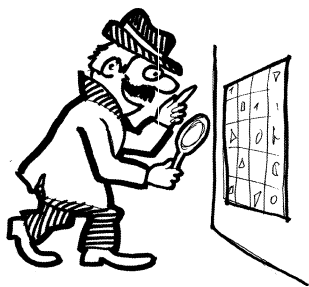


Počítačová bezpečnost ve výuce informatiky

(3. část – luštění jednoduché záměny, frekvenční analýza)

MICHAL MUSÍLEK – ŠTĚPÁN HUBÁLOVSKÝ

Přírodovědecká fakulta Univerzity Hradec Králové



V minulých článcích našeho seriálu [7], [8] jsme se naučili šifrovat otevřený text na šifrový a naopak dešifrovat šifrový text zpět na otevřený. Dešifrovat znamená převádět šifrový text na otevřený se znalostí kryptografické metody a použitého klíče. Pokud se jedná o jednoduchou záměnu, znamená to, že známe příslušnou převodovou tabulku. Zcela jinou činností je luštění, kdy šifrový text na otevřený nepřevádíme.

vádí příjemce, kterému byla zpráva určena, ale třetí osoba, která chce odhalit tajemství šifrovaného textu, určit použitou kryptografickou metodu a klíč. Pokud je tento člověk vybaven potřebnými znalostmi, označujeme ho jako *kryptoanalytika*.

Ve 2. části našeho seriálu jsme odvodili, že všech možných jednoduchých záměn v rámci znaků mezinárodní abecedy (26 písmen) je $26! = 49 \cdot 10^{21}$. Pokud vezmeme jen záměny, které jsou inverzní samy k sobě, bude jich mnohokrát méně, ale i tak zbude $7,9 \cdot 10^{12}$ (7,9 biliónů) různých převodo-

vých tabulek. Z tohoto pohledu se zdá být jednoduchá záměna na první pohled bezpečným způsobem utajení obsahu zprávy. Realita je ovšem jiná. Na skutečnost, že jednoduchou záměnu lze poměrně snadno luštit, přišli jako první patrně arabští učenci v 9. století našeho letopočtu. Simon Singh ve své knize [5] odkazuje na překvapivý objev, k němuž došlo v Sülejmanově osmanském archivu v Istambulu v roce 1987. Byl zde nalezen *Rukopis o dešifrování kryptografických zpráv* z pera al-Kindího, arabského filosofa, lékaře, astronoma, matematika, hudebníka a jazykovědce.

Al-Kindí ve dvou stručných odstavcích svého rukopisu popsal princip frekvenční analýzy, která je jedním z nejdůležitějších nástrojů klasické kryptoanalýzy. Pomocí ní nejprve rozhodneme, zda je šifrový text vůbec zašifrován pomocí jednoduché záměny, nebo jinou kryptografickou metodou. V případě, že se jedná o jednoduchou záměnu, určíme pomocí frekvenční analýzy, buď přesně, nebo alespoň přibližně s nutností „dolaďení“ použitím jiných metod, převodovou tabulku použitou u dané šifrové zprávy.

Je pravděpodobné, že ke vzniku frekvenční analýzy přispělo studium koránu, které bylo tak důkladné, že zkoumalo nejen četnost výskytu jednotlivých slov v súrách, ale dokonce četnost jednotlivých písmen. V arabštině se velmi často vyskytují písmena *a* a *l*, jistě i proto, že spolu tvoří určitý člen *al-*, zatímco četnost písmene *j* je asi desetkrát nižší. Frekvence, neboli četnost výskytu jednotlivých písmen abecedy je ovšem v každém jazyce jiná. Frekvenční analýzou šifrovaného textu, jemuž odpovídá otevřený text v angličtině se zabývají Simon Singh [5] a Vlastimil Klíma [2]. Komplikovaná by pro nás byla analýza jazyků, které používají jinou sadu znaků, např. ruština s azbukou. Ale ani jazyky, které používají latinu to nemají úplně jednoduché. Např. pro italštinu jsou typické zdvojené souhlásky, zdvojují se ale jen některé z nich, takže při prostém použití převodové tabulky by je kryptoanalytik snadno odhalil (viděl by v šifrovém textu dvojité znaky a k nim by přiřazoval příslušné souhlásky). Proto se před vlastním šifrováním italský otevřený text zpravidla upraví tak, že se všechny zdvojené souhlásky nahradí jednoduchými a pak se teprve provede šifrování.

V českém jazyce je zase zvykem odstranit veškerou diakritiku (čárky a háčky) a teprve potom šifrovat. Písmeno *ch*, které je pozůstatkem spřežkového pravopisu, se zpravidla šifruje jako dvě samostatná písmena *c* a *h*. Tak se z otevřeného českého textu stane text bez diakritiky, který používá 26 písmen mezinárodní abecedy, stejně jako angličtina. Jak je to s frekvencí jednotlivých hlásek v českém textu zjistíme, když vezmeme nějaký

delší text a spočítáme četnost jednotlivých písmen. Diakritiku si přitom také odmyslíme. Tak například první článek našeho seriálu sestával celkem z 11 444 písmen. Relativní četnost jednotlivých písmen, vyjádřená v procentech byla:

A	B	C	D	E	F	G	H	I	J	K	L	M
9,0	1,5	3,6	3,1	10,4	0,6	0,6	2,1	7,5	2,2	3,6	3,3	3,8
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6,3	8,3	3,8	0,0	4,8	5,4	5,6	3,9	4,0	0,1	0,3	2,8	3,3

Mnohem důkladnější a přesnější je statistika frekvencí jednotlivých písmen získaná analýzou textů sestávajících z 5 396 882 písmen Centrem zpracování přirozeného jazyka Fakulty informatiky Masarykovy univerzity v Brně [3]:

A	B	C	D	E	F	G	H	I	J	K	L	M
8,8	1,7	3,6	3,6	10,5	0,4	0,3	2,3	7,7	2,0	3,8	4,1	3,3
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6,7	8,3	3,5	0,0	5,2	5,4	5,6	3,8	4,4	0,1	0,1	2,7	3,1

Porovnáním obou tabulek frekvencí vidíme, že přesné hodnoty relativní četností písmen se sice trochu liší, ale pořadí nejčastějších písmen zůstává stejné. V obou případech jsou pět nejčastěji používaných písmen E, A, O, I a N. Frekvenční analýza nám pomůže v luštění tím víc, čím je šifrový text delší a čím víc je čistě český (bez příměsi cizích slov). Jaký postup luštění můžeme doporučit?

Proveďte frekvenční analýzu textu. Pokud máte text k dispozici v elektronické podobě, využijete s výhodou pomůcku na webu [4]. Písmena s nejvyšší četností jsou pravděpodobně E, A a O tedy samohlásky. Další dvě pravděpodobně budou I, N a s určitým odstupem by pak měla následovat další písmena. Čím je text kratší, tím je pravděpodobnější, že dojde k změnám pořadí písmen oproti výše uvedenému. Proto si pomáháme dalšími jazykovými znalostmi. Souhlásky a samohlásky se v českém textu poměrně

pravidelně střídají. Skupina několika po sobě jdoucích samohlásek je přitom mnohem vzácnější než skupina po sobě jdoucích souhlásek. Kromě frekvence jednotlivých písmen se můžeme zabývat frekvencí dvojic nebo trojic po sobě jdoucích písmen, kterým říkáme *bigramy*, resp. *trigramy*.

Pokud budeme zkoumat všechny bigramy, je v českém textu nejčastějších [3] těchto patnáct: ST, NI, PO, OV, CH, RO, EN, NA, JE, PR, TE, LE, KO, NE, OD. Jinou možností je sledovat pouze souhláskové bigramy ve chvíli, kdy se nám podaří odlišit souhlásky od samohlásek. Pavel Vondruška [6] jich uvádí šest: ST, PR, CH, SK, DN a TR. Navíc ovšem uvádí také typické vlastnosti (zvláštnosti) pro tři nejčastější:

ST	• Písmena S a T mají přibližně stejnou frekvenci.
	• Existuje i opačný bigram TS.
	• Je součástí množství souhláskových trigramů (STR, STN, STL, STV, ...)
PR	• Písmeno P má přibližně poloviční frekvenci než písmeno R.
	• Obrácený bigram RP se téměř nevyskytuje (výjimka: chrpa).
	• Většinou jej nelze rozšířit na souhláskový trigram doprava (výjimka: PRV), ale často jej lze rozšířit směrem vlevo (SPR, ZPR, ...)
CH	• Písmeno H má jen o něco menší frekvenci než písmeno C.
	• Často je na konci slov spolu s Y, I, A, E (YCH, ICH, ACH, ECH).
	• Většinou platí: je-li před CH souhláska, následuje po CH samohláska a naopak (OBCHOD, NECHTĚL, ...)

Nejčastějšími trigramy jsou PRO, OVA, OST, STA, PRI, PRE, TER, ENI, POD, KTE, PRA, EHO, STL, RED, KON. Nejčastějším souhláskovým trigramem je STR.

Kromě frekvence písmen a zkoumání bigramů a trigramů nám při luštění může pomoci odhad některých slov otevřeného textu, zejména když od začátku tušíme, nebo během luštění začneme tušit, jakému tématu se šifrový text věnuje.

Pokud bychom při frekvenční analýze zjistili, že nejen v otevřeném, ale i v šifrovaném textu je nejčastějším písmenem E, dalšími A, I, O atd.

a v textu se podezřele často vyskytují i skupiny po sobě jdoucích souhlásek, pak se vůbec nejedná o substituční šifru, ale o zašifrování vzniklé změnou pořadí písmen tvořících zprávu, tedy o tzv. transpozici. Kdyby všechny znaky šifrového textu měly podobnou frekvenci a nevyskytla by se výrazná maxima a minima, jednalo by se nejspíš o polyalfabetickou šifru. V těchto případech bychom museli zvolit jiné postupy luštění než výše uvedené.

Nakonec si ukažme celý postup luštění šifrového textu zašifrovaného pomocí jednoduché záměny na příkladu. Máme rozluštit šifrový text:

```
HCNON WBTOT XKNHA TRWLC OTATB HTEOM OLIXN EMTWS OENHN ETIRW
SATBH MZXKM OXTYM RCGOY TZVET AMYMW OEHNH YMWOR TVSCM HSENN
YTRWL COMXS YSCMI TSENC RGBNZ MBOSG ESHYN OEATY MWTGX NWATI
TWSAT INYLZ VOHMC TYGGX SYMEM ZRLZV JNWSR SAZTG ANJSC TAORL
ZVHGW ZGJLH NHTEO MASGW SOEXS ZVLJA TCWNH LRSAN YMHMZ WSJKT
VSATC OXNEA TVSIL OYMIO MXKSE SWSZT YNHNC ATCTA TBHMT BTZYS
HTRGA NXKSO XTZVR WLCOT ANWZV ATXKS ATBNR SGGYV TZVEM YSGHT
ZNENR OTATX KMISO NICGO YTZVE GBTES VYTOT HXKHA MKNWT ELRNH
LCANI ALZVG IYVZG VATWX SESIH ONRMJ NWNEN YGBTO XKNHA TCTZY
SHTRN SJSVN ZGBMN CGOYT ZVEGB MATHL OYTWR LJNWN AMNYT OANVN
ESNSA SXSZV SXMES ANXKS WGREM HAMNK TZTXE MHAMW GZVSH AMZMA
ASOEX KSESJ LATJL YSXXM YMOAN IMOET ZVEME XSOGC SHNEV SWASE
GENRS HTVSE NYIGW GXSWY TBTVS MAETY TREGN YAMZV XYSWG
```

Frekvenční analýzou zjistíme, že zpráva má 645 znaků. Z nich se nejčastěji vyskytuje znak T, který je ve zprávě 67krát, tedy má relativní četnost 10,4 %. Dalšími v pořadí jsou znaky S, N, M a A s relativními četnostmi 8,8 %, 8,1 %, 6,5 % a 6,5 %. Zkusme tyto znaky nahradit podle dříve uvedených tabulek po řadě písmeny E, A, O, I a N a zbývající znaky zatím nahradíme pomlčkou. Vidíme, že střídání samohlásek se souhláskami odpovídá, takže zkusíme vytipovat další znaky pro samohlásky (U a Y) a označíme je hvězdičkou a křížkem.

```
--O-O -E-E- --O-N E--+ -ENE--E--I +---O -IE-A --O-O -E---
ANE-- I---I --E-I --*-- E---E NI-I- ---OO -I--- E-A-I -A-OO
-E--+ --I-A -A-I- EA-O- *-O- I--A* -A--O --NE- I-E*- O-NE-
E-ANE -O++ ---I- E***- A-I-I +--- -O-A- AN-E* NO-A- EN---
---*- *-+- O-E-- INA*- A---A ---N E--O +--ANO -I-I- -A--E
-ANE- --O-N E-A++ --I-- I--A- A-A-E -O-O- NE-EN E--I- -E--A
```

```

-E-*N O--A- -E--- --E- NO--- NE--A NE-O- A***- E---I -A*-E
-O-O- -ENE- -I-A- O--*- -E--- *-E-A --E-E ---N I-O-E --O-
+-NO- N+---* -E--- -NE-- A-A-- -O-I- O-O-E *-E- --O-N E-E--
A-E-O A-A-O *-IO ---E ---* INE+ --E-- +-O-O NIO-E -NO-O
-AOAN A-A-- A-I-A NO--A ----I -NIO- E-E-- I-NI- ***A- NI-IN
NA--- -A-A- +NE++ -A--I -I-NO -I--E ---I- -A-* A-O-- A-NA-
*-O-A -E-A- O--*- *-A-- E-E-A IN-E- E--*O -NI-- --A-*

```

Nyní se přeorientujeme ze samohlásek na souhlásky, přesněji řečeno na souhláskové bigramy. Protože nás nyní nezajímají samohlásky, nahradíme je v zašifrovaném textu tečkami a budeme si všímat skupin několika po sobě jdoucích souhlásek.

```

HC.O. W.B.O XK.HA .RW.C O.A.B H.EO. O.IX. E..W. OE.H. E.IRW
.A.BH .ZXK. OX.Y. RC.OY .ZVE. A.Y.W OEH.. Y.WOR .V.C. H.E..
Y.RW. CO.X. Y.C.I .E.C R.B.Z .BO.. E.HY. OEA.Y .W..X .WA.I
.W.A. I.Y.Z VOH.C .Y..X .Y.E. ZR.ZV J.W.R .AZ.. A.J.C .AOR.
ZVH.W Z.J.H .H.EO .A..W .OEX. ZV.JA .CW.H .R.A. Y.H.Z W.JK.
V.A.C OX.EA .V.I. OY.IO .XK.E .W.Z. Y.H.C A.C.A .BH.Z B.ZY.
H.R.A .XK.O X.ZVR W.CO. A.WZV A.XK. A.B.R ..OY .ZVE. Y..H.
Z.E.R O.A.X K.I.O .IC.O Y.ZVE .B.E. VY.O. HXKHA .K.W. E.R.H
.CA.I A.ZV. I.YZ. VA.WX .E.IH O.R.J .W.E. Y.B.O XK.HA .C.ZY
.H.R. .J.V. Z.B.. C.OY. ZVE.B ...H. OY.WR .J.W. A..Y. OA.V.
E...A .X.ZV .X.E. A.XK. W.RE. HA..K .Z.XE .HA.W .ZV.H A.Z.A
A.OEX K.E.J .A.J. Y.XK. Y.OA. I.OE. ZVE.E X.O.C .H.EV .WA.E
.E.R. H.V.E .YI.W .X.WY .B.V. .AE.Y .RE.. YA.ZV XY.W.

```

Projdeme-li uvedený text, např. pomocí funkce „Najít“ textového editoru (klávesová zkratka Ctrl + F), zjistíme že obsahuje bigram ZV 15krát, přičemž frekvence znaku Z 4,7 % je o něco větší než frekvence znaku V 3,9 %. Tento bigram je navíc zpravidla obklopen z jedné strany samohláskou a z druhé strany souhláskou. Proto usuzujeme, že jde v otevřeném textu o CH. Druhým nejčastějším bigramem je XK, který je zde 11krát, přitom text neobsahuje opačný bigram KX. Proto usuzujeme, že jde v otevřeném textu o bigram PR. Třetím v pořadí je bigram OE, který najdeme 6x, ale také je zde 2x opačný bigram EO. Oba znaky mají zhruba stejnou frekvenci. Proto usuzujeme, že jde v otevřeném textu o bigram ST. Dalším bigramem je HA, o kterém již víme, že znak A představuje v otevře-

ném textu N, a protože se mezi šesti nejčastějšími českými souhláskovými bigramy vyskytuje DN a frekvence znaků tomu také odpovídají, usuzujeme, že se jedná právě o DN. Nyní už známe část převodové tabulky:

A	B	C	E	G	H	I	J	K	L	M	N	O	R	S	T	V	W	X	Y	Z
N	-	-	T	-	D	-	-	R	-	I	O	S	-	A	E	H	-	P	-	C

První řádek tabulky neobsahuje znak D, F, P, Q a U, protože se v šifrovaném textu nevyskytují. Ve druhém řádku jsou odpovídající písmena otevřeného textu, která známe. Zatím neznámá jsou nahrazena pomlčkou. Po provedení této záměny získáme ještě stále obtížně čitelný text, ze kterého vystupuje na začátku sedmého řádku skupina DE–N OPRAS PECH, což by mohlo znamenat NA PROSPĚCH, a na začátku jedenáctého řádku je skupina TAOAN APACH API, což by mohlo znamenat TO A ONO POCHOPÍ. Oba převody ovšem vyžadují vzájemnou výměnu písmen A a O ve druhém řádku tabulky. To znamená, že v daném otevřeném textu má písmeno O vyšší četnost než písmeno A. Podobnou úvahou opravíme ještě jeden chybný odhad. Souhláskový bigram HA nenahrazuje v otevřeném textu DN, nýbrž VN. Po úpravách dostaneme převodovou tabulku:

A	B	C	E	G	H	I	J	K	L	M	N	O	R	S	T	V	W	X	Y	Z
N	-	-	T	-	V	-	-	R	-	I	A	S	-	O	E	H	-	P	-	C

Když si tabulku pozorně prohlédneme, vidíme, že zobrazení, které představuje na množině písmen abecedy, je samo k sobě inverzní. Tedy: když A nahrazuje N, pak N nahrazuje A, když místo E píšeme T, pak místo T píšeme E atd. Doplníme-li tabulku použitím tohoto pravidla, budeme mít k dispozici další informace:

A	B	C	E	G	H	I	J	K	L	M	N	O	P	R	S	T	V	W	X	Y	Z
N	J	Z	T	-	V	M	-	R	-	I	A	S	X	K	O	E	H	D	P	-	C

Pomocí této poslední tabulky získáme sice ještě neúplný, ale již poměrně dobře čitelný převod do otevřeného textu:

VZASA -E-ES PRAVN EK--Z SENE- VETSI S-MPA TIE-O STAVA TEMK-
ONE-V ICPRI SPE-I KZ-S- ECHTE NI-I- STVAA -I-SK EHOZI VOTAA
-EK-- ZSIPO -OZIM EOTAZ K--AC I-SO- TOV-A STNE- I-E-P A-NEM
E-ONE MA--C HSVIZ E---P O-ITI CK-CH -A-OK ONCE- NA-OZ ENSK-
CHV-- C---V AVETS INO-- OSTPO CH--N EZ-AV -KONA -IVIC -O-RE
HONEZ SPATN EHOM- S-IMS IPROT O-OCE -AVAZ NEZEN E-VIC -EC-O
VEK-N APROS PECHK --ZSE NA-CH NEPRO NE-AK O--S- ECHTI -O-VE
CATAK SENEP RIMOS AMZ-S -ECHT --ETO H-ESE VPRVN IRA-E T-KAV
-ZNAM N-CH- ME-C- HNE-P OTOMV SAKI- A-ATE ---ES PRAVN EZEC-
OVEKA O-OHA C--IA Z-S-E CHT-- INEV- S-E-K --A-A NIA-E SNAHA
TOAON OPOCH OPITO NAPRO --KTI VNIAR ECEPT IVNI- -CHOV NICIN
NOSTP ROTO- -NE-- -OPRI -ISNA MISTE CHTIT POS-Z OVATH O-NOT
-TAKO VEHOT A-M-- -PO-- E-EHO INTE- EKT-A -NICH P-O--

Nyní zbývá odhadnout zbývající písmena otevřeného textu, doplnit převodovou tabulku, dokončit převod celého šifrovaného textu na otevřený, zrušit členění do pětímístných skupin a nahradit jej členěním do slov a doplnit diakritiku (čárky a háčky). Tyto poslední kroky již nejsou nijak obtížné, proto ponecháváme čtenáři, aby sám dokončil luštění daného šifrovaného textu. Odměnou mu budou moudrá slova Alberta Einsteina [1].

Literatura

- [1] *Einstein, A.*: Jak vidím svět. Praha: Nakladatelství Lidové noviny, s. r. o. 1993.
- [2] *Klíma V.*: Utajené komunikace – 2. díl. CHIP: magazín informačních technologií, červen 1994, roč. 4, č. 6.
- [3] *Kolektiv*: Frekvence písmen, bigramů, trigramů, délka slov [online]. Centrum zpracování přirozeného jazyka Fakulty informatiky Masarykovy univerzity, Brno. [cit. 2010-05-05] Dostupné z:
http://nlp.fi.muni.cz/cs/Frekvence_pismen_bigramu_trigramu_delka_slov
- [4] *Musílek M.*: Šifry a kódy [online]. 2010 [cit. 2010-05-05] Dostupné z:
<http://www.musilek.eu/michal/sifry.html?menu=mat>
- [5] *Singh S.*: Kniha kódů a šifer. 2. vyd. Praha: Argo a Dokořán, 2009.
- [6] *Vondruška P.*: Kryptologie, šifrování a tajná písma. 1. vyd. Praha: Albatros, 2006.

Autorkou úvodní ilustrace je Mgr. Jaroslava Čermáková.

Strukturované programování ve VBA pro Excel

ŠTĚPÁN HUBÁLOVSKÝ

Přírodovědecká fakulta Univerzity Hradec Králové

Aplikace MS Excel lze s výhodou využívat jako nástroje, který umožňuje propojit tradiční metodu výuky s možnostmi, které nabízejí informační technologie, viz také např. [1] a [2]. Článek odráží zkušenosti z výuky programování na střední a vysoké škole a zabývá se možnostmi využití MS Excel ve výuce algoritmizace a programování na střední škole.

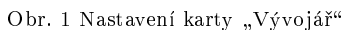
MS Office jako programovací prostředí

Všechny MS Office produkty jako je Excel, Access, Word, PowerPoint, Outlook a další, mají implementován programovací jazyk Visual Basic for Application (VBA). Jedná se o objektově orientovaný programovací jazyk, jehož základem je Visual Basic [3]. VBA neobsahuje takové množství objektů, jako vlastní Visual Basic, umí však využívat a manipulovat s objekty „své“ mateřské aplikace (buňkami, výběry buněk, listy atd. u Excelu, odstavci, číslováním a formátováním atd. u Wordu apod.), což lze považovat za jeho hlavní výhodu, především ve výuce programování. Každý uživatel MS Office sady má k dispozici programovací jazyk, který si může dle libosti vyzkoušet a vytvořit svoji „malou užitečnou aplikaci“, která se může stát základem pro jeho další zájem o programování.

Základy algoritmizace začínáme žákům a studentům vysvětlovat na tvorbě algoritmů pro řešení jednoduchých matematických úloh. Vzhledem k tomu, že MS Excel je aplikace, která sama o sobě umožňuje provádět matematické úkony, jedná se o tabulkový kalkulátor, i z hlediska programování proto považují za nejvhodnější začít s integrovaným programovacím jazykem Visual Basic pro Excel. Jak uvidíme v dalším textu, tento programovací jazyk, přestože se jedná o objektově orientovaný jazyk, umožní zpracovat základní algoritmické úlohy bez toho, aby bylo nutné se již v začátcích výuky zmiňovat o objektech, vlastnostech a metodách objektů a o událostmi řízeném programování. Naopak můžeme se s výhodou při výuce držet plně strukturovaného, sekvenčního přístupu, což je zejména vhodné v počátečních fázích výuky algoritmizace a programování.

Dříve než ukážeme příkazy na konkrétním příkladě v jazyce VBA, je třeba si stručně popsat vývojové prostředí. Jak již bylo řečeno, VBA je součástí každé MS Office aplikace.

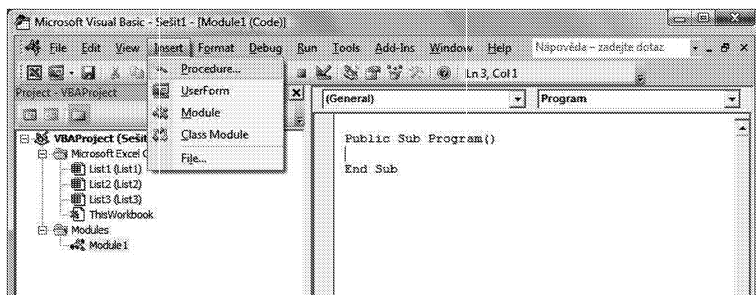
Vývojové prostředí je dostupné v Excelu 2007 z karty „Vývojář“. Tato karta není standardně zobrazena. Kliknutím na Tlačítko Office (kruhová ikona v levém horním rohu aplikace Excel) a poté na tlačítko Možnosti aplikace Excel otevřeme dialogové okno Možnosti aplikace Excel. Zde zaškrtneme políčko „Zobrazit na pásu karet Vývojář“ (obr. 1).



Aby bylo možné toto prostředí využívat, je třeba dále nastavit zabezpečení Maker na nejnížší úroveň. V Excelu 2007 se toto nastavení provádí na kartě Vývojář – Zabezpečení Maker, kde v nastavení maker zaškrtneme políčko „Povolit všechna makra“.

Vývojové prostředí lze otevřít dvěma způsoby – buď na kartě Vývojář klikneme na ikonu „Visual Basic“ nebo použijeme klávesovou zkratku Alt-F11. Otevře se nové okno. V levé části jsou dvě podokna – Project (zobrazuje seznam aktuálně otevřených projektů – sešitů, listů, modulů, popř. formulářů) a Properties (zobrazuje jejich vlastnosti). V začátcích programování můžeme „Properties Window“ křížkem zavřít, používá se hlavně při tvorbě vlastních formulářů.

Kód jazyka VBA se píše a je uložen do tzv. modulů. Modul vložíme pomocí nabídky „Insert – Module“. Objeví se bílá plocha, na kterou již píšeme svůj kód jazyka VBA (obr. 2).



Obr. 2 Vývojové prostředí

Každý program má začátek a konec. VBA umožňuje vložit začátek a konec opět pomocí nabídky „Insert – Procedure“. Do dialogového okna zadáme Název programu a stiskneme OK. Na bílé ploše se objeví `Public Sub Název()`, což značí začátek našeho programu a uzavírací řádek `End Sub` jako konec programu. Název programu musí být jednoznačný, bez mezer. Na jedné pracovní ploše můžeme mít napsáno více programů, přehlednější je však další program psát do nového modulu.

Program zapsaný ve VBA – rekurentní výpočet odmocniny

V následujícím textu ukážeme, jak VBA můžeme vhodně využít pro výuku základů algoritmizace a programování. Dle vlastní zkušenosti doporučuji výuku programování v počáteční fázi a výuku algoritmizace vzájemně prolínat, což znamená programovat dle zásad strukturovaného programování, kdy algoritmus řešení úlohy zapsaný pomocí vývojového diagramu přepisujeme pomocí příkazů programovacího jazyka. Na konkrétním příkladu výpočtu druhé odmocniny čísla N ($X = \sqrt{N}$) pomocí rekurentního vztahu ukážeme základní programovací prvky a syntaxi jazyka VBA.

Rekurentní vztah pro výpočet odmocniny je odvozen na základě Newtonovy metody sečen pro řešení rovnice $X^2 - A = 0$ například v [4] a má tvar

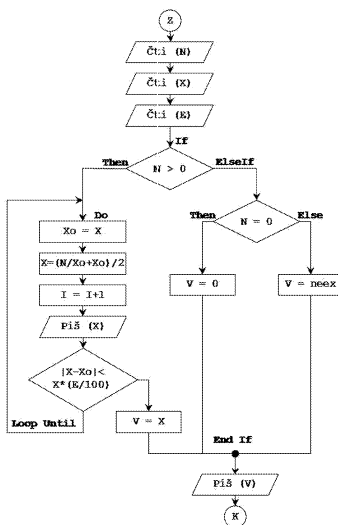
$$X_{i+1} = \frac{1}{2} \left(\frac{N}{X_i} + X_i \right),$$

kde X_0 je počáteční odhad a X_i ($i = 1, 2, 3, \dots$) jsou členy posloupnosti iterací, které se postupně přibližují k výsledné hodnotě. Výpočet bude

ukončen, pokud dosáhneme uživatelem zadané procentuální přesnosti výpočtu E , což lze vyjádřit podmínkou

$$|X_i - X_{i+1}| < X_i \cdot E/100.$$

Z hlediska algoritmizace to znamená, že rekurentní výpočet se opakuje tak dlouho, dokud není splněna výše uvedená podmínka přesnosti. Pro ukončení výpočtu v algoritmu použijeme cyklus s podmínkou na konci. Vývojový diagram algoritmu je na obrázku 3 – viz např. [5]. Algoritmus je složen z příkazů vstupu, podmíněného příkazu, příkazu cyklu s podmínkou na konci a příkazů výstupu. (Přepis vývojového diagramu pomocí příkazů jazyka VBA a jejich specifika probereme v dalším textu.)



Obr. 3 Výpočet odmocniny – vývojový diagram

```

Public Sub Odmocnina()
N = InputBox("Zadej N:")
X = InputBox("Zadej X0:", , 1)
E = InputBox("Zadej E:", , 0.001)
Cells(2, 1) = N
If N > 0 Then
    I = 0

```

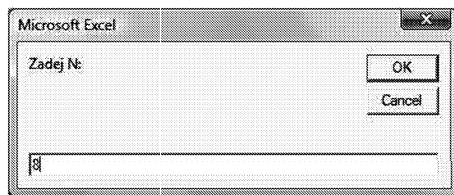
```

Do
    Xo = X
    X = (N / Xo + Xo) / 2
    I = I + 1
    Cells(I+1, 2) = Xo
Loop Until (Abs(X - Xo) < X *(E /100))
V = X
ElseIf N = 0 Then
    V = 0
Else
    V = "neex"
End If
Cells(I + 2, 2) = V
MsgBox ("X = " & V)
End Sub

```

Vstup dat

Vstupní hodnoty N , X , E načítáme pomocí příkazu InputBox. Příkazový řádek v programu $X = \text{InputBox}(\text{"Zadej } X\text{:"}, , 1)$ otevře jednoduché dialogové okno s hlášením, které vyžaduje zadání dané hodnoty proměnné, a čeká na okamžitou akci uživatele (obr. 4). S výhodou můžeme použít jednoho z parametrů InputBox (1 před poslední závorkou), který umožňuje nastavit počáteční hodnotu proměnné. Vstup hodnot lze realizovat i z jednotlivých buněk Excelu příkazem $X = \text{Cells}(R,S)$, který do proměnné X načte hodnotu z buňky na řádku R a ve sloupci S .



Obr. 4 Vstup hodnot pomocí příkazu InputBox

Podmíněný příkaz

Po přečtení vstupních hodnot následuje vícenásobný podmíněný příkaz If - Then - ElseIf - Else - End If, který větvi program podle

hodnoty zadaného N . Ve větvi pro $N > 0$ provádíme rekurentní výpočet odmocniny, ve větvi pro $N = 0$ dochází s ohledem na zaokrouhlovací chyby po několika krocích k dělení 0, proto výsledek do proměnné $V = 0$ přiřadíme okamžitě a pro $N < 0$ do výsledné proměnné V vložíme řetězec $V = „neex“$. V jazyce VBA jsou složené příkazy, tedy i podmíněný příkaz, ukončeny tzv. uzavíracím řádkem (**End If**), který tyto příkazy ukončí. (Poznámka: U jazyka Pascal složený příkaz píšeme mezi *begin end*).

Rekurentní výpočet odmocniny

V případě, že zadaná hodnota N je větší než 0, provede se rekurentní výpočet odmocniny. Tento výpočet realizujeme pomocí cyklu s podmínkou na konci. VBA umožňuje cyklus s podmínkou na konci ukončit, pokud podmínka je splněna, ale také pokud splněna není, viz tab. 1. (Poznámka: Jazyk Pascal umožňuje u cyklu s podmínkou na konci ukončit cyklus pouze tehdy, když podmínka je splněna.). V našem případě se použil cyklus typu *until*, který končí, pokud podmínka splněna je, tedy pokud $|X - X_0| < X \cdot E / 100$, kde X značí nově spočtenou a X_0 předešlou hodnotu odmocniny, E je přesnost v procentech. (Úplný přehled syntaxe cyklů s podmínkou jazyka VBA je v tab. 1).

Tab. 1 Přehled cyklů s podmínkou

Cykly s podmínkou na konci	
Typ until	Typ while
Do <i>Příkazy cyklu</i> Loop Until Podmínka	Do <i>Příkazy cyklu</i> Loop While Podmínka
Cykly s podmínkou na začátku	
Typ Until	Typ While
Do Until Podmínka <i>Příkazy cyklu</i> Loop	Do While Podmínka <i>Příkazy cyklu</i> Loop

Kromě cyklů s podmínkou umožňuje VBA i cykly řízené proměnnou – cykly typu *For*. Syntaxe tohoto příkazu je:

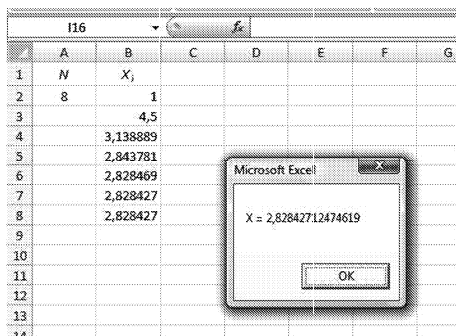
```
For I = P To N Step S
    Příkazy cyklu
Next I
```

Zde I je řídicí proměnná cyklu, která nabývá hodnot v mezích od P do N s krokem S . Cyklus, jak je to ve VBA standardem, je ukončen koncovým řádkem *Next*.

Výstupy hodnot

Konečnou vypočtenou hodnotu odmocniny zobrazíme příkazem výstupu MsgBox ("X = " & V).

Tento příkaz vypíše obsah proměnné V i s doplňkovým textem ($X = \dots$) – viz obr. 5.



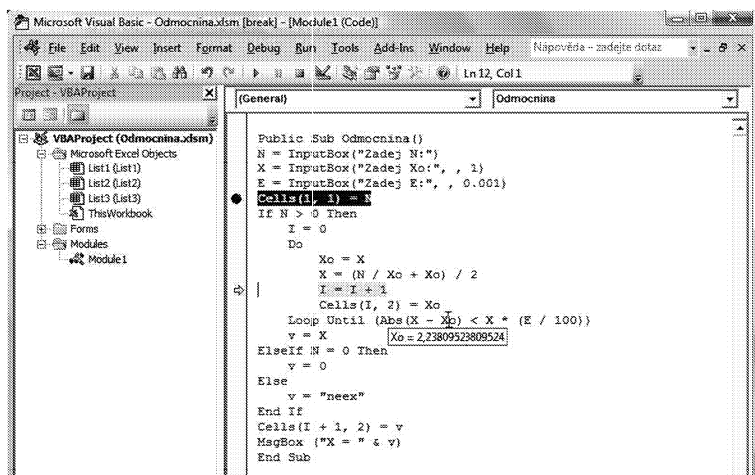
Obr. 5 Výpis hodnot

V případě zadání $N < 0$ se proměnná V naplní informačním textem „neex“, což značí, že odmocninu nelze vypočítat. Je tedy zřejmé, že do jedné proměnné umožňuje VBA zapisovat jak číselnou hodnotu, tak hodnotu typu text. Z hlediska deklarací tento programovací jazyk není striktní. Pokud není na začátku programu proměnná deklarována, má standardně přiřazen typ *Variant*, který umožňuje do této proměnné načíst jakékoliv číslo (celé, reálné), text a všechny další typy, které VBA připouští. Změna typu proměnné (textu na číslo apod.) probíhá „automaticky“, kompilátor ve většině případů hodnotě přiřadí správný typ. Pro lepší pochopení typu *Variant* si lze představit, že proměnná tohoto typu ve VBA odpovídá v Excelu typu buňky Obecný.

V našem příkladu program vypisuje (pro informaci) do buněk sloupce B průběžné hodnoty X . Tyto výstupy provádí příkaz `Cells(I+1, 2) = X0`, přičemž I určuje index řádku a číslo 2 určuje index sloupce – viz obr. 5.

Odladění programu a jeho spuštění

Program se spouští klávesou F5 nebo zelenou šipkou v nabídce. Velice užitečnou funkcí, zejména pro ladění, je možnost tzv. krokování programu. Program krokujeme stiskem klávesy F8 a každým dalším stiskem této klávesy se posuneme na další řádek programu (aktuálně prováděný řádek je označen žlutě). Zastavit program v průběhu ladění můžeme také pomocí stop značky umístěné vlevo vedle řádku, na kterém chceme program pozastavit. Pokračovat dál umožní stisk klávesy F8 nebo F5. VBA v režimu krokování umožňuje zjistit aktuální hodnotu proměnné – stačí umístit kurzor na tuto proměnnou a hodnota se objeví v malém rámečku – viz obr. 6.



Obr. 6 Krokování programu ve VBA

Program napsaný ve VBA se neukládá samostatně, ale vždy společně se sešitem Excelu v rámci něhož je vytvořen. Ve verzi Excelu 2007 je třeba aplikaci uložit jako Sešit aplikace Excel s podporou maker (*.xlsm).

Závěr

Článek představuje Visual Basic for Application (VBA) jako programovací jazyk vhodný pro výuku základů algoritmizace a programování. Přestože VBA patří mezi moderní objektově orientované a událostmi řízené programovací jazyky, lze ho využít i v plně sekvenčním strukturovaném

programování aniž by bylo nutné zabývat se otázkami objektů a událostmi. Tento přístup je v základech výuky programování nezastupitelný. Další výhodou tohoto programovacího jazyka je to, že se jedná o jazyk implementovaný do všech MS Office produktů, je v těchto aplikacích dostupný a plně využitelný. Navíc si s využitím tohoto jazyka můžeme při využívání těchto aplikací usnadnit práci, pokud si naprogramujeme nástroj, který za nás vykonává určitou častokrát se opakující stejnou činnost.

Literatura

- [1] *Pražák, P. – Pražáková, B.*: Excel a rekurentně zadané posloupnosti ve finanční matematice. MFI, r. 18 (2008/9), č. 1, s. 42–51.
- [2] *Hubálovský, Š.*: Využití aplikace MS Excel pro výpočet těžiště soustavy kvádrů. MFI, r. 19 (2009/10), č. 9, s. 551 – 558.
- [3] *Walkenbach, J.*: Microsoft Excel, Programování ve VBA. Computer Press, Brno, 2004.
- [4] *Redakce MFI*: Odmocnina. MFI, r. 18 (2008/9), č. 1, s. 52 – 53.
- [5] *Taufer, I. – Kotyl, J. – Hrubina, K.*: Algoritmy a algoritmizace – vývojové diagramy. Univerzita Pardubice, 2009.

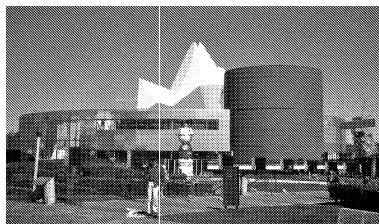
ZKUŠENOSTI

Kam chodí žáci a studenti v Toulouse na exkurze

Některá města se pyšní akvaparky či různými pouťovými atrakcemi s nepříliš hodnotnou zábavou a stávají se přitom – bohužel – i cílem školních výletů. Se skupinou přírodovědně vzdělaných učitelů z několika zemí EU jsem měl možnost ve francouzském městě Toulouse poznat podstatně hodnotnější zábavu. Velmi přitažlivou a hojně navštěvovanou lokalitou je zde astronomicky a technicky zaměřené vzdělávací centrum. Rozlehlý areál s nápisem *Cité de l'Espace* nad vchodem je zaplněný expozicemi, které návštěvníkovi přibližují

vše, co souvisí se vstupem člověka do kosmu (obráz. 1).

Je to obdoba proslulého *Space & Rocket Center* v americkém Huntsvillu. Zatímco Američané jsou velmi pyšní na vlastní podíl dobývání vesmíru (v tamním *Muzeu času* jde téměř o ryze americký obsah expozice), Francouzi jsou zde objektivnější. Je pravdou, že první družici, pejska i člověka poslali do kosmu Rusové, a právě těmto událostem je věnována



Obr. 1