

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

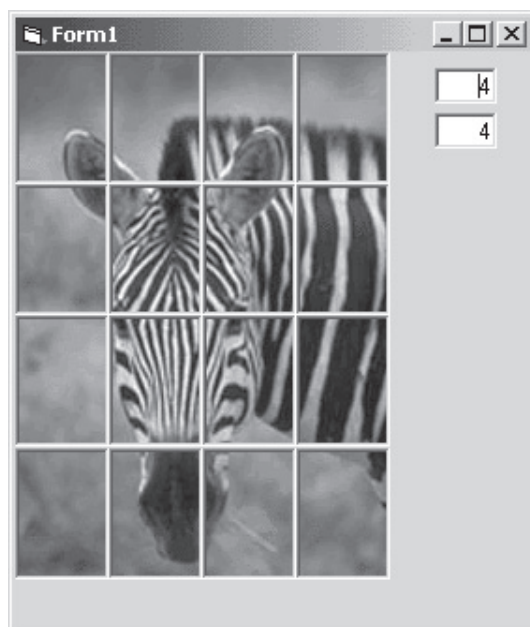
To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz



Skládačka



Obr. 48 **G15.VBP**

Co to dělá:

Načte vybraný obrázek (vybraný pomocí `CommonDialog1`), vytvoří $M \times N$ komponent `PictureBox` obsahujících odpovídající části načteného obrázku a tyto části dovolí přesouvat myší (`MouseDown`, `MouseMove`, `MouseUp`). Při konci přesouvání tyto komponenty doskakují na pozice odpovídající mřížce zadaných rozměrů.

Jak:

```
Dim VX As Integer
Dim VY As Integer
Dim SX As Integer
Dim SY As Integer

Private Sub Form_Click()
Dim JmenoSouboru As String
Dim PX As Integer
Dim PY As Integer
Dim X As Integer
Dim Y As Integer
Dim i As Integer
```

```

CommonDialog1.Filter = "Obrazky|*.bmp;*.gif;*"
    & ".jpg;*.dib;*.wmf;*.emf;*.ico|Všechno|*.*"
CommonDialog1.ShowOpen
JmenoSouboru = CommonDialog1.filename
If JmenoSouboru = "" Then
    Exit Sub
End If
imObrazek.Picture = LoadPicture(JmenoSouboru)

On Error GoTo Chyba
    PX = Val(Text1.Text)
    PY = Val(Text2.Text)
On Error GoTo 0

VX = Int(imObrazek.Width / PX)
VY = Int(imObrazek.Height / PY)
i = 1
For Y = 1 To PY
    For X = 1 To PX
        Load pbPuzzle(i)
        With pbPuzzle(i)
            .Left = (X - 1) * VX
            .Top = (Y - 1) * VY
            .Width = VX
            .Height = VY
            .PaintPicture imObrazek.Picture, 0, 0, _
                VX, VY, _
                (X - 1) * VX, _
                (Y - 1) * VY, VX, VY
            .Visible = True
        End With
        i = i + 1
    Next X
Next Y
imObrazek.Picture = LoadPicture
Exit Sub
Chyba:
    MsgBox "Špatně zadané rozměry"
End Sub

Private Sub pbPuzzle_MouseDown _
    (Index As Integer, Button As Integer, _
    Shift As Integer, X As Single, Y As Single)
If Button <> vbLeftButton Then Exit Sub
SX = X
SY = Y
End Sub

```

```

Private Sub pbPuzzle_MouseMove _
    Index As Integer, Button As Integer, _
        Shift As Integer, X As Single, Y As Single)
If Button <> vbLeftButton Then Exit Sub
With pbPuzzle(Index)
    .Left = .Left + X - SX
    .Top = .Top + Y - SY
End With
End Sub

Private Sub pbPuzzle_MouseUp _
    (Index As Integer, Button As Integer, _
        Shift As Integer, X As Single, Y As Single)
If Button <> vbLeftButton Then Exit Sub
With pbPuzzle(Index)
    .Left = VX * Int(.Left / VX)
    .Top = VY * Int(.Top / VY)
End With
End Sub

```

Vysvětlení

- Jako pomocný objekt pro natažení celého obrázku je z úsporných důvodů použita komponenta **Image**, která má nastavenou vlastnost **Visible** na **FALSE**.
- Přiřazení výsledku volání funkce **LoadPicture** bez parametru do vlastnosti **Picture** způsobí uvolnění obrázku z objektu.

Poznámka

*Nezapomeňte komponentě **PictureBox** nastavit vlastnost **AutoRedraw** na **TRUE**.*

Skládačka 2



Obr. 49 G16.VBP

Co to dělá:

Jako minulý příklad, několik vylepšení.

Jak:

1. Kontroluje se velikost okna formuláře a případně se okno zvětší.

```
...  
If ScaleWidth < VX * (PX + 1) + Text1.Width + 10 Then  
    ScaleWidth = VX * (PX + 1) + Text1.Width + 10  
    Text1.Left = ScaleWidth - Text1.Width - 10  
    Text2.Left = Text1.Left  
End If  
If ScaleHeight < VY * (PY + 1) + 10 Then  
    ScaleHeight = VY * (PY + 1) + 10  
End If  
...
```

2. Kostičky jsou při vytvoření náhodně rozházeny.

```
...  
.Left = Int(Rnd * (imObrazek.Width - VX))  
.Top = Int(Rnd * (imObrazek.Height - VY))  
...
```

3. Kontroluje, aby se kostičky nezarovnávaly za okraj formuláře a nepřekrývaly se.

```
Private Sub pbPuzzle_MouseUp(Index As Integer, _  
    Button As Integer, Shift As Integer, _  
    X As Single, Y As Single)  
  
    ...  
    Dim BX As Integer  
    Dim BY As Integer  
  
    If Button <> vbLeftButton Then Exit Sub  
    BX = VX * Int(pbPuzzle(Index).Left / VX)  
    BY = VY * Int(pbPuzzle(Index).Top / VY)  
  
    If BX < 0 Then BX = 0  
    If BY < 0 Then BY = 0  
  
    k = 1  
    For j = 1 To PY  
        For i = 1 To PX  
            With pbPuzzle(k)  
                If Abs(.Left - BX) < 1 And _  
                    Abs(.Top - BY) < 1 Then  
                    Beep  
                    Beep  
                    Exit Sub  
                End If  
            End With  
            k = k + 1  
        Next i  
    Next j  
    pbPuzzle(Index).Left = BX  
    pbPuzzle(Index).Top = BY  
    ...
```

4. Kontroluje se, zda je již obrázek složen.

```
'General Declaration  
Dim PX As Integer  
Dim PY As Integer  
  
...  
Private Sub pbPuzzle_MouseUp(Index As Integer, _  
    Button As Integer, Shift As Integer, _  
    X As Single, Y As Single)  
  
    ...  
    k = 1  
    For j = 1 To PY  
        For i = 1 To PX
```

```

    If Abs(pbPuzzle(k).Left - (i - 1) * VX) > 1 Then _
        Exit Sub
    If Abs(pbPuzzle(k).Top - (j - 1) * VY) > 1 Then _
        Exit Sub
    k = k + 1
Next i
Next j
MsgBox "Hurááááá!!!!"
End Sub

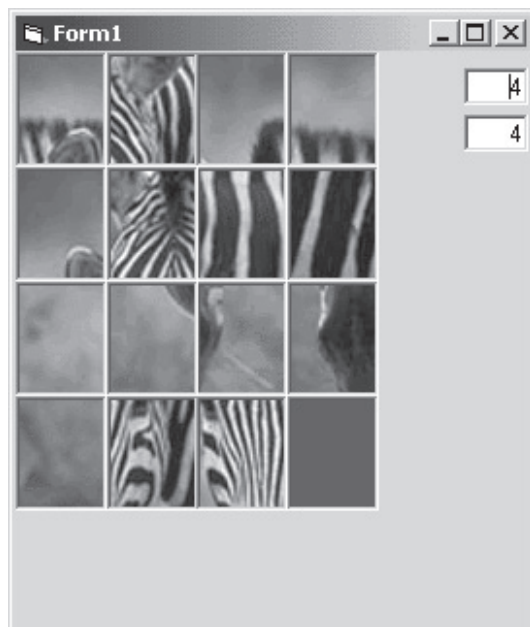
```

Vysvětlení

Jednotlivé kostičky jsou při vytváření ukládány do pole ve správném pořadí složeného obrázku, to dovolí kontrolovat složení. Každá kostička si pamatuje i svoji polohu, takže můžeme zabránit pokládání kostiček přes sebe.

Vlastnosti **Left** a **Top** nejsou deklarovány jako celočíselné proměnné a proto je nelze testovat na rovnost s požadovanou hodnotou, ale lze pouze testovat, zda jejich odchylka od požadované hodnoty je dostatečně malá. Zkuste si to!

Patnáctka



Obr. 50 G17.VBP

Co to dělá:

Také skládačka, ale jedna z kostiček je nahrazena „dírou“. Skládání spočívá ve vyměňování „díry“ a sousední kostičky pomocí šipek.

Jak:

Kostičky jsou vyrobeny na svých místech, poslední z nich je přebarvena, bude tvořit díru. Zároveň se ukládá pozice díry.

```
Private Sub Form_Click()  
...  
i = 1  
For Y = 1 To PY  
    For X = 1 To PX  
        Load pbPuzzle(i)  
        With pbPuzzle(i)  
            .Left = (X - 1) * VX  
            .Top = (Y - 1) * VY  
            .Width = VX  
            .Height = VY  
            .PaintPicture imObrazek.Picture, _  
                0, 0, VX, VY, (X - 1) * VX, _  
                (Y - 1) * VY, VX, VY
```



```

        .AutoRedraw = True
        .Visible = True
    End With
    Index(X, Y) = i
    i = i + 1
Next X
Next Y
pbPuzzle(i - 1).Line (0, 0)-(VX, VY), vbBlue, BF
DX = PX
DY = PY
...

```

Procedura TAH...

```

Private Sub Tah(Smer As Integer)
Dim XX As Variant
Dim YY As Variant
Dim NX As Integer
Dim NY As Integer
Dim pom As Integer

XX = Array(1, 0, -1, 0)
YY = Array(0, 1, 0, -1)

NX = DX + XX(Smer)
NY = DY + YY(Smer)

If NX < 1 Or NX > PX Or NY < 1 Or NY > PY Then
    'Beep
    Exit Sub
End If

With pbPuzzle(Index(NX, NY))
    .Left = (DX - 1) * VX
    .Top = (DY - 1) * VY
End With
With pbPuzzle(Index(DX, DY))
    .Left = (NX - 1) * VX
    .Top = (NY - 1) * VY
End With

pom = Index(DX, DY)
Index(DX, DY) = Index(NX, NY)
Index(NX, NY) = pom
DX = NX
DY = NY

End Sub

```

...slouží k počátečnímu zamíchání kostiček...

```
...  
For i = 1 To 100  
    Tah Int(Rnd * 4) + 1  
Next i
```

... i tahům hráče

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)  
If KeyCode >= 37 And KeyCode <= 40 Then  
    Tah KeyCode - 36  
End If  
End Sub
```

Poznámka

*Pro volání procedury **Form_KeyDown** je třeba v okně *Properties* nastavit vlastnost **KeyPreview** okna **Form1** na **TRUE**.*