Visual Basic for Application (VBA)

Visual Basic

- programovací jazyk od Microsoftu

VBA

- modifikace jazyka určená k programování v aplikacích pod Windows, skriptovací jazyk

skriptovací jazyk

- jeho překlad probíhá až za běhu programu, nevytváří se spustitelný soubor

překlad programu

- vytvoření spustitelného souboru z programového kódu, proběhne překlad z programovacího jazyka do strojového kódu

konvence při psaní kódu

- odsazování řádků pomocí tabulátoru, odsazuje se blok kódu, který patří do nějaké nadřízené struktury, například příkazy v cyklu
- VBA je case insensitive, nerozlišuje velká a malá písmena
- komentáře uvozujeme znakem apostrof (‘), jedná se o řádky kódu, které se nebudou při běhu programu provádět
- příkazy píšeme po jednom na jeden řádek, více příkazů na řádku oddělíme dvojtečkou
- provádění nekonečného cyklu (zacyklení programu) se ukončuje stisknutím kláves Ctrl + Pause

proměnná

- identifikátor označující místo v paměti, v různých fázích běhu programu může nabývat různých hodnot

konstanta

- identifikátor, který při celém běhu programu nabývá stejné hodnoty
- `Const PI = 3.14`

datový typ

- vymezuje hodnoty, kterých může proměnná nabývat

základní datové typy ve VB

- Integer – celá čísla na 2 bytech: 1, 22252, -10
- Long (Integer) – celá čísla s větším rozsahem než Integer (na 4 bytech)
- Double – reálná čísla: 0.5
- String – libovolný řetězec znaků, píše se do uvozovek: `"dnes je pekny den"`
- Boolean – pravdivostní hodnoty true nebo false
- Date – datum, VBA má dvě speciální funkce Time a Now

```
Dim d as Date
d = #10/5/2007 6:00:01 PM#
d = Now
```

deklarace proměnné

- rezervace jména proměnné a určení datové typu

- explicitní deklarace pomocí klíčového slova Dim přiřadí proměnné pevný datový typ

```
Dim cislo as Integer
Dim retez1, retez2 as String
```

- implicitní deklarace se provede až při přiřazení hodnoty, nepoužívá se klíčové slovo Dim, proměnné se nastaví datový typ variant

```
jmeno = "Honza"
```

- rezervace místa v paměti probíhá až při prvním přiřazení hodnoty

operátory

=	má dva významy, přiřazení hodnoty proměnné a zároveň funkce porovnání dvou hodnot s výstupní hodnotou true nebo false
<	menší než
>	větší než
<=	menší nebo rovno
>=	větší nebo rovno
<>	nerovnítko
+	sčítání
-	odčítání
*	násobení
/	dělení
^	umocnění
\	celočíselné dělení
mod	zbytek po dělení, zbytek = a mod b
Is	porovnání dvou objektů
Not	negace
And	konjunkce
Or	disjunkce
Ekv	ekvivalence
Imp	implikace
Xor	exklusive or neboli negace ekvivalence
&	spojení dvou řetězců

rozhodovací struktura If Then Else

- struktura provede blok příkazů pokud platí podmínka uvedená za klíčovým slovem If (podmínka vyhodnocena na true)
- strukturu lze doplnit o nepovinný blok příkazů za klíčovým slovem Else, které se provedou pokud podmínka za if splněna není

```
If a<0 Then
      MsgBox "zaporne cislo"
Else ‘nepovinný blok
      MsgBox "nezaporne cislo"
End If
```

rozhodovací struktura If Then ElseIf

- rozšíření struktury If
- v případě nesplnění podmínky za If se vyhodnocují další bloky s podmínkami uvozené klíčovým slovem ElseIf, pokud není splněna žádná z podmínek v blocích ElseIf, provede se blok příkazů za Else

```
Dim a,b as Integer
If a<0 Then
      MsgBox "zaporne cislo"
ElseIf a=0 Then
      MsgBox "cislo rovno nule"
ElseIf b = 5 Then
        MsgBox "dffgfg"
Else
      MsgBox "kladne cislo"
End If
```

rozhodovací struktura Select Case

- podobná rozhodovací struktura jako If Then ElseIf, rozdíl spočívá v tom, že Select Case vyhodnocuje jediný výraz a bloky uvozené klíčovým slovem Case se provádějí v závislosti na hodnotě vyhodnocovaného výrazu, pokud žádná Case hodnota neodpovídá výrazu, provede se blok příkazů za Else Select

```
Select Case body
      Case 1 To 10
            vysledek = "neuspel"
      Case 11 To 20
            vysledek = "uspel"
      Else Select
            vysledek = "chyba"
```

cykly

- cyklus je blok příkazů uvozený klíčovými slovy a podmínkou, který se provádí na základě splnitelnosti podmínky splněna podmínka

for cyklus

- cyklus s předem daným počtem opakování

```
For i = 1 To 10 Step
      a = i*a
Next i
```

for each cyklus

- cyklus se provede pro všechny hodnoty v poli nebo objekty v kolekci, nemusí se znát jejich počet

```
For Each File In Folder
      <blok prikazu>
Next
```

Cykly s podmínkou na začátku

- podmínka se vyhodnocuje před začátkem provádění bloku příkazů v cyklu
- vyhodnocování podmínky na začátku cyklu nám zaručí, že cyklus nemusí proběhnout ani jednou

while cyklus

– while si můžeme přeložit jako pokud = pokud je splněna podmínka, cyklus se provádí

```
a = 0
While a <= 10
      <blok prikazu>
      a = a+1
Wend
```

cyklus Do While ... Loop

– cyklus while s jinou syntaxí

```
a = 0
Do While a <= 10
      <blok prikazu>
      a = a+1
Loop
```

cyklus until

– dokud není splněna podmínka, blok příkazů v cyklu se provádí

– provádění cyklu skončí ve chvíli kdy se podmínka splní

cyklus Do Until ... Loop

```
a = 3
Do Until a = 0
      <blok prikazu>
      a = a-1
Loop
```

Cykly s podmínkou na konci

– podmínka se vyhodnocuje až po prvním provedení bloku příkazů v cyklu

– vyhodnocování podmínky na začátku cyklu nám zaručí, že cyklus proběhne minimálně jednou

cyklus Do ... While Loop

```
a = 3
Do
      <blok prikazu>
      a = a-1
Loop While a <= 0
```

cyklus Do .... Until Loop

```
a = 1
Do
      <blok prikazu>
      a = a-1
Loop Until a = 0
```

Příkaz Exit

– ukončí provádění cyklu a pokračuje plněním příkazů nacházejících se za cyklem

– platí pro všechny typy cyklů: Exit For, Exit Do

– lepší je pokrýt ukončení cyklu jeho podmínkou než používat příkaz Exit, může se stát, že to někdy nebude možné a ukončení pomocí Exit bude nevyhnutelné :-)