ZÁKLADY PROGRAMOVÁNÍ A ALGORITMIZACE VE VBA

CO JE TO ALGORITMUS?

- Algoritmus je přesný návod či postup, kterým lze vyřešit daný typ úlohy. Pojem algoritmu se nejčastěji objevuje při programování, kdy se jím myslí teoretický princip řešení problému. Obecně se algoritmus může objevit v jakémkoli jiném vědeckém odvětví

- Algoritmus je pracovní postup, který splňuje tyto povinné vlastnosti
  - » Rezultatitvnost – to znamená, že algoritmus má vždy nějaký výsledek
  - » Finitnost (konečnost) – to znamená, že někdy skončí. (Skončí po konečném počtu provedených kroků)
  - » Elementárnost (jednoduchost) popisu - algoritmus je popsán konečným počtem základních instrukcí. Tedy takových, o kterých je jasné, jak se provedou (neumožňují tedy žádný osobitý výklad některého vykonavatele).
  - » Determinovanost (jednoznačnost) – postup práce je jasně daný a vždy závisí pouze na popisu algoritmu a na vstupu. Na průběh algoritmu nemá žádný vliv náhoda nebo svobodná vůle vykonavatele.

Algoritmizace = proces vytváření a sestavování algoritmů
Programování = zakódování algoritmu do zvoleného programovacího jazyka

K ČOMU SLÚŽI MAKRO ?

- Makro od Gréckého slova „μακρό – „veľké“alebo „ďaleko“je postupnosť inštrukcií, ktoré sa začnú po aktivácií makra realizovať postupne. Dajú sa doňho uložiť často používané akcie v programe Visual Basic. Makro slúži k definovaniu symbolov využívaných pri podmienenom preklade, ale aj k definovaniu komplikovaných sekvencií, ktoré sú následne v zápise zdrojového kódu.
- Výhodou makra je, že ich preddefinovanie sa pri preklade zdrojového kódu automaticky zamení a všetky ich výskyty, čo minimalizuje chyby, ktoré by priniesli mnohonásobnú editáciu kódu.

Príklady často opakujúcich sa krokov v exceli pri ktorých možno použiť makro:

- Úprava vzhľadu tabulky
- Obsluha ovládajúcich prvkob
- Definovanie vlastných funkcií

VBA editor je vstavanou súčasťou každého Excelu a spúšťa sa pomocou klávesnicovej skratky Alt + F11. Skladá sa z hlavného okna, v ktorom otvoríme module a z postranných panelov, prostredníctvom ktorých vieme nakonfigurovať rôzne nastavenia, čo nám zjednoduší Excel VBA programovanie. Na editovanie VBA kódu slúži VBA editor, ktorý je súčasťou každého Office program.

VBA makro sa skladá z Modulov (Module), Formulárov (Form), a Štýlov (Style).

- Module - je základný stavebný prvok, ktorý v sebe bude obsahovať VBA kód.
- Form - slúži na vytvorenie užívateľského prostredia ktoré umožní používanie programu

koncovému užívateľovi

Príklad : ako vymazať bunky A1:B2 (výsledok bude taký, že užívateľ klikne na tlačidlo a tým zmaže všetko, čo je v bunkách A1 až B2 napísané.)

```
Sub vyukaexcelu()

' mojemakro

   Range("A1:B2").Select

   Selection.ClearContents

End Sub
```

Makrá bez parametrov

- Používajú saa pre definíciu konštánt, kedy sa miesto konštanty používa nejaké špeciálne slovo. Pravidlom je písať identifikátor makra bez veľkých písmen. Okrem štandartných makier sa dajú vytvárať aj valstné makra (definovanie počtov prvkov v poli). Makro sa definuje za direktivou #define a dá sa zrušiť direktivou #undef.

Makrá s parametrami

- Makrá môžu byť argumenty, ktoré sú uzavreté v gulatých zátvorkách a pokiaľ obsahujú viac ako jeden argument, tak sú oddelené čiarkou. Medzi pomenovanie makra a zátvorkou obsahujúcou argumenty nesmie byť medzera.

JAKÝ JE ZÁKLADNÍ ROZDÍL MEDZI MAKREM A FUNKCIÍ ?

Funkce jsou již v Excelu předdefinovány. Pomocí makra můžeme sloučit několik funkcí dohromady a tím si vytvořit v Excelu nad tabulkou, sešitem, vlastní novou funkcionalitu.

A, pojem Funkce obecně:

- funkce je příkaz či sled příkazů vykonávaných jako celek a tvořících uzavřenou jednotku,
- kromě toho, že funkce provádí nějaké příkazy v určitém sledu (pořadí), tak také vrací určitou hodnotu, a tuto hodnotu můžeme uložit do proměnné (resp . do paměti PC) a následně zpracovat.
- např. funkce y = log(x), tzn. známý zápis funkce „logaritmus“, ze zadaného čísla „x“ vypočítá jiné číslo a vyjádří (resp. na-vrátí) je jako číslo „y“. Funkce v programech, tzn i v Excelu, se chovají stejně, tzn tak jak je v předchozím komentáři zmíněno,
- návratová hodnota funkce může obsahovat buď výsledek výpočtu, nebo může informovat o úspěšném či neúspěšném výsledku průběhu funkce.

A1, VBA-vytvoření vlastní funkce

Jednotlivé verze Excelu mají integrovány řádově stovky funkcí. Přesto se můžeme dostat do situace, kdy by se nám hodila funkce, která v Excelu není. Nebo nás nebaví opakovaně zapisovat dlouhý vzorec obsahující více funkcí a chceme si vytvořit vlastní funkci, která tuto kombinaci funkcí nahradí.

Příklad:

V mém případě chci vytvořit jednoduchou funkci, která spočte obsah obdélníka na základě dvou vstupních buněk.

Návod:

V editoru maker (karta Vývojář / tlačítko Visual Basic), vytvořím nový modul a zapíšu funkci. V mém případě vypadá takto:

Function Obsah_obdelnika(Delka, Sirka)
Obsah_obdelnika = Delka * Sirka
End Function

Vysvětlení/poznámka:

- *Function Obsah_obdelnika(Delka, Sirka)*
  - *Function* ... říká, že je to funkce,
  - *Obsah_obdelnika* … je název funkce,
  - *Delk*a a *Sirka* … jsou názvy vstupních hodnot (parametry funkce)
- *Obsah_obdelnika = Delka * Sirka*
  - obsah je roven délce krát šířce
- *End Function*
  - představuje konec zápisu funkce

Od tohoto okamžiku se s mojí (výše definovanou) funkcí pracuje jako s jakoukoliv jinou. Jen si musím uvědomit, že tato funkce existuje v zásadě jen v souboru (excelovské tabulce), kde jsem ji vytvořil.

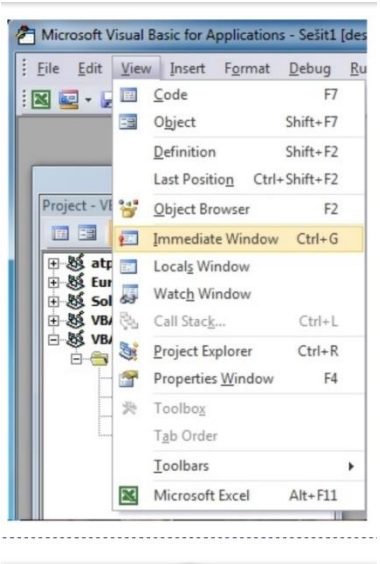
B, pojem „Makro“ (v kancelářských aplikacích Microsoft Office):

- v kancelářských aplikacích (Microsoft Office, příp. OpenOffice a podobně) označuje makro posloupnost akcí nebo funkcí, které usnadňují určitou činnost (např. v programech Word a Excel). Používají se většinou jako posloupnost kroků při výpočtech, úpravách textu a podobně. Jednoduchým příkladem může být makro pro odstranění speciálních znaků (např. tabulátoru, či odřádkování) v textu nebo makro aktivující určitou akci po stisknutí vybrané kombinace kláves.
- jinak laicky řečeno: „Představte si, že v Excelu (nebo ve Wordu, Accessu nebo v jiné aplikaci MS Office) děláte opakovaně nějakou rutinní činnost. A protože je zbytečné, abyste to dělali znovu a znovu úplně stejně a ztráceli tím čas, potřebujete, aby to dělal Excel automaticky. Aby to ale dělat mohl, musíte ho to nejdříve "naučit". Chcete tedy Excelu jakoby říci "teď ukážu, co chci, abys dělal, a pak to uděláš sám pokaždé, když kliknu na tlačítko". K tomuto účelu slouží makro.“
- Technicky je makro aplikace napsaná v programovacím jazyce Visual Basic for Applications (VBA), což je jazyk používaný v Microsoft Office.

CO ZNAMENÁ POJEM DEBUG?

= LADĚNÍ, neboli vychytání chyb, odstranění chyb z počítače programu

- Základem úspěšného ladění je zjišťování hodnot proměnných a návratových hodnot funkcí, protože to nám často napoví, kde je chyba. Lze to kontrolovat přes MsgBox zařazené do kódu, nebo přes Run to cursor a podržení myši nad proměnnou, nebo konečně přes Debug.Print, které vypisuje hodnoty proměnných či návratové hodnoty funkcí do okamžitého okna (které se v editoru VB otevře pomocí Ctrl+G; je potřeba mít ho otevřené předem).
- SPUŠTĚNÍ
  Pokud okno nevidíme lze jej spustit klávesovou zkratkou Ctrl+G nebo přes menu View - Immediate Window.



Předávání parametrů

K zobrazení v okně se používá jednoduchý příkaz:

```
Debug.Print
```

Praktický příklad

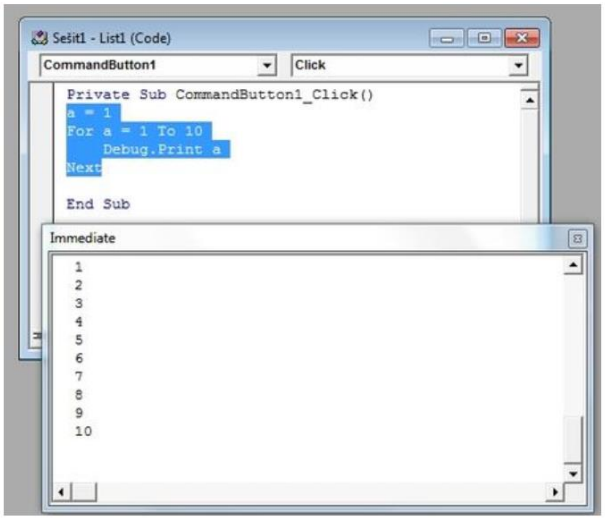
Použití příkazu ukážu na jednoduchém VBA kódu:

```
a = 1
```

```
For a = 1 To 10
```

```
    Debug.Print a
```

```
Next
```



Rozšíření

Můžeme doplnit do výpisu ještě informaci, která proměnná nabývá vypsané hodnoty. Pokud vypisujeme jen jednu proměnnou, nemá to cenu, ale pokud výpisu máme v programu hodně, je vhodné vědět, čeho se týkají.

```
Debug.Print "Hodnota a: " & a
```

Spočít počet listů v aktuálním sešitě s výpisem do ladicího okna.

```
Sub TestPocetListu()
```

```
    Debug.Print "Počet listů v sešitu: " &
ActiveWorkbook.Worksheets.Count
End Sub
```

Vypis oblasti buněk

```
Set r = Range("A1:C3")
    Debug.Print "Moje oblast: " & r.Address
```

Další využití okna immediate

Lze využít jako kalkulačku. Zadáte otazník a požadovaný výpočet.

```
? 2 + 2
```

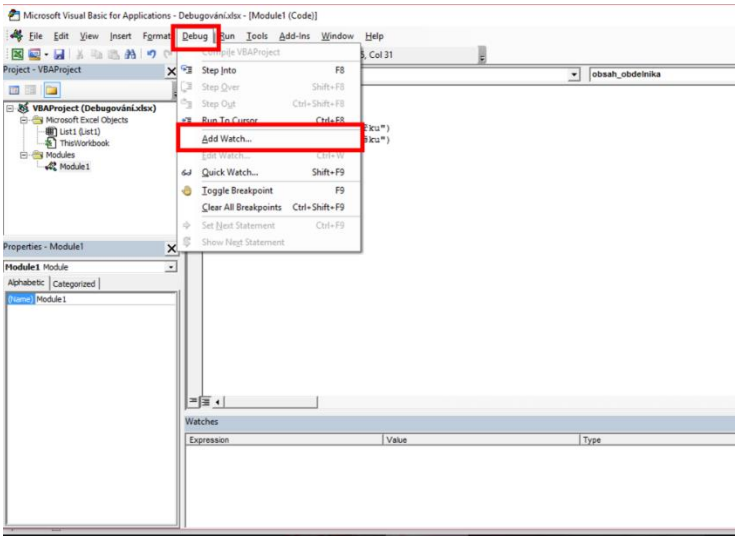
Výsledek 4 :)

Možnosti hledání chyb v kódu VBA (debug)

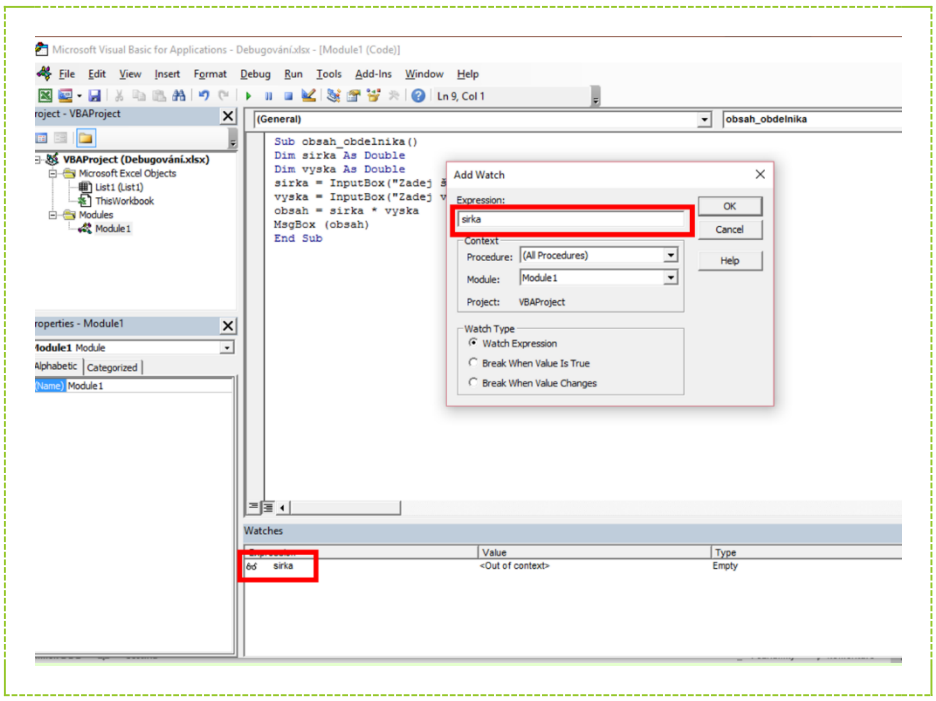
Sledování hodnot proměnných

Chyby často vzniknou tím, že se do proměnné načte nebo uloží jiná hodnota, než která by tam měla být. Proto je šikovné sledovat, jak se hodnota proměnné mění. Ideální je samozřejmě spojit sledování proměnné s krokováním, a sledovat, jak se proměnné mění po jednotlivých krocích.

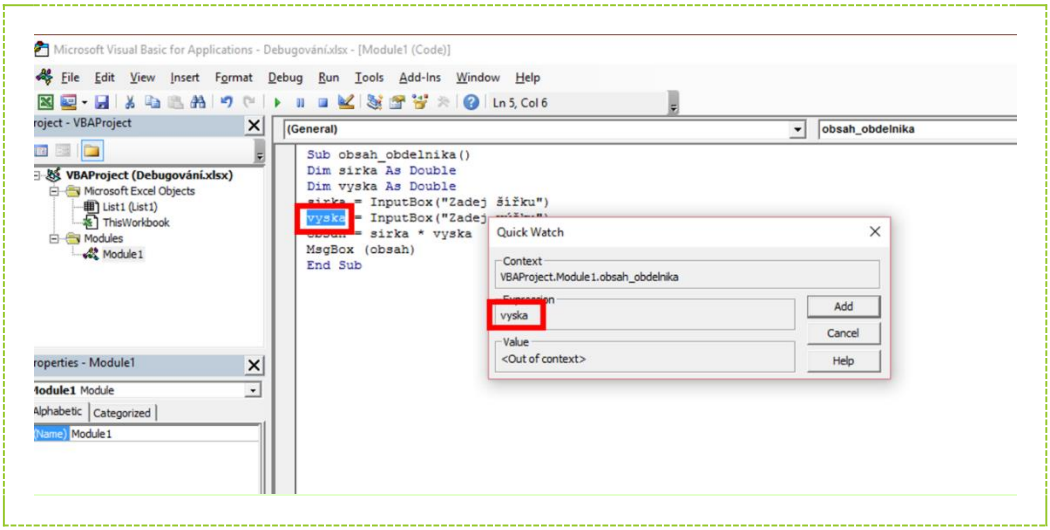
Sledovanou proměnnou přidáme přes Debug / Add watch..



V dialogu zapíšeme název proměnné. Proměnná se pak objeví v přehledu dole - a to jak její hodnota, tak datový typ.

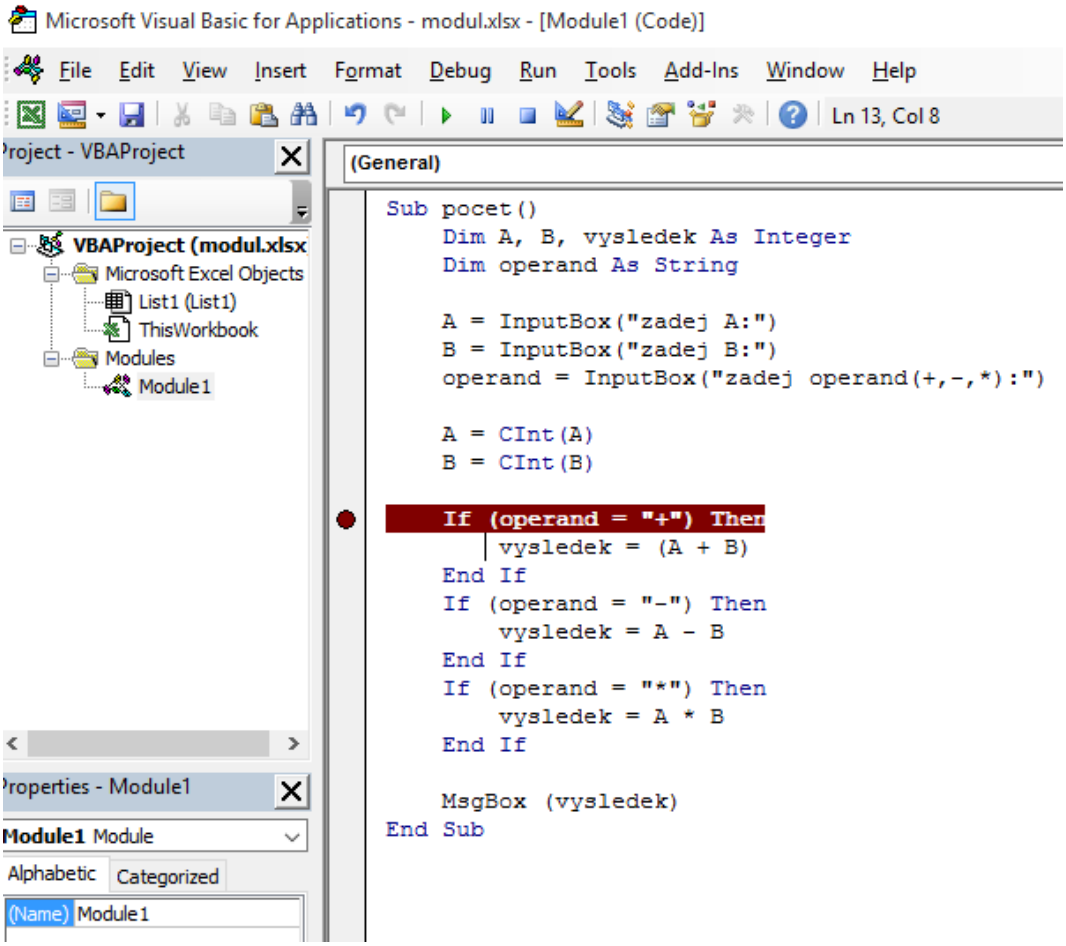


Při krokování makra se pak tyto hodnoty mění.
Sledovanou proměnnou můžeme přidat také jednorázově - pokud se chceme v konkrétním okamžiku podívat, jakou hodnotu má vybraná proměnná.
Stačí v textu označit název proměnné, a pak Debug / Quick watch... .



Krokování

Breakpoint – slouží k přerušení programu v daném bodu, následně je pak možné použít krokování -> step into – procházení programu krok po kroku ve všech funkcích apod…
Step over - přeskočí podmínku nebo funkci, která nás nezajímá.
Step out – vyskočí např. z cyklu.



Při zastaveném programu můžeme kurzorem najet na jednotlivé proměnné a vidět jejich hodnoty. Po stisknutí F8 (step into) krokujeme program dále a vidíme, jak postupně probíhá (které podminky jsou splněny, které ne apod…). Můžeme program dokrokovat do konce nebo tlačítkem continue nechat běžet program samovolně dál (zelený trojúhelník (jako play)).

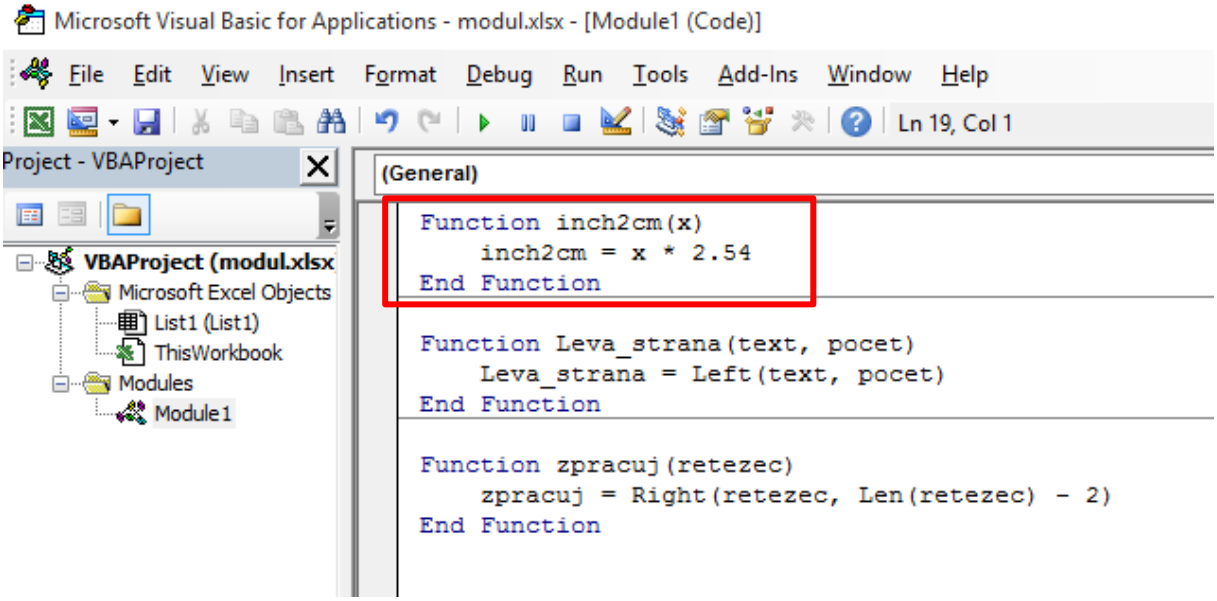
Tímto docílíme toho, že vidíme jak program funguje a případně co předcházelo chybě v programu, nebo proč je výstup danného kódu jiný než jsme očekávali.

NADEFINUJTE FUNKCI INCH2CM(X), KTERÁ DÉLKU X VYJÁDŘENOU V PALCÍCH PŘEVEDE NA CENTIMETRY.

a. Funkce se jmenuje inch2cm se vstupním parametrem x.
b. Návratová hodnota funkce je parametr x / 2,54.
c. Na obrázku v prvním sloupci vidíme pojmenování naší funkce, čili Inch2cm.
d. Vstupní parametr máme pouze jeden a vidíme ho ve sloupečku B. Což je naše hodnota délky v palcích.

	A	B	C	D	E
1		Vstupní data		Volání makra	Výstup makra
2	Inch2cm	2		=inch2cm(B2)	5,08
3	Leva_strana	pondělí	2	=Leva_strana(B3;C3)	po
4	zpracuj	CZ00216305		=zpracuj(B4)	00216305

e. Po vydělení této hodnoty (5,08) konstantou 2,54 (což je hodnota jednoho palce v centimetrech) dostaneme délku v centimetrech. Tato hodnota je naší návratovou hodnotou funkce inch2cm.
f. Nadefinování funkce inch2cm provedu ve Visual Basicu, který si otevřu stisknutím kláves Alt a F11. Poté si v hlavním menu V.B. rozkliknu Insert a kliknutím na Module vložím nový modul.
g. Poté do okna Modulu napíšu Function Inch2cm (x). Jakožto funkci inch to cm s proměnnou x.
h. Řádek odentruju. Na další řádek musím napsat, co chci aby moje funkce dělala. V tomhle případě chci, aby funkce převáděla palce na centimetry.
i. Takže můj vstupní parametr x budu zadávat v palcích. Jeden palec má délku 2,54 centimetrů. Abych dostala hodnotu v centimetrech, musím palce vynásobit hodnotou 2,54. Naše funkce bude vypadat takto: Inch2cm = x * 2,54.
j. Řádek opět odentruju, aby V.B. funkci ukončil a uložil.



CO TO JE ASCII TABULKA?

- american standard code for information interchang (americký standardní kód pro výměnu informací)
- kódová tabulka, která definuje znaky anglické abecedy a jiné znaky používané v informatice
- na češtinu nestačí – nezakóduju např. č,š,..
- 48 – 57 čísla
- 69 – 90 velká písmena
- 97 – 122 malá písmena
- interpunkce
- speciální znaky
- základní původní 7 bitové => 128 znaků, pro potřeby dalších jazyků (rozšíření znakové sady) se používá 8 bitové rozšíření =>dalších 12 znaků
- pro potřeby jednotlivých jazyků různé kódové tabulky (význam kódů nad 127 se může lišit)
- 1. verze r. 1963 – neobsahovala malá písmena a některé znaky; r. 1967 základ většiny kódování znaků
- žádné formátování (tučné, kurzíva, ...)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

The ASCII code

American Standard Code for Information Interchange

www.theasciicode.com.ar

ASCII control characters

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de línea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`			
33	21h	!	65	41h	A	97	61h	a			
34	22h	"	66	42h	B	98	62h	b			
35	23h	#	67	43h	C	99	63h	c			
36	24h	$	68	44h	D	100	64h	d			
37	25h	%	69	45h	E	101	65h	e			
38	26h	&	70	46h	F	102	66h	f			
39	27h	'	71	47h	G	103	67h	g			
40	28h	(	72	48h	H	104	68h	h			
41	29h	)	73	49h	I	105	69h	i			
42	2Ah	*	74	4Ah	J	106	6Ah	j			
43	2Bh	+	75	4Bh	K	107	6Bh	k			
44	2Ch	,	76	4Ch	L	108	6Ch	l			
45	2Dh	-	77	4Dh	M	109	6Dh	m			
46	2Eh	.	78	4Eh	N	110	6Eh	n			
47	2Fh	/	79	4Fh	O	111	6Fh	o			
48	30h	0	80	50h	P	112	70h	p			
49	31h	1	81	51h	Q	113	71h	q			
50	32h	2	82	52h	R	114	72h	r			
51	33h	3	83	53h	S	115	73h	s			
52	34h	4	84	54h	T	116	74h	t			
53	35h	5	85	55h	U	117	75h	u			
54	36h	6	86	56h	V	118	76h	v			
55	37h	7	87	57h	W	119	77h	w			
56	38h	8	88	58h	X	120	78h	x			
57	39h	9	89	59h	Y	121	79h	y			
58	3Ah	:	90	5Ah	Z	122	7Ah	z			
59	3Bh	;	91	5Bh	[	123	7Bh	{			
60	3Ch	<	92	5Ch	\	124	7Ch	|			
61	3Dh	=	93	5Dh	]	125	7Dh	}			
62	3Eh	>	94	5Eh	^	126	7Eh	~			
63	3Fh	?	95	5Fh	_						

theASCIIcode.com.ar

Extended ASCII characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	á	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	≡
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ò	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	▉	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	▌	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

frequently-used (spanish language)		vowels acute accent (spanish language)		vowels with diaresis		mathematical symbols		commercial / trade symbols		quotes and parenthesis	
ñ	alt + 164	á	alt + 160	ä	alt + 132	½	alt + 171	$	alt + 36	"	alt + 34
Ñ	alt + 165	é	alt + 130	ë	alt + 137	¼	alt + 172	£	alt + 156	'	alt + 39
@	alt + 64	í	alt + 161	ï	alt + 139	¾	alt + 243	¥	alt + 190	(	alt + 40
¿	alt + 168	ó	alt + 162	ö	alt + 148	¹	alt + 251	¢	alt + 189	)	alt + 41
?	alt + 63	ú	alt + 163	ü	alt + 129	³	alt + 252	¤	alt + 207	[	alt + 91
¡	alt + 173	Á	alt + 181	Ä	alt + 142	²	alt + 253	®	alt + 169	]	alt + 93
!	alt + 33	É	alt + 144	Ë	alt + 211	ƒ	alt + 159	©	alt + 184	{	alt + 123
:	alt + 58	Í	alt + 214	Ï	alt + 216	±	alt + 241	ª	alt + 166	}	alt + 125
/	alt + 47	Ó	alt + 224	Ö	alt + 153	×	alt + 158	º	alt + 167	«	alt + 174
\	alt + 92	Ú	alt + 233	Ü	alt + 154	÷	alt + 246	°	alt + 248	»	alt + 175

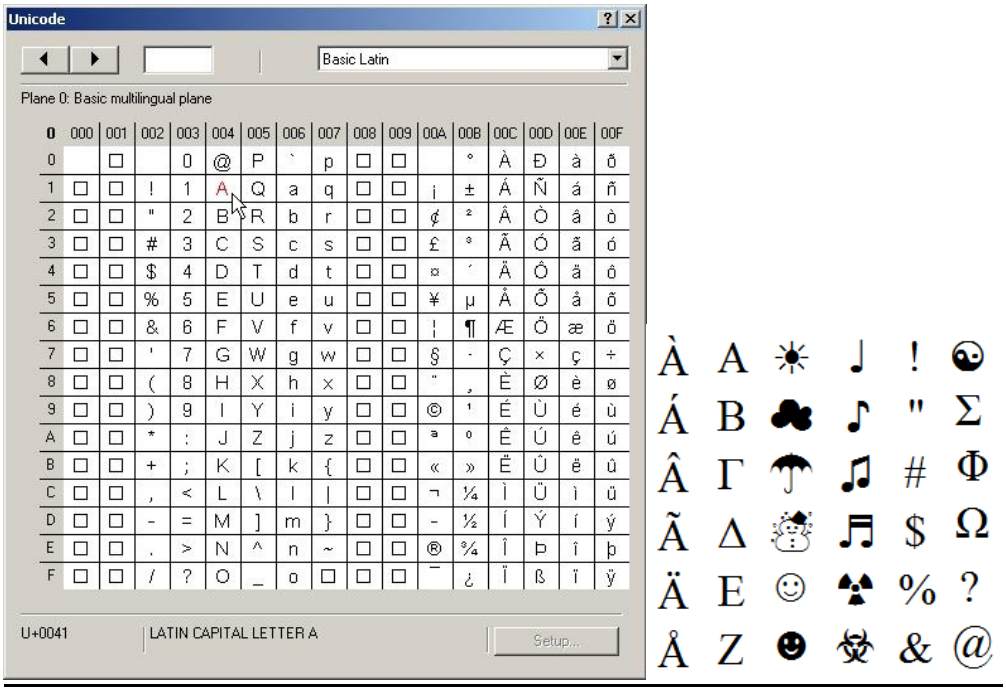
CO TO JE A PROČ SE POUŽÍVÁ UNICODE?

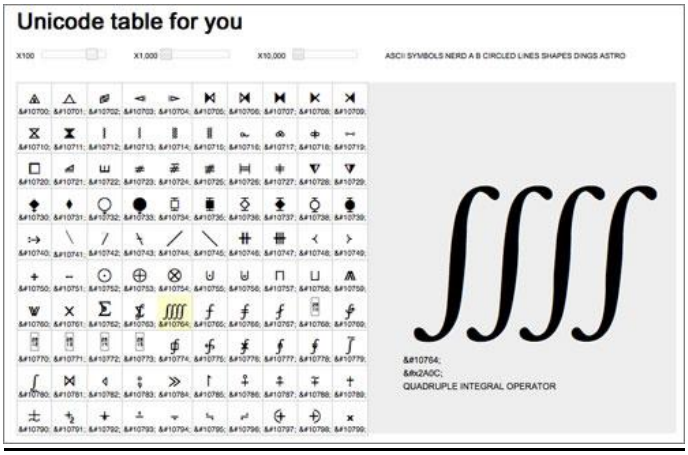
- technická norma pro oblast výpočetní techniky
- používá se proto, že umí zakódovat všechny znaky a proto, že umí pracovat s různými jazyky najednou (naráz zvládá např. francouzštinu a ruštinu)
- pro většinu písem používaných v současnosti
- 129 písem
- 120.000 znaků
- sady tabulek pro vizuální referenci, popisu metod kódování
- kóduje pro všechny jazyky = více mezinárodní
- několik způsobů reprezentace textů různými znakovými kódy (nejpoužívanější UTF-8 a UTF-16; zastaralé UCS-2
- UTF-8 používá pro ASCII znaky na zakódování 1 byte (8 bitů) a mají stejné kódové hodnoty jako v ASCII (tzn. Unicode, konkrétně UTF-8, má např. pro A stejný kód jako je v ASCII tabulce)
- všechny verze Unicode od 2.0 a výše jsou kompatibilní (znaky pouze přidávány, existující znaky nejsou vyřazeny nebo přejmenovány)
- nejnovější verze Unicode 8.0 – červen 2015
- cíle standardu Unicode: jednotnost, jednoznačnost, univerzálnost, maximální využití

Table of UNICODE codes,
for **Czech, Hungarian, Polish, Scandinavian** and some other Central European Languages.
The hexadecimal digits hhh used in the &#Xhhh; code.

Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code
Ā	100	Đ	110	Ę	118	Ķ	136	Ń	143	Ó	d3	Ś	15a	Ű	170
ā	101	đ	111	ę	119	ķ	137	ń	144	ó	f3	ś	15b	ű	171
Ă	102	Ď	10e	Ě	11a	Ĺ	139;	Ņ	145	Œ	152	Š	160	Ų	172
ă	103	ď	10f	ě	11b	Í	13a	ņ	146	œ	153	š	161	ų	173
Ą	104	Ē	112			Ļ	13b	Ň	147	ŕ	155	Ţ	162	Ÿ	178
ą	105	ē	113	Ģ	122	ļ	13c	ň	148	Ŗ	156	ţ	163	Ź	179
Ć	106	ĕ	115	ģ	123	Ľ	13d	Ō	14c	ŗ	157			ź	17a
ć	107	Ė	116	Ī	12a	ľ	13e	ō	14d	Ř	158	ť	165	Ż	17b
Č	10c	ė	117	ī	12b			Ő	150	ř	159			ż	17c
č	10d			Į	12e	Ł	141	ő	151	Ş	15e			Ž	17d
				į	12f	ł	142			ş	15f			ž	17e

Example: Ł = Ł





JAKOU HODNOTU VRÁTÍ PŘÍKAZ LEFT("PONDĚLÍ", 2)?

a. Funkce se jmenuje Leva_strana. Vstupními parametry této funkce jsou text a počet.
b. Návratovou hodnotou funkce Leva_strana je příkaz Left se vstupními parametry text a počet. Příkaz Left vrací z námi zadaného řetězce ve vstupním parametru text počet znaků z levé strany podle hodnoty ve vstupním parametru počet.
c. Řetězec je v podstatě slovo – neboli konečná posloupnost symbolů dané abecedy.
d. Parametry musí být odděleny středníkem.
e. Na obrázku v prvním sloupci vidíme pojmenování naší funkce, čili Leva_strana.
f. Naši funkci v excelu zavoláme zadáním rovnítka a napsáním leva, excel by nám měl sám nabídnout naši naprogramovanou funkci Leva_strana. Příklad zadání vidíte ve sloupci D.
g. Naším vstupním parametrem pro text je slovo Pondělí ve sloupci B a vstupní parametr pro počet je 2 ve sloupci C.

h. Po zavolání funkce Leva_strana zadáváme nejprve parametr Pondělí, oddělíme středníkem a můžeme zadat parametr 2.
i. Výstupem této funkce budou 2 znaky z levé strany zadaného řetězce.

JAKOU HODNOTU VRÁTÍ FUNKCE ZPRACUJ PRO HODNOTU PARAMETRU RETEZEC = "CZ00216305"? Function zpracuj(retezec) zpracuj = Right(retezec, Len(retezec) - 2) End Function.

a. Funkce zpracuj má vstupní parametr řetězec.
b. Návratová hodnota funkce zpracuj je výsledek příkazu Right (řetězec, Len (řetězec) - 2).
c. Příkaz Right (řetězec, Len(řetězec)-2) nejprve vykoná příkaz Len (řetězec). Příkaz Len je od anglického slova Length, což je délka. Takže příkaz Len se vstupním parametre řetězec nám vrátí délku zadaného řetězce číselně.
d. Od této hodnoty následně odečteme 2. Tímto jsme vyřešili příkaz Len (řetězec) -2.
e. Zbyde nám příkaz Right (řetězec, výsledek předchozí operace).
f. Vstupními parametry příkazu Right jsou řetězec a počet.
g. Příkaz Right vrací řetězec obsahující zadaný počet znaků z pravé strany řetězce.
h. Na obrázku v prvním sloupci vidíme pojmenování naší funkce, čili zpracuj.
i. Naši funkci v excelu zavoláme zadáním rovnítka a napsáním zpracuj, excel by nám měl sám nabídnout naši naprogramovanou funkci zpracuj. Příklad zadání vidíte ve sloupci D.
j. Naším vstupním parametrem pro řetězec je CZ00216305 ve sloupci B.
k. Výstupem této funkce bude 8 znaků z pravé strany zadaného řetězce.

	A	B	C	D	E
1		Vstupní data		Volání makra	Výstup makra
2	Inch2cm	2		=inch2cm(B2)	5,08
3	Leva_strana	pondělí	2	=Leva_strana(B3;C3)	po
4	zpracuj	CZ00216305		=zpracuj(B4)	00216305