

Accessによる自社専用カレンダーフォームの作成 No.98019

キーワード： Access、プログラム、カレンダー、フォーム、自社専用カレンダー

概要

Access（マイクロソフト社製）で作成するシステムにおいて日付入力時の補助としてカレンダーを表示すると便利である。ここでは、休日が一般のカレンダーとは異なる会社で自社独自の休日を色を変えて表示できるカレンダーフォームの作成方法について紹介する。

データベーステーブルの作成

自社独自の休日を登録しておくデータベーステーブル(名前「calen」)を作成する。

	項目	フィールド名	型	備考
1	日付	Ddate	日付型	主キー
2	休日	Holiday	Yes/No型	Yes=休日

カレンダーフォーム作成

カレンダーを表示するフォーム(名前: カレンダ)の外観を次のように作成する。

フォームの新規作成を選択し、デザインビュー(ウィザードを使用せずに作成)を選択し、フォームを表示する。

図1に示すように日付を表示するラベルを37個、フォームに配置する。この時、配置する順番に気をつける。左上(図1 L0)から

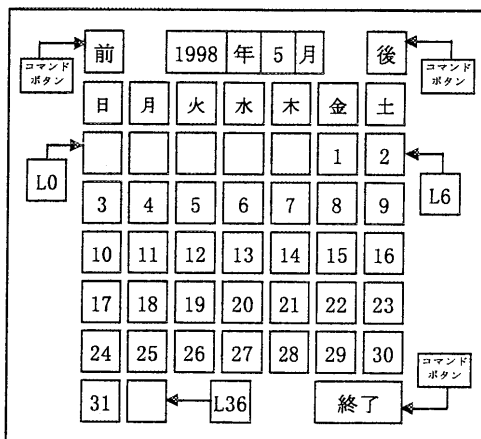


図1 カレンダーフォームレイアウト

順番に1行に7つ配置し、それを4回繰り返し5行とし、更に2つ最下行に配置する。(これでラベルのコントロール番号を0から36にでき、プログラムが簡単になる)ラベル名をL0 ~ L36とする。

他にテキストボックス2つ(年、月表示用)、コマンドボタン3つ(前月表示、次月表示、終了)を配置する。

プログラムの概略

上記のように、最初にラベルを37個順番に配置することで、プログラム中で各ラベルを指定する時、次のように書くことができる。

```
Dim F As Form
```

```
Set F = Forms!カレンダー
```

とすれば、変数Fをカレンダーフォームに割り当て、L0はF(0)、... L36はF(36)で指定できる。例えば、L6に数字5を表示したければ、F(6).Caption = 5とプログラムすればよい。

カレンダーは表示する月の1日の曜日と最後の日が何日(31日、30日、29日、28日)かが分かれば、作成することができる。

月の1日の曜日を知るには、Accessの組み込み関数として用意されている、DateSerial()とWeekDay()の2つの関数を使用する。Accessには、日付型という変数型があり、例えば、DateSerial(98,5,1)とすれば、1998年5月1日の日付型数値が求まる。WeekDay()は、引数に日付型数値を入れることにより、日曜日は1、月曜日は2...土曜日は7という数値を関数の値として返す。例えばWeekDay(DateSerial(98,5,1))は、6(金曜日)を返す。したがって、98年5月は、1日が金曜日(6)なので1を表示するラベルは、L5である。したがって計算式WeekDay(DateSerial(98,5,1)) - 1によ

り1を表示するラベルが計算できる(今の場合は5)。1日の位置が決まれば、後は順番に1ずつずらしていけばよい。

月の最後の日は次のようにして求める。DateSerial()関数では、たとえばDateSerial(98,5,32)は1998年6月1日という日付が帰ってくるので、組み込み関数Month()を使用して月の数値が表示月と同じかどうかを調べれば、月の最終日が分かる。

カレンダー作成関数

以上で、カレンダーを作成する関数を図2のように作成できる。ここでは変数Iを0から36まで変化させ、Label0 ~ Label36のそれぞれを表示しない(該当する日がない)のか、表示するならその数字を表示し、さらに休日かどうかを調べ、休日なら背景色を赤色で、その他の日は、白色で表示する。

コマンドボックスのプログラム(図3)

```
Function Calender(YY As Integer, MM As Integer)
    ' YY は表示年、MM は表示月
    Dim dd As Integer, I As Integer, StWeekDay As Integer
    Dim Today As Date
    ' カレンダーテーブルで、表示する月の1日を検索
    rstProduct.Seek "=", DateSerial(YY, MM, 1)
    StWeekDay = WeekDay(DateSerial(YY, MM, 1)) - 1: dd = 1
    For I = 0 To 36 ' I はラベルの番号
        Today = DateSerial(YY, MM, dd) ' 日付型に変換
        If I < StWeekDay Or MM <> Month(Today) Then
            F(I).Visible = False ' ラベル非表示
        Else
            F(I).Caption = dd ' 日付を表示
            F(I).Visible = True ' ラベル表示
            ' レコード内の休日(休日)かどうかを検査
            If rstProduct("holiday") = False Then
                ' Yes でない(= False) なら出勤日・白色表示
                F(I).BackColor = 16777215
            Else
                ' Yes なら休日・赤色表示
                F(I).BackColor = 255
            End If
            dd = dd + 1
            rstProduct.MoveNext ' 次のレコード(次日)に移動
        End If
    Next
End Function
```

図2 カレンダー作成関数のプログラムリスト

```
Dim DBcurrent As Database: Dim rstProduct As Recordset
Dim F As Form
Function Form_Open1(F1 As Form)
    Set DBcurrent = CurrentDb
    Set F = Forms!カレンダー
    Set rstProduct = DBcurrent.OpenRecordset("calen", dbOpenTable)
End Function
```

図4 フォームを開けたときの処理

3つのコマンドボタンは、それぞれマウスでクリックすると、前月のカレンダーの表示、次月のカレンダーの表示、カレンダーフォームの消去を行う。

前月カレンダーの表示は、現在の月(変数MM)が1月なら、年(変数YY)をマイナス1し、MMに12を代入する。1月以外なら、MM=MM-1を実行した後、カレンダー表示関数Calender(YY, MM)を呼び出す。同様に次月カレンダーの表示の場合は、MMが12の場合に、YY=YY+1,MM=1とし、その他の場合は、MM=MM+1を実行した後、カレンダー表示関数Calender(YY, MM)を呼び出す。

フォームを開けたときの処理(図4) フォームを開けたときに、休日を記憶したテーブルをレコードセット変数rstProductに割り当てておく必要がある。

```
Private Sub 前_Click()
    ' 前月のカレンダーを表示
    Dim Dummy As Integer
    Dim YY As Integer, MM As Integer
    YY = F.年.Caption: MM = F.月.Caption
    If MM = 1 Then
        YY = YY - 1: MM = 12
    Else
        MM = MM - 1
    End If
    F.年.Caption = YY: F.月.Caption = MM
    Dummy = SubCalender2(YY, MM)
End Sub
Private Sub 後_Click()
    ' 次月のカレンダーを表示
    Dim Dummy As Integer
    Dim YY As Integer, MM As Integer
    YY = F.年.Caption
    MM = F.月.Caption
    If MM = 12 Then
        YY = YY + 1: MM = 1
    Else
        MM = MM + 1
    End If
    F.年.Caption = YY
    F.月.Caption = MM
    Dummy = SubCalender2(YY, MM)
End Sub
Private Sub 終了_Click()
    ' フォームの消去
    Docmd.Close
End Sub
```

図3 コマンドボックスのプログラム