

1.5 Lekce 4 – Kalendářní a časové hodnoty

I když by bylo možné v předtím jmenovaných datových typech udržovat hodnoty pro čas a kalendář, z vývojářského a uživatelského pohledu by tato akce byla příliš nepraktická. Z tohoto důvodu VBA nabízí možnost práce s časem jiným (jednodušším) způsobem.

1.5.1 Kalendářní hodnoty

Do proměnné lze uložit pevnou kalendářní hodnotu pomocí zápisu ve tvaru *#měsíc/den/rok#*, který vychází z anglického formátování kalendářního data. Mřížky v zápise označují, že se jedná o celek jako datum, jinak by bylo provedeno dělení se zapsanými čísly.

```
Function PocetDniOdDnes (datum)
    dnes = Date
    PocetDniOdDnes = Abs (dnes - datum)
End Function
```

```
Function PocetDniOdNovehoRoku2023 (datum)
    ' 1. leden 2023 --> leden / 1 / 2023
    novyRok = #1/1/2023#
    PocetDniOdNovehoRoku = datum - novyRok
End Function
```

Obrázek 28 – Ukázka kódu práce s kalendářními hodnotami

Od kalendářních hodnot lze přičítat a odečítat počet dní obdobně jako při počítání příkladů. VBA umí pracovat i s přičtením a odečtením pevně zapsaných kalendářních hodnot. Aktuální datum lze zjistit pomocí funkce *Date*, která navrátí aktuální den v kalendáři.

1.5.2 Časové hodnoty

Podobně jako v případě kalendářních hodnot se pro proměnné zapisují hodnoty času ve tvaru *#hodiny:minuty:sekundy#*. Mřížky opět označují, že se jedná o celek ve formátu času. Editor VBA uzpůsobí zapsaný formát vloženého času do anglického formátu.

```
Sub ScheduleMeeting ()
    ' Zadáno ve formátu #22:35:44#
    ' --> 22 hodin 35 minut a 44 sekund
    meeting = #10:35:44 PM#

    ' Upozornění dvě hodiny předem (formát #02:00:00#)
    upozorneni = meeting - #2:00:00 AM#
End Sub
```

Obrázek 29 – Ukázka kódu práce s časovými hodnotami

U časových hodnot nelze přičítat ani odečítat samostatná čísla, ale pouze vždy časové hodnoty. Aktuální čas lze získat funkcí *Time*.

1.5.3 Práce s kalendářem a časem

Mezi další funkce pro práci s časem patří funkce *Now*, která navrátí aktuální datum a čas spojené v jedné hodnotě. Hodnota data a času lze zapsat společně ve tvaru *#měsíc/den/rok hodiny/minuty/sekundy AM/PM#*.

Funkce *Weekday* vrátí na základě vloženého data (první parametr), číselnou hodnotu dne v týdnu. Druhý parametr určuje, který den v týdnu je jako první. Výchozí hodnota je nastavena na neděli (hodnota *vbSunday*, vychází z anglicky mluvících zemí). Pro českou lokalizaci je nutné uvést hodnotu *vbMonday* (pondělí), aby pro nás funkce fungovala správně.

Mezi ostatní funkce, které navrátí číselnou hodnotu z parametru datumu patří:

- *Day* – Navrátí číselnou hodnotu dne datumu
- *Month* – Navrátí číselnou hodnotu měsíce datumu
- *Year* – Navrátí číselnou hodnotu roku datumu
- *WorksheetFunction.WeekNum* – Navrátí číselnou hodnotu týdne v roce datumu (začínající v pondělí, pokud je druhý parametr *vbMonday*)
- *Hour* – Navrátí číselnou hodnotu hodin datumu
- *Minute* – Navrátí číselnou hodnotu minut datumu
- *Second* – Navrátí číselnou hodnotu sekund datumu

```
' datum 13.8.2023
datum = #8/13/2023 10:55:01 AM#

' Navrátí hodnotu (den v týdnu) 7 (Neděle)
MsgBox (Weekday(datum, vbMonday))
' Navrátí hodnotu (den) 13
MsgBox (Day(datum))
' Navrátí hodnotu (měsíc) 8
MsgBox (Month(datum))
' Navrátí hodnotu (rok) 2023
MsgBox (Year(datum))
' Navrátí hodnotu (týden) 33
MsgBox (WorksheetFunction.WeekNum(datum, vbMonday))
' Navrátí hodnotu (hodina) 10
MsgBox (Hour(datum))
' Navrátí hodnotu (minuta) 55
MsgBox (Minute(datum))
' Navrátí hodnotu (sekunda) 01
MsgBox (Second(datum))
```

Obrázek 30 – Ukázka použití funkcí pro práci s datumem

Opakem funkcí pro převod určité informace z datumů je funkce *DateSerial*, která z drobných informací je schopna vytvořit kompletní datum. Funkci je vhodné použít při

kombinování rozložených dat. Funkce přijímá celočíselné parametry *rok*, *měsíc* a *den* v daném pořadí.

```
den = 17
mesic = 2 ' Únor
rok = 2001

' Vytvoří datum datového typu Date
datum = DateSerial(rok, mesic, den)
' Zobrazí "17.02.2001"
MsgBox (datum)
```

Obrázek 31 – Ukázka kódu použité funkce *DateSerial*

Pro pomoc s počítáním nových dat od předem zadaného data existuje funkce *DateAdd*. Funkce na základě prvního parametru *interval* zjistí, jakou jednotku času má přičítat/odečítat. Jednotky jsou zadávány do řetězce dle tabulky nastavení. Druhý parametr *number* určuje, kolik jednotek se má k datu přičíst nebo odečíst. Třetí parametr funkce *date* je pro datum, které určuje časový bod, od kterého se nové datum bude počítat.

```
Dim datum As Date
datum = #2/17/2001#

' o 8 týdnů později
noveDatum = DateAdd("ww", 8, datum)

' Zobrazí "14.4.2001"
MsgBox (noveDatum)
```

Obrázek 32 – Ukázka kódu použité funkce *DateAdd*

Nastavení	Popis
yyyy	Rok
q	Čtvrtletí
m	Měsíc
y	Pořadové číslo v roce
d	Den měsíce
w	Den v týdnu
ww	Týden
h	Hodina
n	Minuta
s	Sekunda

Tabulka 8 – Nastavení pro parametr *interval* v *DateAdd*, *DateDiff*, *DatePart*

Pro měření časových intervalů (čtvrtletí, týdnů, měsíců, čtvrtků atd.) mezi dvěma daty je možné použít funkci *DateDiff*. Funkce obsahuje 3 povinné parametry a 2 nepovinné. První parametrem *interval* se vyjadřuje, v jaké jednotce času bude funkce měřit. Druhý a třetí parametr (*date1*, *date2*) jsou daty, které ohraničují měřený úsek. Čtvrtý parametr *firstday-ofweek* určuje, který den v týdnu je jako první – pro Českou republiku by zde měla být nastavena hodnota *vbMonday*.

Poslední parametr *firstweekofyear* určuje, jakým způsobem se má započítat první týden v novém roce. Výchozím nastavením je týden, ve kterém se objeví datum 1. ledna (hodnota *vbFirstJan1*). Kromě výchozího nastavení existují nastavení *vbFirstFourDays* (první týden je ten, který má v sobě alespoň 4 dny v novém roce) a *vbFirstFullWeek* (první týden je ten, který má celý týden v novém roce).

```
Dim datum As Date
datum = #2/17/2001#
' o 8 týdnů později
noveDatum = DateAdd("ww", 8, datum)

' Spočítá počet týdnů mezi daty
pocetTydnu = DateDiff("ww", datum, noveDatum, vbMonday, vbFirstJan1)

' Zobrazí 8
MsgBox (pocetTydnu)
```

Obrázek 33 – Ukázka kódu použité funkce *DateDiff*

Funkce *DatePart* vrací ze vstupní kalendářní hodnoty část, která je specifikovaná v parametru *interval*. Jedná se o podobnou funkčnost jako mají funkce *Year*, *Month*, *Day*, *Hour*, *Minute* a *Second*. Parametry funkce jsou obdobné funkci *DateDiff*.

1.5.4 Úkoly k procvičení

- 1) Vytvořte makro *OdRoku2000*, které po spuštění oznámí uživateli informaci o počtu dní, které uběhly od 1. ledna 2000.
- 2) Vytvořte makro *ZaStoLet*, které po spuštění oznámí uživateli, jaký den v týdnu bude za 100 let od dnešního dne.
- 3) Vytvořte funkci *DoJeziska*, která zobrazí počet hodin do nejbližšího vánočního dne. Ověřte, zda bude funkce fungovat i v příštích letech.
- 4) Upravte funkci *DenVTydu*, aby jako argument přijala datum místo pořadí dne v týdnu a navrátila textovou hodnotu dne v týdnu, podle tohoto data.
- 5) Vytvořte makro *Nejblizi29*, které uživateli vypíše datum nejbližšího 29. února společně s dobou, která do tohoto data uplyne.