String = "//LawName[text()='" & LawName & "']"
75              Dim xmlLawNameListInfoEntry As XmlNode = xdoc.SelectSingleNode
                  (ThisXpath)
76              Return xmlLawNameListInfoEntry.ParentNode.SelectSingleNode
                  ("LawNo").InnerText
77          End Using
78      End Function
79
80      Public Function GetNumberOfArticles() As Long
81          Dim L, C, R, W As Long
82          L = 1 : R = 2
83          While Process1(CurrentLawNumber, R) = "OK"
84              W = R
85              R += (R - L) * 2
86              L = W
87          End While
88          Do
89              C = (L + R) ¥ 2
90              Select Case Process1(CurrentLawNumber, C)
91                  Case "OK"
92                      L = C
93                      If R - C <= 1 Then Return C
94                  Case "DELETED"
95                      R = C
96                      If C - L <= 1 Then Return L
97              End Select
98          Loop
99      End Function
100
101     Sub GetAllArticleNumbers1()
102         Dim Result As String
103         Dim I As Long = 0
104         Dim StartDT As DateTime = DateTime.Now
105         Dim ElapsedTime As Long
106         CurrentArticleNumber = "第一条"
107         Me.調査結果1.AppendText("<<<<<<<<<<<<>>>>>>>>>>>>" & vbCrLf &
108                                 "<<" & Me.法令名.Text & ">>" & vbCrLf &
109                                 "<<基本条文数計" & Me.条文数.Text & "条>>" &
                        vbCrLf)
```

```
110         Do
111             I += 1
112             Me.調査結果1.AppendText("[" & CStr(I) & "] " & CurrentArticleNumber &
                  " " & Int((DateTime.Now - StartDT).TotalSeconds) & vbCrLf)
113             Result = Process2(CurrentLawNumber, CurrentArticleNumber, 1)
114             If Result = "DELETED" Then Exit Do
115             CurrentArticleNumber = GetLatter(CurrentArticleNumber, 1)
116         Loop
117         ElapsedTime = Int((DateTime.Now - StartDT).TotalSeconds)
118         Me.調査結果1.AppendText("<<処理時間" & ElapsedTime & "秒>>" & vbCrLf &
119                                 "<<基本条文１件当たり" & Math.Round(ElapsedTime /
                        TotalArticlesNumber, 2) & "秒／件>>" & vbCrLf &
120                                 "<<確認済条文数計" & I - 1 & "条>>" & vbCrLf)
121     End Sub
122
123     Sub GetAllArticleNumbers2()
124         Dim StartDT As DateTime = DateTime.Now
125         Dim ElapsedTime As Long
126         For I As Long = 0 To ParallelLevel
127             LastLawNum(I) = ""
128             LastArticleNum(I) = ""
129             LastArticleXML(I) = ""
130             調査結果(I) = ""
131         Next I
132         調査結果(0) = "<<<<<<<<<<<<>>>>>>>>>>>>" & vbCrLf &
133                       "<<" & Me.法令名.Text & ">>" & vbCrLf &
134                       "<<基本条文数計" & Me.条文数.Text & "条>>" & vbCrLf &
135                       "<<処理単位" & ProcessUnits & "条>>" & vbCrLf &
136                       "<<処理プロセス数" & ParallelLevel & "重>>" & vbCrLf
137         Parallel.For(1, ParallelLevel + 1,
138                 Sub(I As Long)
139                     Dim Result As String = ""
140                     Dim J As Long = (I - 1) * ProcessUnits + 1
141                     Dim CurrentArticleNumber As String = NormalizeANum(J)
142                     Dim LimitArticleNumber As String = NormalizeANum(J +
                         ProcessUnits)
143                     Dim K As Long = 0
144                     While CurrentArticleNumber <> LimitArticleNumber
145                         K += 1
146                         調査結果(I) += "[" & CStr(I) & ":" & CStr(K) & "] " &
                        CurrentArticleNumber & " " & Int((DateTime.Now -
                        StartDT).TotalSeconds) & vbCrLf
147                         Result = Process2(CurrentLawNumber, CurrentArticleNumber,
                        I)
148                         If Result = "DELETED" Then Exit While
149                         CurrentArticleNumber = GetLatter(CurrentArticleNumber, I)
150                     End While
151                 End Sub)
152         For I = 0 To ParallelLevel : Me.調査結果2.AppendText(調査結果(I)) : Next I
153         ElapsedTime = Int((DateTime.Now - StartDT).TotalSeconds)
154         Me.調査結果2.AppendText("<<処理時間" & ElapsedTime & "秒>>" & vbCrLf &
155                                 "<<基本条文１件当たり" & Math.Round(ElapsedTime /
                        TotalArticlesNumber, 2) & "秒／件>>" & vbCrLf)
156     End Sub
157
158     Private Function GetLatter(ByVal CAN As String,
159                                ByVal CP As Long) As String
160         Dim L = 1, LevelStack(5) As Integer
161         LevelStack(L) = ConvToNumber(Mid(CAN, 2, (InStr(CAN, "条") - 2)))
162         CAN = Mid(CAN, (InStr(CAN, "条") + 2))
```

```
163         While CAN <> ""
164             L += 1
165             If InStr(CAN, "の") > 0 Then
166                 LevelStack(L) = ConvToNumber(Mid(CAN, 1, (InStr(CAN, "の") - 1)))
167                 CAN = Mid(CAN, (InStr(CAN, "の") + 1))
168             Else
169                 LevelStack(L) = ConvToNumber(CAN)
170                 CAN = ""
171             End If
172         End While
173         Do
174             If ServerAccessAsync2(CurrentLawNumber, KanjiArticleNumber(LevelStack,
                    L), CP).Result = "" Then
175                 LevelStack(L) += 1
176                 Exit Do
177             End If
178             L += 1
179             LevelStack(L) = 2
180             While L >= 1
181                 If ServerAccessAsync2(CurrentLawNumber, KanjiArticleNumber
                        (LevelStack, L), CP).Result <> "" Then
182                     Exit Do
183                 End If
184                 If L = 1 Then Exit Do
185                 LevelStack(L) = 0
186                 L -= 1
187                 LevelStack(L) += 1
188             End While
189         Loop
190         GetLatter = KanjiArticleNumber(LevelStack, L)
191     End Function
192 
193     Function Process1(ByVal Lnum As String,
194                       ByVal Anum As Long) As String
195         Dim ReturnValue As String
196         ReturnValue = ServerAccessAsync1(Lnum, NormalizeANum(CStr(Anum))).Result
197         If Len(ReturnValue) = 0 Then
198             Return "DELETED"
199         End If
200         Return "OK"
201     End Function
202 
203     Public Async Function ServerAccessAsync1(ByVal LawNum As String,
204                                              ByVal ArticleNum As String) As Task
                        (Of String)
205         Using client = New HttpClient()
206             Dim responseText As String
207             Dim mytext As String
208             Dim URL As String = "https://elaws.e-gov.go.jp/api/1/articles;lawNum="
209             URL += encodeUTF8(LawNum)
210             URL += ";article="
211             URL += encodeUTF8(ArticleNum)
212             Try
213                 responseText = Await client.GetStringAsync(URL).ConfigureAwait
                        (False)
214             Catch ex As Exception
215                 Return ""
216             End Try
217             Dim xdoc As XmlDocument = New XmlDocument()
218             xdoc.LoadXml(responseText)
```

```
219             Dim xmlResultCode As XmlNode = xdoc.SelectSingleNode("//DataRoot/
                  Result/Code")
220             If xmlResultCode.InnerText <> "0" Then
221                 Return ""
222             End If
223             Dim xmlLawContents As XmlNode = xdoc.SelectSingleNode("//DataRoot/
                  ApplData/LawContents/Article")
224             mytext = xmlLawContents.InnerXml
225             Return mytext
226         End Using
227     End Function
228
229     Function Process2(ByVal Lnum As String,
230                       ByVal Anum As String,
231                       ByVal CP As Long) As String
232         Dim ReturnValue As String
233         ReturnValue = ServerAccessAsync1(Lnum, NormalizeANum(Anum)).Result
234         If Len(ReturnValue) = 0 Then
235             Return "DELETED"
236             Exit Function
237         End If
238         Return "OK"
239     End Function
240
241     Public Async Function ServerAccessAsync2(ByVal LawNum As String,
242                                              ByVal ArticleNum As String,
243                                              ByVal CP As Long) As Task(Of String)
244         Using client = New HttpClient()
245             Dim responseText As String
246             Dim mytext As String
247             If (LawNum = LastLawNum(CP)) And (ArticleNum = LastArticleNum(CP))
                  Then
248                 Return LastArticleXML(CP)
249             Else
250                 Dim URL As String = "https://elaws.e-gov.go.jp/api/1/
                      articles;lawNum="
251                 URL += encodeUTF8(LawNum)
252                 URL += ";article="
253                 URL += encodeUTF8(ArticleNum)
254                 Try
255                     responseText = Await client.GetStringAsync(URL).ConfigureAwait
                          (False)
256                 Catch ex As Exception
257                     Return ""
258                 End Try
259                 Dim xdoc As XmlDocument = New XmlDocument()
260                 xdoc.LoadXml(responseText)
261                 Dim xmlResultCode As XmlNode = xdoc.SelectSingleNode("//DataRoot/
                      Result/Code")
262                 If xmlResultCode.InnerText <> "0" Then
263                     Return ""
264                 End If
265                 Dim xmlLawContents As XmlNode = xdoc.SelectSingleNode("//DataRoot/
                      ApplData/LawContents/Article")
266                 mytext = xmlLawContents.InnerXml
267                 LastLawNum(CP) = LawNum
268                 LastArticleNum(CP) = ArticleNum
269                 LastArticleXML(CP) = mytext
270                 ' Debug.Write(Str(CP) & ":" & ArticleNum & "/")
271                 Return mytext
```

```
272             End If
273         End Using
274     End Function
275
276     Function NormalizeANum(ByVal T As String) As String
277         Dim S As String
278         Dim V As Integer
279         Dim FirstDigit As Boolean
280         NormalizeANum = ""
281         FirstDigit = True
282         S = StrConv(T, vbWide)
283         Do While Len(S) > 0
284             Select Case LeftMost(S)
285                 Case "０" To "９"
286                     V = Val(StrConv(LeftMost(S), vbNarrow))
287                     S = Mid(S, 2)
288                     Do While Len(S) > 0
289                         Select Case LeftMost(S)
290                             Case "０" To "９"
291                                 V = V * 10 + Val(StrConv(LeftMost(S), vbNarrow))
292                                 S = Mid(S, 2)
293                             Case Else
294                                 Exit Do
295                         End Select
296                     Loop
297                     If FirstDigit Then
298                         NormalizeANum = NormalizeANum & "第"
299                     End If
300                     NormalizeANum =
301                     NormalizeANum & ConvToKansuuji(V)
302                     If FirstDigit Then
303                         NormalizeANum = NormalizeANum & "条"
304                         FirstDigit = False
305                     End If
306                 Case "－", "―"
307                     NormalizeANum = NormalizeANum & "の"
308                     S = Mid(S, 2)
309                 Case "　"
310                     S = Mid(S, 2)
311                 Case Else
312                     NormalizeANum = NormalizeANum & LeftMost(S)
313                     S = Mid(S, 2)
314             End Select
315         Loop
316     End Function
317
318     Function LeftMost(ByVal str As String) As String
319         LeftMost = Mid(str, 1, 1)
320     End Function
321
322     Function ConvToKansuuji(ByVal Num As Long) As String
323         ConvToKansuuji = ""
324         If Num >= 1000 Then
325             Select Case Int(Num / 1000)
326                 Case 1 : ConvToKansuuji += "千"
327                 Case 2 : ConvToKansuuji += "二千"
328                 Case 3 : ConvToKansuuji += "三千"
329                 Case 4 : ConvToKansuuji += "四千"
330                 Case 5 : ConvToKansuuji += "五千"
331                 Case 6 : ConvToKansuuji += "六千"
```

```
332                 Case 7 : ConvToKansuuji += "七千"
333                 Case 8 : ConvToKansuuji += "八千"
334                 Case 9 : ConvToKansuuji += "九千"
335             End Select
336             Num -= Int(Num / 1000) * 1000
337         End If
338         If Num >= 100 Then
339             Select Case Int(Num / 100)
340                 Case 0
341                 Case 1 : ConvToKansuuji += "百"
342                 Case 2 : ConvToKansuuji += "二百"
343                 Case 3 : ConvToKansuuji += "三百"
344                 Case 4 : ConvToKansuuji += "四百"
345                 Case 5 : ConvToKansuuji += "五百"
346                 Case 6 : ConvToKansuuji += "六百"
347                 Case 7 : ConvToKansuuji += "七百"
348                 Case 8 : ConvToKansuuji += "八百"
349                 Case 9 : ConvToKansuuji += "九百"
350             End Select
351             Num -= Int(Num / 100) * 100
352         End If
353         If Num >= 10 Then
354             Select Case Int(Num / 10)
355                 Case 0
356                 Case 1 : ConvToKansuuji += "十"
357                 Case 2 : ConvToKansuuji += "二十"
358                 Case 3 : ConvToKansuuji += "三十"
359                 Case 4 : ConvToKansuuji += "四十"
360                 Case 5 : ConvToKansuuji += "五十"
361                 Case 6 : ConvToKansuuji += "六十"
362                 Case 7 : ConvToKansuuji += "七十"
363                 Case 8 : ConvToKansuuji += "八十"
364                 Case 9 : ConvToKansuuji += "九十"
365             End Select
366             Num -= Int(Num / 10) * 10
367         End If
368         Select Case Num
369             Case 0
370             Case 1 : ConvToKansuuji += "一"
371             Case 2 : ConvToKansuuji += "二"
372             Case 3 : ConvToKansuuji += "三"
373             Case 4 : ConvToKansuuji += "四"
374             Case 5 : ConvToKansuuji += "五"
375             Case 6 : ConvToKansuuji += "六"
376             Case 7 : ConvToKansuuji += "七"
377             Case 8 : ConvToKansuuji += "八"
378             Case 9 : ConvToKansuuji += "九"
379         End Select
380     End Function
381
382     Private Function ConvToNumber(ByVal KNum As String) As Integer
383         Dim Factor As Integer = 0
384         ConvToNumber = 0
385         Select Case Mid(KNum, 1, 1)
386             Case "一" : Factor = 1 : KNum = Mid(KNum, 2)
387             Case "二" : Factor = 2 : KNum = Mid(KNum, 2)
388             Case "三" : Factor = 3 : KNum = Mid(KNum, 2)
389             Case "四" : Factor = 4 : KNum = Mid(KNum, 2)
390             Case "五" : Factor = 5 : KNum = Mid(KNum, 2)
391             Case "六" : Factor = 6 : KNum = Mid(KNum, 2)
```

```
392             Case "七" : Factor = 7 : KNum = Mid(KNum, 2)
393             Case "八" : Factor = 8 : KNum = Mid(KNum, 2)
394             Case "九" : Factor = 9 : KNum = Mid(KNum, 2)
395             Case "千", "百", "十" : Factor = 1
396         End Select
397         If Mid(KNum, 1, 1) = "千" Then
398             ConvToNumber += Factor * 1000 : Factor = 0
399             KNum = Mid(KNum, 2)
400             Select Case Mid(KNum, 1, 1)
401                 Case "一" : Factor = 1 : KNum = Mid(KNum, 2)
402                 Case "二" : Factor = 2 : KNum = Mid(KNum, 2)
403                 Case "三" : Factor = 3 : KNum = Mid(KNum, 2)
404                 Case "四" : Factor = 4 : KNum = Mid(KNum, 2)
405                 Case "五" : Factor = 5 : KNum = Mid(KNum, 2)
406                 Case "六" : Factor = 6 : KNum = Mid(KNum, 2)
407                 Case "七" : Factor = 7 : KNum = Mid(KNum, 2)
408                 Case "八" : Factor = 8 : KNum = Mid(KNum, 2)
409                 Case "九" : Factor = 9 : KNum = Mid(KNum, 2)
410                 Case "百", "十" : Factor = 1
411             End Select
412         End If
413         If Mid(KNum, 1, 1) = "百" Then
414             ConvToNumber += Factor * 100 : Factor = 0
415             KNum = Mid(KNum, 2)
416             Select Case Mid(KNum, 1, 1)
417                 Case "一" : Factor = 1 : KNum = Mid(KNum, 2)
418                 Case "二" : Factor = 2 : KNum = Mid(KNum, 2)
419                 Case "三" : Factor = 3 : KNum = Mid(KNum, 2)
420                 Case "四" : Factor = 4 : KNum = Mid(KNum, 2)
421                 Case "五" : Factor = 5 : KNum = Mid(KNum, 2)
422                 Case "六" : Factor = 6 : KNum = Mid(KNum, 2)
423                 Case "七" : Factor = 7 : KNum = Mid(KNum, 2)
424                 Case "八" : Factor = 8 : KNum = Mid(KNum, 2)
425                 Case "九" : Factor = 9 : KNum = Mid(KNum, 2)
426                 Case "十" : Factor = 1
427             End Select
428         End If
429         If Mid(KNum, 1, 1) = "十" Then
430             ConvToNumber += Factor * 10 : Factor = 0
431             KNum = Mid(KNum, 2)
432             If KNum <> "" Then
433                 Select Case Mid(KNum, 1, 1)
434                     Case "一" : Factor = 1 : KNum = Mid(KNum, 2)
435                     Case "二" : Factor = 2 : KNum = Mid(KNum, 2)
436                     Case "三" : Factor = 3 : KNum = Mid(KNum, 2)
437                     Case "四" : Factor = 4 : KNum = Mid(KNum, 2)
438                     Case "五" : Factor = 5 : KNum = Mid(KNum, 2)
439                     Case "六" : Factor = 6 : KNum = Mid(KNum, 2)
440                     Case "七" : Factor = 7 : KNum = Mid(KNum, 2)
441                     Case "八" : Factor = 8 : KNum = Mid(KNum, 2)
442                     Case "九" : Factor = 9 : KNum = Mid(KNum, 2)
443                     Case Else : Factor = 0
444                 End Select
445                 ConvToNumber += Factor
446             End If
447         Else
448             ConvToNumber += Factor
449         End If
450     End Function
451
```

```
452     Private Function KanjiArticleNumber(LevelStack, ByVal L) As String
453         Dim I As Integer
454         KanjiArticleNumber = "第" & ConvToKansuuji(LevelStack(1)) & "条"
455         For I = 2 To L
456             KanjiArticleNumber &= "の" & ConvToKansuuji(LevelStack(I))
457         Next
458     End Function
459 
460     Public Function encodeUTF8(ByVal mytext As String) As String
461         Dim mystream As New ADODB.Stream
462         Dim mybinary, mynumber
463         With mystream
464             .Open()
465             .Type = ADODB.StreamTypeEnum.adTypeText
466             .Charset = "UTF-8"
467             .WriteText(mytext)
468             .Position = 0
469             .Type = ADODB.StreamTypeEnum.adTypeBinary
470             .Position = 3
471             mybinary = .Read
472             .Close()
473         End With
474         encodeUTF8 = ""
475         For Each mynumber In mybinary
476             encodeUTF8 = encodeUTF8 & "%" & Hex(mynumber)
477         Next
478     End Function
479 
480 End Class
481 
```