

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**OPENXML DOKUMENT MS EXCELU 2010 A JEHO
VNITŘNÍ STRUKTURA**
BAKALÁŘSKÁ PRÁCE

Aneta Bauerová

Přírodovědná studia, obor Informatika se zaměřením na vzdělávání

Vedoucí práce: Mgr. Tomáš Jakeš

Přeštice, 2014

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně s použitím uvedené literatury a zdrojů informací.

V Přešticích, 15. duben 2014

.....
vlastnoruční podpis

RÁDA BYCH PODĚKOVALA SVÉMU VEDOUCÍMU PRÁCE, PANU
MGR. TOMÁŠI JAKEŠOVI, ZA ČAS, KTERÝ MI VĚNOVAL PŘI
ŘEŠENÍ RŮZNÝCH SLOŽITÝCH PROBLÉMŮ, KTERÉ PSANÍ MÉ
BAKALÁŘSKÉ PRÁCE DOPROVÁZELY.

ZDE SE NACHÁZÍ ORIGINAL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

Úvod	2
1 DŮVODY VEDOUcí KE VZNIKU XML	3
1.1 DŮVODY VZNIKU OPEN XML DOKUMENTU MS EXCELU	4
2 ZÁKLADY XML	5
2.1 ZÁKLADNí PODMíNKY PRO ZÁPIS XML DOKUMENTU.....	5
2.1.1 Xml dokument musí obsahovat alespoň jeden element	5
2.1.2 Pravidla pro jména XML elementů	5
2.1.3 Kořenový Element	6
2.1.4 Velká a malá písmena.....	6
2.1.5 Elementy v XML se nesmí Křížit.....	7
2.1.6 XML atributy a jejich pravidla.....	8
2.1.7 Zakázané znaky v XML.....	9
3 OPENXML DOKUMENT MS EXCELU 2010 A JEHO VNITRNí STRUKTURA.....	10
3.1 VNITRNí STRUKTURA A ORGANIZACE OPENXML DOKUMENTU MS EXCELU 2010	10
3.1.1 Obsah souboru zip.....	10
3.2 VÝZNAMNÉ SOUBORY A JICH ELEMENTY	14
3.2.1 Vnitřní struktura souboru [Content_Types].xml	14
3.2.2 Vnitřní struktura souboru App.xml.....	15
3.2.3 Vnitřní struktura souboru docProps\core.xml	17
3.2.4 Vnitřní struktura souboru Workbook.xml.rels.....	18
4 EXCEL 2010 V PŘÍKLADECH	20
4.1 ULOŽENí DATOVÉ HODNOTY DO BUŇKY.....	20
4.2 ZJIŠTĚNí TYPU DAT	22
4.2.1 Typ obecný	22
4.2.2 Typ zlomky.....	23
4.2.3 Typ datum	25
4.2.4 Typ čas	27
4.2.5 Typ číslo.....	29
4.2.6 Typ matematický.....	30
4.2.7 Typ měna.....	30
4.2.8 Typ procenta.....	31
4.2.9 Typ speciální	32
4.2.10 Typ text.....	33
4.3 SLUČOVÁNí BUNĚK	36
4.3.1 Sloučení dvou buněk	36
4.3.2 Sloučení tří buněk.....	37
4.3.3 Dvojitě sloučení	37
4.4 VZORCE.....	38
4.5 FUNKCE.....	40
4.6 PODMíNĚNÉ FORMÁTOVÁNí	42
4.7 TABULKA	45
4.8 GRAFY	47
ZÁVĚR	51
RESUMÉ	52
POUŽITÁ LITERATURA	53
SEZNAM OBRÁZKŮ	55
SEZNAM TABULEK	56
PŘÍLOHY	I

Úvod

Svět informačních a komunikačních technologií prochází neustálým a rychlým vývojem, i přesto však vývojáři nezapomínají, že dnes uložená data bude potřeba otevírat i v budoucnu. Za tímto účelem vznikají různé standardy, jež mají za úkol poskytnout pravidla pro vytváření nových programů, které by měli mít i pro další generace pochopitelnou strukturu.

Tato práce se zabývá jedním z těchto standardů, a to standardem Office OpenXML dokumentu MS Excelu 2010, využívající značkovací jazyk XML.

Jazyk XML je vhodný především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, ovšem za předpokladu, že jazyk popisuje jeho strukturu z hlediska věcného obsahu a nezabývá se vzhledem.

V první kapitole si nastíníme důvody vzniku standardu Office OpenXML pro MS Excel 2010, tak abychom si mohli uvědomit jeho potřebu.

V kapitole Základy XML si představíme základní pravidla pro zápis XML kódů, protože jazyk můžeme používat, pouze za předpokladu, že známe základní pravidla pro jeho zápis, a proto si tyto pravidla rozebereme.

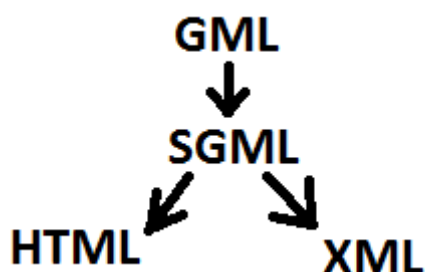
Kapitola OpenXML dokument a jeho vnitřní struktura nám pomůže nahlédnout do způsobu řazení a organizace v souboru zip a popíše nám základní soubory v něm obsažené.

Po přečtení kapitoli Excel 2010 v příkladech, pochopíte, jak pracuje Excel 2010 s XML kódem.

Jedním z důvodů vzniku této bakalářské práce byl nedostek česky psaných materiálů a jejich kvalita.

1 DŮVODY VEDOUcí KE VZNIKU XML

Vše odstartovala v šedesátých letech firma IBM, která se rozhodla vyřešit problémy s ukládáním velkého množství právních dokumentů. Jejím snahou bylo nejen uložit data na delší dobu, ale i to, aby bylo takto uložená data možné otevřít i v jiných programech. Často se totiž může přihodit, že data uložená před delší dobou v určitém programu nemusí být schopné otevřít novější programy, což je v právním oboru problém, kvůli otevření starších či odložených případů. Dalším důvodem k tomuto činu byl neustálý konkurenční boj, díky němuž vznikly různé programy, které ale navzájem nebyly kompatibilní, a to zejména díky různému způsobu ukládání dat. A tak vznikl v 60. letech obecný značkovací jazyk GeneralizedMarkupLanguage¹, který již zvládal hierarchickou strukturu, ale bohužel byl příliš obecný na to, aby našel velké uplatnění. Proto se v roce 1986 z GML vyvinul jazyk Standard GeneralizedMarkupLanguage², jenž je výstupem práce organizací ANSI a GCA, které se za účelem vytvořit vlastní značkovací jazyk v roce 1980 sloučily. Jazyk se vyznačoval svou nezávislostí na platformě a aplikacích. Využíval uložení pomocí kódu ASCII, což je znaková sada, která má podobu tabulky, obsahuje čísla, písmena anglické abecedy a vybrané znaky využívané v informatice a matematice. Avšak ani tento jazyk nenašel velké využití, neboť byl velmi složitý. I když se tento jazyk příliš neuplatnil, dal možnost vzniku nejznámějšímu a nejrozšířenějšímu jazyku HyperTextMarkupLanguage³. V roce 1998 vznikl konečně jazyk XML, který je zjednodušenou podmnožinou jazyka SGML, nikoliv HTML, jak se mnoho lidí v dnešní době domnívá. Jazyk již nebyl tak složitý jako jeho rodič, ale stále umožňuje tvorbu vlastních tagů. (1) (2) (3) (4)



Obrázek 1: Vývoj

¹ Zkráceně GML

² Zkráceně SGML

³ Zkráceně HTML

1.1 DŮVODY VZNIKU OPEN XML DOKUMENTU MS EXCELU

V roce 2007 vznikl kancelářský balík Microsoft Office 2007, který již podporuje otevřený formát. Stále však docházelo k problémům s otevřením některých souborů vytvořených ve starší verzi a naopak, což bylo vylazeno v Microsoft Office 2010. Další příčinou byla snaha o to, aby data byla stálá, tzn., že data, která byla vytvořena dnes, bude možné otevřít pomocí jiných technologií, které se v informatickém oboru stále vyvíjí. Dalším důvodem vzniku je pak schopnost otevřít částečně poškozená data (podsoubory). Představme si, že máme Excel dokument, který se skládá z několika listů a v jednom listu dojde k poškození dat, v předešlých verzích firmy Microsoft bychom tento dokument neotevřeli, protože využíval tzv. binární formáty⁴, které neměly možnost zachránit alespoň část svých dat, ale v novějších verzích 2007 a 2010 můžeme otevřít nepoškozené sešity, pokud jsou nekonzistentní. Díky kompresi lze také podstatně snížit velikost, kterou dokument zabírá v paměti.(2) Dalším důvodem je pak snaha o vytvoření stále lepších a obsáhlejších programů, které budou vyhovat nejvíce potřebám jejich uživatelů.

⁴ Ukládají soubory jako sled jedniček a nul.

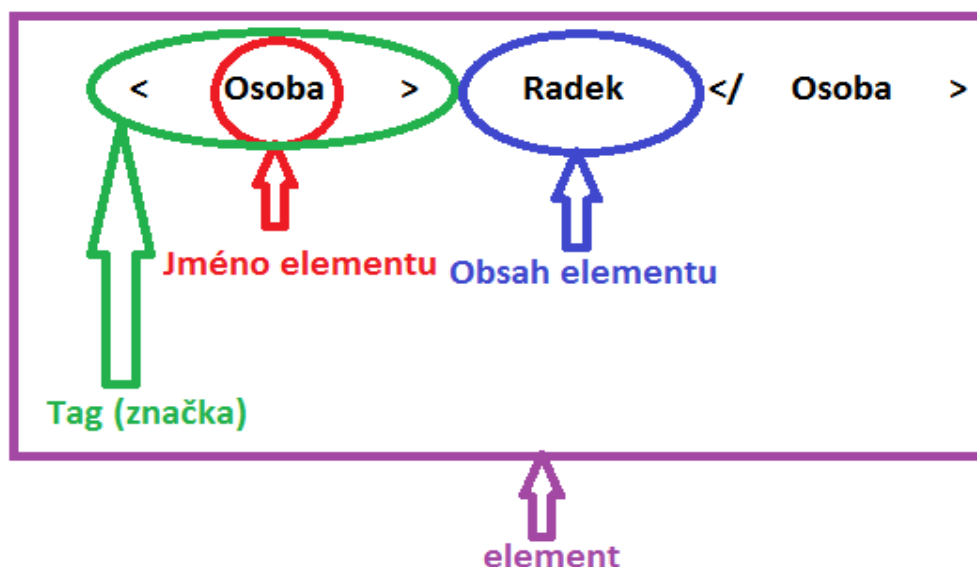
2 ZÁKLADY XML

2.1 ZÁKLADNÍ PODMÍNKY PRO ZÁPIS XML DOKUMENTU

Pokud chcete používat XML, musíte dodržovat několik pravidel.

2.1.1 XML DOKUMENT MUSÍ OBSAHOVAT ALESPŮŇ JEDEN ELEMENT

„Základním stavebním kamenem každého XML dokumentu jsou tzv. elementy. Ty jsou složeny z počáteční značky, samotného obsahu elementu a koncové značky.“ (5) Můžeme se setkat s označením značky jako anglického názvu tag. V této práci budeme používat označení tag. Každý tag se skládá z lomených závorek a jména elementu. Koncový tag se skládá opět z lomených závorek a jména elementu, ale za první lomenou závorkou se připisuje lomítko. Mezi počáteční a koncový tag se píše obsah (text). Existují ovšem i případy, kdy lze použít element bez obsahu. Ten můžeme zapsat dvěma způsoby. Zapišeme ho buď prvním způsobem `<JmenoElementu></JmenoElementu>`, nebo druhým způsobem, který je zkrácený `<JmenoElementu/>`. (5)



Obrázek 2: Základní problematika elementu

2.1.2 PRAVIDLA PRO JMÉNA XML ELEMENTŮ

„Při zvolení jména elementů má programátor téměř volnou ruku až na pár jednoduchých pravidel, jinak ovšem není omezen. Neexistuje žádný seznam povolených či přímo definovaných tagů.“ (5) Kodér se nemusí omezovat pouze na názvy typu např. A a B, ale můžeme ho pojmenovat jakoliv, doporučuje se proto využívat jména, která co nejlépe

vystihující obsah elementů. Jméno elementu by tak mělo popisovat a specifikovat, co je jejím obsahem. Jaká jsou pravidla pro označení elementů, které se musí striktně dodržovat? První pravidlo hovoří o tom, že všechny musí začínat buď písmenem, nebo podtržítkem, druhé pak o tom, že celý název elementu smí obsahovat písmena, číslice, podtržítka, pomlčky a tečky. Jiné než tyto znaky nelze použít. (5)

2.1.3 KOŘENOVÝ ELEMENT

Jak již víme, každý správně zapsaný XML dokument musí obsahovat alespoň jeden element. Můžeme jich samozřejmě použít tolik, kolik je nám libo, ale musí být uzavřeny ve zvláštním nadřazeném elementu, tzv. kořenovém elementu, který se nesmí vyskytnout v obsahu jiných elementů. Naopak všechny jiné elementy musí být v kořenovém elementu obsaženy. Kdybychom si ho měli abstrahovat, můžeme si představit například nákupní vozík, který obsahuje rohlíky, chleba a jiné zboží. Bez něj bychom velký nákup nedonesli k pokladně. Vystupuje zde jako vyšší element neboli kořenový element. Na celý XML dokument může být použit pouze jeden kořenový element. Pro znázornění si uvedeme příklad zápisu. (5)

```
<KořenovýElement>
  <JménoElementu1>ObsahElementu</JménoElementu1>
  <JménoElementu2>ObsahElementu</JménoElementu2>
  ...
  <JménoElementu3>ObsahElementu</JménoElementu3>
</KořenovýElement> (5)
```

2.1.4 VELKÁ A MALÁ PÍSMENA

XML rozlišuje, zda při zápisu použijeme velká či malá písmena. Je ale nepodstatné a nijak nás neomezuje to, jestli v jednom elementu použijeme malé či velké písmeno či jejich kombinaci. Můžeme tedy používat malá či velká písmena dle libosti, ale musíme dodržet to, že počáteční i koncový tag mají úplně stejná jména elementu a to i s použitím malých či velkých písmen. Pro názornost si opět ukážeme příklady.

```
<Jmeno>obsah</jmeno>      //špatně
<JMENO>obsah</jmeno>     //špatně
<Jmeno>obsah</Jmeno>     //správně
<jmeno>obsah</jmeno>     //správně
```

První a druhý zápis je špatně, protože počáteční a koncový tag mají rozdílná jména. V prvním příkladu je ve jméně elementu počátečního tagu napsané velké J, kdežto v

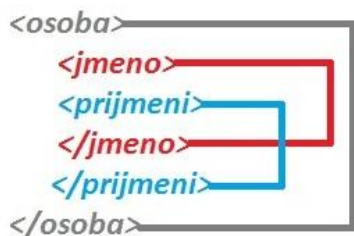
koncovém tagu je napsané malé j. V druhé ukázce jsou jména elementu počátečního a koncového tagu zcela rozdílná z pohledu malých a velkých písmen. Třetí a čtvrtý zápis je zapsán správně, protože jméno elementu koncového a počátečního tagu se zcela shoduje. Můžeme si samozřejmě uvést více příkladů, ale pro názornost výše uvedené postačí. (5)

2.1.5 ELEMENTY V XML SE NESMÍ KŘÍŽIT

„Další pravidlo, které si musíme zapamatovat, říká: pokud nějaký element obsahuje počáteční značku jiného elementu, musí pak obsahovat i příslušnou koncovou značku.“ (5)

Toto na první pohled složité pravidlo lze jednoduše vysvětlit na příkladech.

První příklad je zapsán špatně a druhý správně, abychom si vysvětlili, kde vzniká ono křížení, bude nejlepší si jednotlivé počáteční a koncové značky elementů pospojovat.



Obrázek 3: Chybné křížení elementů

Zde je zcela zřejmé překřížení, které nám vzniklo uzavření elementu pomocí koncového tagu jmeno dříve než jsme ukončili element koncovým tagem prijmeni. Tento zápis je v XML nepřipustný.



Obrázek 4: Správný kód bez křížení elementů

Na Obrázku 2: *Správný kód bez křížení elementů* jsme tento problém vyřešili pomocí přehození koncového tagu jmeno s počátečním tagem prijmeni.

2.1.6 XML ATRIBUTY A JEJICH PRAVIDLA

Stejně jako v HTML můžeme v XML využít atributy. Atribut lze využít pouze v počátečním tagu elementu případně ve zkráceném zápisu prázdného tagu. Atribut nám v XML upřesňuje či doplňuje informace o elementu.

```
<JmenoElementu NazevAtributu="hodnota atributu"></JmenoElementu>
<JmenoElementu NazevAtributu="hodnota atributu" />
```

Na prvním příkladu vidíte způsob zápisu atributu u neprázdných elementů a v druhém případě pak způsob zápisu atributu u zkráceného tzv. prázdného elementu. Abychom rozlišili část jméno elementu od názvu atributu, musíme mezi tyto dvě části vložit mezeru. Po názvu atributu následuje znaménko „=“, za kterým následuje hodnota atributu uzavřená do uvozovek či apostrofů. XML dovoluje si vybrat zejména z důvodu, že mohou nastat situace, kdy budeme muset použít v hodnotě atributu jeden z těchto znaků. Abychom zabránili chybám, používáme jeden druh symbolů k uzavření a druhý druh symbolů do hodnoty atributu. Stejně jako u Elementů můžeme využívat a vytvářet atributy dle své libosti, ale musíme dodržet opět stejná pravidla, jako při zápisu elementů viz Pravidla pro názvy XML elementů a velká a malá písmena. Hodnotu atributu můžeme uzavřít do uvozovek či apostrofů. Na následujících příkladech se opět znázorníme tuto problematiku.

```
<JEle aaa="text" text"></JEle> //chyba v ohraničení i uvnitř hodnoty jsou
                               stejné symboly
<JEle aaa='text' text'\></JEle> //chyba stejného charakteru
<JEle aaa="text" text"></JEle> //správně
<JEle aaa='text' text'\></JEle> //správně
```

Další chyby mohou vzniknout v případě, že se počáteční symbol nebude shodovat s koncovým symbolem. Opět si uvedeme příklady.

```
<JEle aaa="text'\></JEle> //chyba neshoduje se počáteční a koncový symbol
<JEle aaa='text"></JEle> //chyba neshoduje se počáteční a koncový symbol
<JEle aaa="text"></JEle> //správně
<JEle aaa='text'\></JEle> //správně
```

V případě, že chceme použít více atributů, musíme je oddělit mezerami. (5)

```
<JmenoElementu NazevAtributu1="HodnotaAtributu" NazevAtributu2=
"hodnotaatributu"></JmenoElementu>
```

2.1.7 ZAKÁZANÉ ZNAKY V XML

Stejně jako jakýkoliv jiný jazyk, má i XML znaky se specifickým významem, které nelze použít jinak, jedná se o tzv. rezervované znaky. Může se ovšem stát, že tyto znaky budeme muset i přesto použít a tak místo nich můžeme využít tzv. vestavěné znakové entity. Které znaky můžeme nahradit a jak, nám znázorňuje následující tabulka.

Znak	Entita	Rezervované pro označení
>	<	Konce tagu
<	>	Začátku tagu
&	&	Začátku entity
“	"	Hodnoty atributu
‘	'	Hodnoty atributu

Tabulka 1: Zakázané znaky

Entity pro apostrof a uvozovky využijeme zejména v případě, kdy potřebujeme v hodnotě atributu uvést oba tyto znaky a neměli bychom poté možnost uzavřít hodnotu do uvozovek či apostrofů. Znaky <, > a & je přímo nutné nahradit entitami nebo pomocí CDATA⁵, protože jinak nastane chyba. (5)

⁵ Sekce označující větší část obsahu elementu, kde se vyskytuje velké množství zakázaných znaků. (6)

3 OPENXML DOKUMENT MS EXCELU 2010 A JEHO VNITŘNÍ STRUKTURA

„Aplikace Excel 2010 používá formát souborů, který je založený na jazyce XML.“ (7) Díky tomu mohou různé aplikace využívat data vytvořená pomocí XML kódu. Těmto formátům se říká otevřené formáty Office Open XML. „Každý sešit je tak tvořen několika oddělenými komponentami⁶, které jsou spojené do jednoho souboru a následně automaticky zkomprimovány.“ (7) Díky čemuž se snižuje jejich velikost.

Aby se rozlišily soubory s otevřeným formátem od původních v binárním formátu, začaly se používat koncovky obsahující písmeno x.

3.1 VNITŘNÍ STRUKTURA A ORGANIZACE OPENXML DOKUMENTU MS EXCELU 2010

Abychom se dostali do složky s xml soubory, musíme přejmenovat koncovku souboru z xlsx na koncovku zip. Následně musíme soubor rozbalit.

3.1.1 OBSAH SOUBORU ZIP

Každý xml soubor obsahuje složku _rels, docProps, xl a soubor [Content_Types]. Soubory a složky mají hierarchické uspořádání.

Název položky	Datum změny	Typ	Velikost
_rels	27.2.2014 19:28	Složka souborů	
docProps	27.2.2014 19:28	Složka souborů	
xl	27.2.2014 19:28	Složka souborů	
[Content_Types].xml	27.2.2014 19:58	Soubor XML	2 kB

Obrázek 5: Obsah souboru zip

Soubor [Content_Types]

Soubor [Content_Types] eviduje a přiřazuje jednotlivé typy souborů obsažené ve složkách zipu. Určuje nám cestu ke konkrétním souborům.

Obsah složky _rels

Tato složka obsahuje pouze jeden soubor a to workbook.xml.rels, který v sobě zahrnuje způsob propojení jednotlivých souborů ve složce zip.

⁶ Označení pro soustavu souborů

Obsah složky docProps

Ve složce docProps nalezneme dva soubory s vlastnostmi dokumentu Excel, a to app.xml a core.xml. Soubor app.xml nese informace o zařízení (počítač, tablet nebo chytrý telefon) a o aplikaci, na které byl dokument Excelu vytvořen. Soubor core.xml pak definuje informace o autorovi, který tento soubor vytvořil a uživateli, kterým byl zmodifikován.

Obsah složky xl

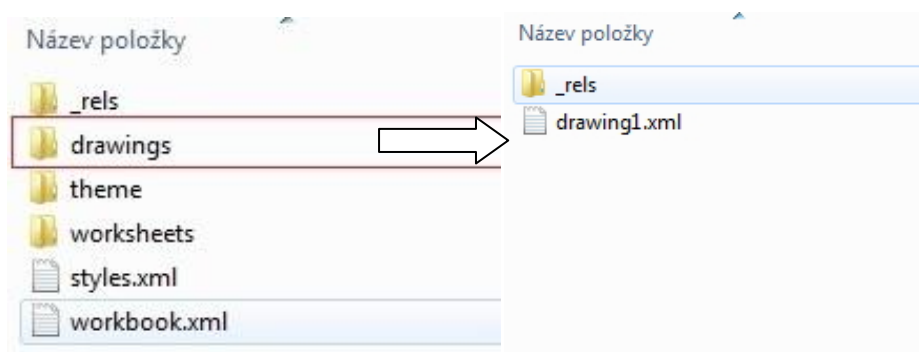
Složka xl je nejobsáhlejší složka ze všech. Obsahuje další tři podsložky a to _rels, theme, worksheets a dva soubory, které se jmenují styles.xml a workbook.xml. V souboru styles.xml nalezneme styly použité v dokumentu a v souboru workbook.xml informace o nastavení vzhledu programu.

Název položky	Datum změny	Typ	Velikost
_rels	15.2.2014 12:09	Složka souborů	
theme	15.2.2014 12:09	Složka souborů	
worksheets	15.2.2014 12:09	Složka souborů	
styles.xml		Soubor XML	2 kB
workbook.xml		Soubor XML	1 kB

Obrázek 6: Obsah složky xl

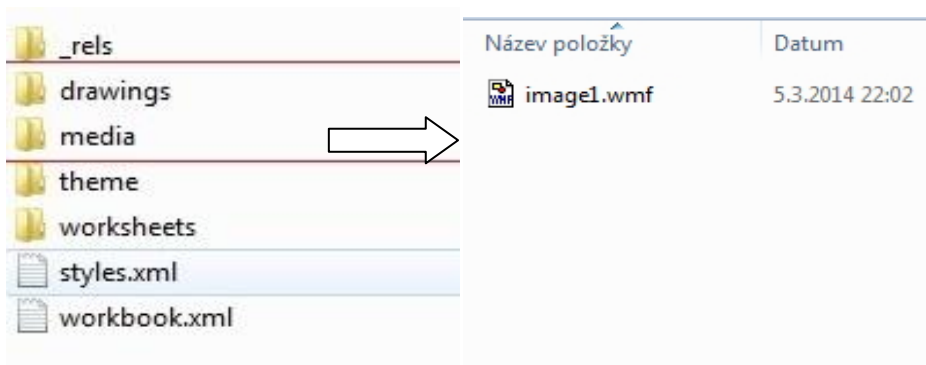
Rozšířený obsah složky xl

Složka drawings slouží pro ukládání informací pro vlastnosti, jako je velikost, umístění, název aj., pro prvky jako je wordart, textová pole, clipart, snímek obrazovky, obrázek, graf, SmartArt a další. Tyto informace lze najít v souboru drawing1.xml.



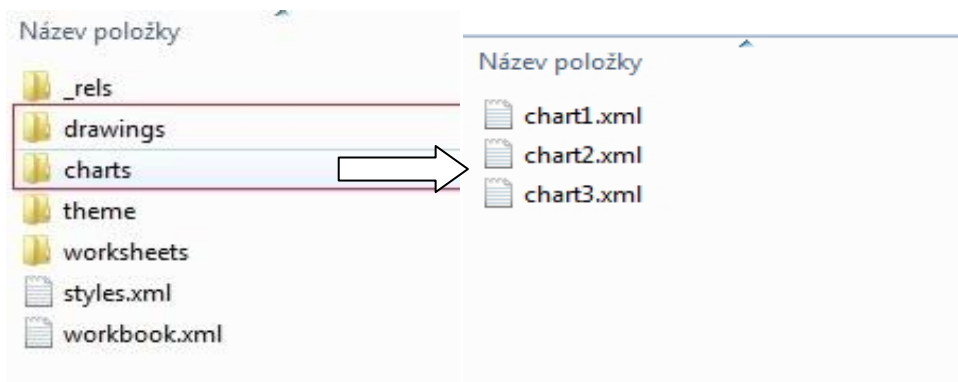
Obrázek 7: Složka drawings

Složka media obsahuje soubor image1.wmf s originály multimediálních souborů, jako jsou například clipart, snímek obrazovky nebo obrázek.



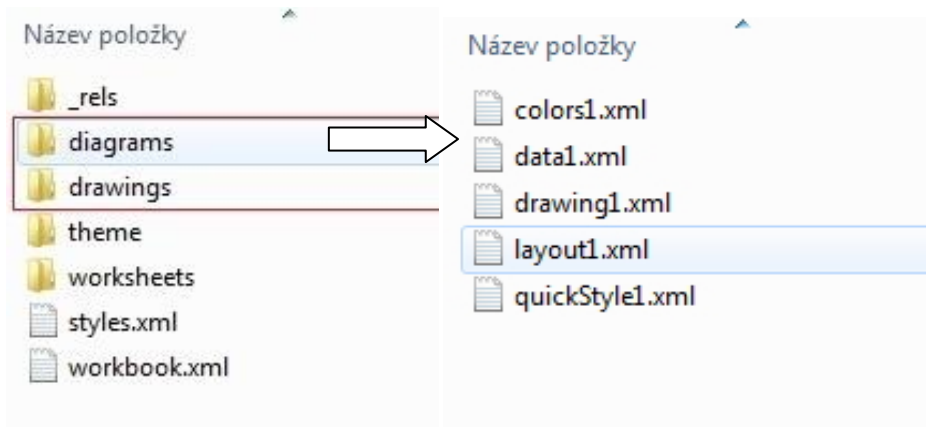
Obrázek 8: Složka media

Pokud do Excelu vložíme Graf, přibudou nám složky charts se soubory chartx.xml (místo x je napsáno pořadí grafu vloženého do souboru) a drawings.



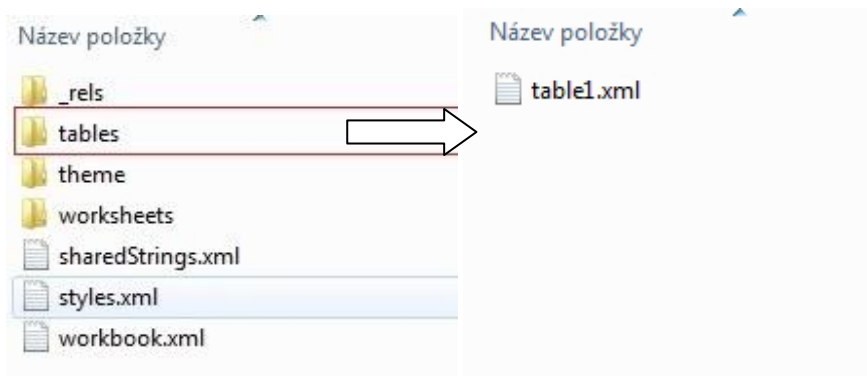
Obrázek 9: Složka charts

Složka diagrams obsahuje soubory colors1.xml, data1.xml, drawing1.xml, layout1.xml a quickStyle.xml s informacemi o vzhledu SmartArtu.



Obrázek 10: Složka diagrams

Ve složce tables lze najít soubor table1.xml s informacemi o tabulce.



Obrázek 11: Složka tables

Obsah složky _rels

Ve složce _rel se nachází soubor workbook.xml.rels, ve kterém naleznete definice a identifikátory (Id) na jednotlivé soubory, a to konkrétně na všechny listy, na styles.xml a theme1.xml. Pokud budeme vytvářet další listy, je nutné jim přidělit Id. Přidáme-li, nebo ubereme-li jeden list, všechny Id se znovu přidělí.

Obsah složky theme

Složka theme má v sobě uložen pouze soubor theme1.xml. Tento soubor uvnitř ukrývá nejen způsob formátování textu zvoleného schématu, jako jsou například fonty a barvy, ale i rozvržení sloupců a řádek pro dané schéma.

Obsah složky worksheets

Název položky	Datum změny	Typ	Velikost
sheet1.xml		Soubor XML	1 kB
sheet2.xml		Soubor XML	1 kB
sheet3.xml		Soubor XML	1 kB

Obrázek 12: Obsah složky worksheets

Složka worksheets obsahuje pro každý list jeden ze souborů, které jsou číslovány. V našem případě je to soubor sheet1.xml, sheet2.xml a sheet3.xml. Počet těchto souborů ovšem může narůstat v závislosti na počtu listů v souboru Excel.

3.2 VÝZNAMNÉ SOUBORY A JICH ELEMENTY

V této části si podrobněji rozebereme významné soubory, se kterými se můžete nejčastěji setkat a jejich elementy.

3.2.1 VNITŘNÍ STRUKTURA SOUBORU [CONTENT_TYPES].XML

Tento soubor se zabývá typem a koncovkami souborů.

Zdrojový kód

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="http://schemas.openxmlformats.org/
package/2006/content-types">
  <Default Extension="bin" ContentType="application/
vnd.openxmlformats-officedocument.spreadsheetml.printerSettings"/>
  <Default Extension="rels" ContentType="application/
vnd.openxmlformats-package.relationships+xml"/>
  <Default Extension="xml" ContentType="application/xml"/>
  <Override PartName="/xl/workbook.xml" ContentType="application/
vnd.openxmlformats-officedocument.spreadsheetml.sheet.main+xml"/>
  <Override PartName="/xl/worksheets/sheet1.xml"
ContentType="application/
vnd.openxmlformats-officedocument.spreadsheetml.worksheet+xml"/>
  <Override PartName="/xl/worksheets/sheet2.xml"
ContentType="application/
vnd.openxmlformats-officedocument.spreadsheetml.worksheet+xml"/>
  <Override PartName="/xl/worksheets/sheet3.xml"
ContentType="application/
vnd.openxmlformats-officedocument.spreadsheetml.worksheet+xml"/>
  <Override PartName="/xl/theme/theme1.xml" ContentType="application/
vnd.openxmlformats-officedocument.theme+xml"/>
  <Override PartName="/xl/styles.xml" ContentType="application/
vnd.openxmlformats-officedocument.spreadsheetml.styles+xml"/>
  <Override PartName="/docProps/core.xml" ContentType="application/
vnd.openxmlformats-package.core-properties+xml"/>
  <Override PartName="/docProps/app.xml" ContentType="application/
vnd.openxmlformats-officedocument.extended-properties+xml"/>
</Types>
```

Vysvětlení kódu

Celý blok dat je zabalen do kořenového elementu <Types></Types>, který jak z názvu vyplývá, se bude zabývat typem souborů.

```
<Default Extension="bin" ContentType="application/vnd.openxmlformats-office
document.spreadsheetml.printerSettings"/>
```

Element <Default /> definuje typ zdrojových dat pro jednotlivé přípony souborů. Atribut **Extension** udává příponu souborů v extrahovaném dokumentu, atribut **ContentType** pak odkazuje na typ obsahu.

Element `<Override PartName="/xl/workbook.xml" ContentType="application/vnd.openxmlformats-officedocument.spreadsheetml.sheet.main+xml"/>` věnuje pozornost konkrétním souborům. Atribut `PartName` odkazuje na určitý soubor ve složce `X`. V našem případě je to soubor `workbook`, který najdeme ve složce `xl` a má koncovku `xml`. Atribut `ContentType` nám opět ukazuje na definici typu souboru. (8)

3.2.2 VNITŘNÍ STRUKTURA SOUBORU APP.XML

V souboru `app.xml` naleznete informace o aplikaci, ve které byl dokument vytvořen.

Zdrojový kód

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Properties
xmlns="http://schemas.openxmlformats.org/officeDocument/2006/extended-
properties"
xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPropsV
Types">
  <Application>Microsoft Excel</Application>
  <DocSecurity>0</DocSecurity>
  <ScaleCrop>>false</ScaleCrop>
  <HeadingPairs>
    <vt:vector size="2" baseType="variant">
      <vt:variant>
        <vt:lpstr>listy</vt:lpstr>
      </vt:variant>
      <vt:variant>
        <vt:i4>3</vt:i4>
      </vt:variant>
    </vt:vector>
  </HeadingPairs>
  <TitlesOfParts>
    <vt:vector size="3" baseType="lpstr">
      <vt:lpstr>List1</vt:lpstr>
      <vt:lpstr>List2</vt:lpstr>
      <vt:lpstr>List3</vt:lpstr>
    </vt:vector>
  </TitlesOfParts>
  <LinksUpToDate>>false</LinksUpToDate>
  <SharedDoc>>false</SharedDoc>
  <HyperlinksChanged>>false</HyperlinksChanged>
  <AppVersion>14.0300</AppVersion>
</Properties>
```

Vysvětlení kódu

Tag `<Properties>` nám atributem `xmlns` odkazuje na jmenné prostory.

`<Application></Application>` je element, který nám určuje, v jaké aplikaci byl soubor vytvořen.

Mezi počáteční tag `<DocSecurity>` a koncový tag `</DocSecurity>` zapisujeme číslo, které nám pak definuje úroveň zabezpečení, v našem případě jsme zvolili nulu, což znamená žádné zabezpečení. Tento tag může dále obsahovat číslo 1, 2, 4 a 8. Čím je vyšší číslo, tím větší úroveň zabezpečení je. Zvláštností Excelu narozdíl od Wordu je to, že nemůže v tomto tagu používat hodnoty 4 a 8, tyto hodnoty totiž nepodporuje. Tím pádem nám zbyly pouze tři úrovně zabezpečení a to 0, 1 a 2. (9)

Element `<ScaleCrop></ScaleCrop>` nám vytyčuje způsob zobrazení náhledu dokumentu. Je datového typu boolean, to znamená, že může obsahovat pouze dvě hodnoty a to buď true nebo false. Pokud je nastavena hodnota true, bude náhled vytvářen z celého dokumentu, bude-li zvolena hodnota false, bude náhled tvořen pouze výřezem. V praxi se ovšem používá přednastavená hodnota false, a to z toho důvodu, že malý náhled na celý dokument, bude pravděpodobně nečitelný. (10)

V elementu `<HeadingPairs></HeadingPairs>` se nachází `<vt:vector size="2" baseType="variant">`, vt nám určuje variantu třídy, která následuje za „:“. Vektor se používá pro případy, kdy jsou podřízené dva prvky. Variant, pro případy jednoho podřízeného prvku. It4 označuje celočíselné typy se znaménkem. Ipstr se používá pro typ řetězec. Atribut size nám značí, kolik záznamů máme očekávat. V našem případě jsou to dva záznamy a to listy a jejich počet. Atribut baseType označuje typ základní matice. (11)

Element `<TitlesOfParts></TitlesOfParts>` nám již definuje seznam listů, a to především jejich jméno, kde v tagu `<HeadingPairs><vt:vector size="3" baseType="lpstr">` nám atribut size opět značí počet, ale tentokrát listů. Uvnitř elementu `<vt:vector size="3" baseType="lpstr"></vt:vector>` jsou pak všechny listy.

Element `<LinksUpToDate></LinksUpToDate>` je datového typu boolean. Hodnota true znamená aktualizované hypertextové odkazy. Hodnota false pak označuje neaktualizované odkazy, to znamená, že mohou být zastaralé. (12)

Element `<SharedDoc></SharedDoc>` je stejného datového typu jako element `<LinksUpToDate></LinksUpToDate>`. Při nabývání hodnoty true znamená, že je dokument spuštěný několika uživateli najednou. Při hodnotě false pak je zobrazený pouze na jednom PC. (15)

Element `<HyperlinksChanged></HyperlinksChanged>` definuje, kdy se mají hypertextové odkazy měnit či nikoliv. Při volbě `true` je potřeba při otevření aktualizovat jeden či více hypertextových odkazů. Při volbě `false` není potřeba hypertextové odkazy při otevření aktualizovat. (16)

Do posledního elementu `<AppVersion> 14.0300</AppVersion>` se zapisuje verze aplikace, kde čísla před tečkou označují verzi. V následující tabulce uvidíte přehledně, pro jaký Excel se používá konkrétní číslo verze.

Označení verze textem	Označení verze číslem
Excel 5	5
Excel 6	6
Excel 95	7
Excel 97	8
Excel 2000	9
Excel 2002	10
Excel 2003	11
Excel 2007	12
Excel 2010	14
Excel 2013	15

Tabulka 2: Číselné označení verze Excelu (13)

Číslo za tečkou nám pak značí, o jakou aplikaci kancelářské sady Microsoft Office se jedná, zda o Excel (.0300) či Word (.0000) apod.

3.2.3 VNITŘNÍ STRUKTURA SOUBORU DOCPROPS\CORE.XML

Soubor `core.xml` informuje o čase a datu vytvoření a úpravy.

Zdrojový kód

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cp:coreProperties
xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-
properties" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dcmitype="http://purl.org/dc/dcmitype/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <dc:creator>Andy</dc:creator>
  <cp:lastModifiedBy>Andy</cp:lastModifiedBy>
  <dcterms:created xsi:type="dcterms:W3CDTF">2013-03-05T09:34:32Z
  </dcterms:created>
  <dcterms:modified xsi:type="dcterms:W3CDTF">2013-03-05T09:38:51Z
  </dcterms:modified>
</cp:coreProperties>
```

Vysvětlení kódu

Celý soubor neobsahuje nic jiného než vlastnosti dokumentu, shodující se s vlastnostmi souboru Excel, jenž se nachází ve vlastnostech na kartě podrobnosti.

V elementu `<dc:creator></dc:creator>` se udává, kdo tento soubor vytvořil. Použito je většinou jméno zvolené při instalaci sady Microsoft Office, které se často shoduje se jménem uživatele, nebo názvem počítače.

Do elementu `<cp:lastModifiedBy></cp:lastModifiedBy>` se pak zaznamená, kdo daný soubor pozměnil, nebo naposledy uložil.

Elementu `<dcterms:created xsi:type="dcterms:W3CDTF"></dcterms:created>` obsahuje datum vytvoření souboru.

V elementu `<dcterms:modified xsi:type="dcterms:W3CDTF"></dcterms:modified>` nalezneme čas změny dokumentu Excel.

3.2.4 VNITŘNÍ STUKTURA SOUBORU WORKBOOK.XML.RELS

Soubor workbook.xml.rels obsahuje relace.

Zdrojový kód

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/
package/2006/relationships">
  <Relationship Id="rId3"
  Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/worksheet" Target="worksheets/sheet3.xml"/>
  <Relationship Id="rId2"
  Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/worksheet" Target="worksheets/sheet2.xml"/>
  <Relationship Id="rId1"
  Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/worksheet" Target="worksheets/sheet1.xml"/>
  <Relationship Id="rId5"
  Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/styles" Target="styles.xml"/>
  <Relationship Id="rId4"
  Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/theme" Target="theme/theme1.xml"/>
</Relationships>
```

Vysvětlení kódu

Tag `<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">` je soupisem souborů složky xl a jejich typů.

V elementu `<Relationship Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/worksheet" Target="worksheets/sheet3.xml"/>` nám atribut `Type` definuje, že je soubor typu `list`. Atribut `Target` obsahuje cestu k určitému souboru, ke kterému se váže `Id`. Takto si Excel vytváří relace.

V elementu `<Relationship Id="rId5" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles" Target="styles.xml"/>` atribut `Type` odkazuje na definici typu `styly`.

V elementu `<Relationship Id="rId4" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme" Target="theme/theme1.xml"/>` atribut `Type` pro změnu určuje soubor, který je typu `téma` a atribut `Target` odkazuje přímo na soubor `theme1.xml`.

4 EXCEL 2010 V PŘÍKLADECH

V této kapitole si vysvětlíme na příkladech nejčastější úkony uživatelů s aplikací Excel.

4.1 ULOŽENÍ DATOVÉ HODNOTY DO BUŇKY

Při zápisu dat do buňky se hlavní změny odehrávají v souboru sheetx.xml, kde místo x bude doplněné číslo listu. Kde najít tento soubor jsme rozebírali již v části Obsah souboru zip na straně 10.

Popis příkladu: V buňce s adresou B2 na listu 1 se nachází hodnota 4.

Složka s xml kódy: ulozeni_datove_hodnoty_do_bunky_1

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
  <row r="2" spans="2:2" x14ac:dyDescent="0.25">
    <c r="B2">
      <v>4</v>
    </c>
  </row>
</sheetData>
```

Vysvětlení kódu

```
<row r="2" spans="2:2" x14ac:dyDescent="0.25">
```

Tag `<row>` slouží ke specifikování řádky. Číslo řádky, do které se budou vkládat data, je zadáno v atributu `r`. V našem případě budeme ukládat hodnotu do druhé řádky.

Atribut `spans` nám označuje čísla sloupců, do kterých se budou data zapisovat. Sloupce jsou uváděny rozsahem a nelze přesně určit adresu buňky. První číslo vyjadřuje číslo počátečního sloupce, následuje dvojtečka a poté číslo posledního sloupce. V našem případě se budou vkládat data pouze do druhého sloupce.

```
<c r="B2">
```

Tag `<c>` nám blíže vymezuje buňku, do které budeme vkládat data. Atribut `r` definuje adresu buňky, do které budou vkládána data, a je zapisován v klasickém formátu (sloupec písmenem, řádek číslem), v našem případě buňku "B2".


```
<v>4</v>
```

Tag `<v>` nám značí, že budeme do buňky vkládat číselnou hodnotu, nikoliv funkci. Pro tu je určen atribut `<f>`. Jak by to bylo v případě funkce, si podrobněji popíšeme v kapitole funkce na straně 40. Mezi počáteční tag `<v>` a koncový tag `</v>` se vkládá hodnota, která má být do buňky vložena, v tomto příkladu je to číslo 4.

Úprava kódu

Vložení do jiné buňky

Popis příkladu: V buňce C8 je číslo 125.

Složka s xml kódy: ulozeni_datove_hodnoty_do_bunky_2

Pokud bychom chtěli vložit do buňky C8 hodnotu 125, musíme kód upravit. Buňka C8 nám značí, že se bude jednat o osmou řádku třetí sloupec. Do atributu `r` v tagu `<row>` musíme napsat číslo 8 a do atributu `spans`, protože vkládáme hodnotu pouze do jednoho sloupce, musíme zvolit rozsah 3:3, data se, budou vkládat od třetího do třetího sloupce. Do atributu `r` v tagu `<c>` pak zapíšeme odkaz na buňku C8. Dále víme, že budeme ukládat datovou hodnotu a ne funkci. Zvolíme počáteční tag `<v>` a koncový tag `</v>`, uvnitř něhož bude číslo 125. Celý zápis bude pak vypadat takto:

```
<sheetData>
  <row r="8"spans="3:3" x14ac:dyDescent="0.25">
    <c r="C8">
      <v>125</v>
    </c>
  </row>
</sheetData>
```

Vložení více hodnot do více buněk

Popis příkladu: Buňka B2 na listu 1 obsahuje číslici 4 a v buňce s adresou C8 na listu 1 nalezneme hodnotu 125.

Složka s xml kódy: ulozeni_datove_hodnoty_do_bunky_3

V tagu `<sheetData>` se budou nacházet dva tagy `<row>` pro hodnotu 4 a 125. Jediná změna uvnitř těchto tagů bude v atributu `spans`. Jak v prvním tak v druhém tagu `<row>` bude hodnota atributu `spans` 2:3 zapisovat se bude od druhého do třetího sloupce.

Kód bude vypadat takto:

```
<sheetData>
  <row r="2" spans="2:3">
    <c r="B2">
      <v>
        4
      </v>
    </c>
  </row>
  <row r="8" spans="2:3">
    <c r="C8">
      <v>
        125
      </v>
    </c>
  </row>
</sheetData>
```

4.2 ZJIŠTĚNÍ TYPU DAT

Typy dat zjišťujeme v souboru Styles.xml.

Složka s podsložkami: zjistení typu dat

4.2.1 TYP OBECNÝ

Soubor sheet1.xml je podobný jako sheet1.xml v příkladu Uložení datové hodnoty do buňky, který naleznete na straně 20, a proto si ho není potřeba vypisovat.

Popis příkladu: Do buňky B2 až B10 jsme vložili hodnotu 123 a změnili jsme Formát buněk na Obecný.

Složka s xml kódy: Obecný_typ

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
  <row r="2" spans="2:2" x14ac:dyDescent="0.25">
    <c r="B2" s="1">
      <v>123</v>
    </c>
  </row>
</sheetData>
```

Vysvětlení kódu

Jelikož obecný formát nabízí pouze jeden způsob zobrazení dat, je zápis poměrně jednoduchý. V tagu <c> atribut s obsahuje číslo, ze kterého po přičtení čísla 1, dostaneme pořadí elementu <xf> v elementu <cellXfs></cellXfs> v souboru styles.xml, kde je

definovaný vzhled buňky. Pokud zvolíme stejných formát více buňkám, budou mít stejné číslo v atributu `s`.

Styles.xml

Část zdrojového kódu

```
<cellXfs count="2">
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"/>
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"
    applyNumberFormat="1"/>
</cellXfs>
```

Vysvětlení kódu

Hodnota atributu `numFmtId` v elementu `<xf>` nám definuje formát a způsob zobrazení. V našem případě obsahuje hodnotu `0`, to znamená, že se bude jednat o obecný typ. Atribut `fontId` obsahuje číslo `0`. Toto číslo nám specifikuje, že bude použitý první font, který nalezneme ve stejném souboru (`styles.xml`) uvnitř elementu ``. Atribut `fillId` nám svou hodnotou poukazuje na barvu pozadí, která se nachází v elementu `<fill></fill>`. Na ohraničení buňky odkazuje do elementu `<border></border>` svou hodnotou atribut `borderId`. Tyto elementy si zobrazí, pouze pokud hodnota bude jiná než `0`.

4.2.2 TYP ZLOMKY

Excel si ukládá zlomky jako desetinná čísla.

Popis příkladu

	A	B
1		
2		1/8
3		3/10
4		8/16
5		1 10/100

Obrázek 13: Obecný typ

Do buňky B2 až B5 jsme vložili hodnoty ve tvaru zlomku.

Složka s xml kódy: Typ_zlomky

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
<row r="2" spans="2:2" x14ac:dyDescent="0.25">
<c r="B2" s="4">
<v>0.1</v>
</c>
</row>
<row r="3" spans="2:2" x14ac:dyDescent="0.25">
<c r="B3" s="3">
<v>0.3</v>
</c>
</row>
<row r="4" spans="2:2" x14ac:dyDescent="0.25">
<c r="B4" s="2">
<v>0.5</v>
</c>
</row>
<row r="5" spans="2:2" x14ac:dyDescent="0.25">
<c r="B5" s="1">
<v>1.1000000000000001</v>
</c>
</row>
</sheetData>
```

Vysvětlení kódu

Pokud k hodnotě atributu `s` v tagu `<c>` přičteme jedničku, dostaneme číslo vyjadřující pořadí tagu `<xf>` v elementu `<cellXfs></cellXfs>`.

Styles.xml

Část zdrojového kódu

```
<numFmts count="4">
  <numFmt numFmtId="165" formatCode="#"&quot;&quot;??/100"/>
  <numFmt numFmtId="166" formatCode="#"&quot;&quot;??/16"/>
  <numFmt numFmtId="167" formatCode="#"&quot;&quot;??/??"/>
  <numFmt numFmtId="168" formatCode="#"&quot;&quot;?/8"/>
</numFmts>
...
<cellXfs count="5">
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"/>
  <xf numFmtId="165" fontId="0" fillId="0" borderId="0" xfId="0"
  applyNumberFormat="1"/>
  <xf numFmtId="166" fontId="0" fillId="0" borderId="0" xfId="0"
  applyNumberFormat="1"/>
  <xf numFmtId="167" fontId="0" fillId="0" borderId="0" xfId="0"
  applyNumberFormat="1"/>
  <xf numFmtId="168" fontId="0" fillId="0" borderId="0" xfId="0"
  applyNumberFormat="1"/>
</cellXfs>
```

Vysvětlení kódu

První část kódu nám specifikuje vzhled modifikovaných formátů. Obsah elementu `<numFmts count="4"></numFmts>` nám určuje, jak má obsah dané buňky vypadat.

Hodnota za lomítkem nám upřesňuje, zda se bude jednat o setiny, šestnáctiny, osminy nebo to není specifikováno viz. ???/???. Typ ???/??? nám označuje zlomky, které mají omezení pouze v počtu čísel ve jmenovateli a v čitateli.

Druhá část pak odkazuje na tyto a základní formáty pomocí atributu `numFmtId`. Ostatní atributy jsou vysvětlené na straně 23.

V dalších typech již způsob odkazování řešit nebudeme.

4.2.3 TYP DATUM

Typ datum se vypočítává poměrně složitým způsobem.

Popis příkladu

Do buněk B4 až B18 jsme vložili různé způsoby zápisu data.

	A	B
4		1.2
5		1.2.14
6		01.02.14
7		1-2.
8		1.2.14
9		01-2-14
10		2-14
11		únor 14
12		1. únor 2014
13		1.2.14 12:00 AM
14		1.2.14 0:00
15		ú
16		ú-14
17		1.2.2014
18		1-2-2014

Obrázek 14: Typ datum

Složka s xml kódy: Typ_datum

Sheet1.xml

Část zdrojového kódu

```
<row r="2" spans="2:2" x14ac:dyDescent="0.25">
  <c r="B2" s="1">
    <v>41671</v>
  </c>
</row>
<row r="3" spans="2:2" x14ac:dyDescent="0.25">
  <c r="B3" s="2">
    <v>41671</v>
  </c>
</row>
```

Vysvětlení kódu

Jak je vidět, datum se nám neukládá jako 1. 2. 2014, ale jako číslo. To nám vyjadřuje počet dnů od 31. 12. 1899. Můžeme si to jednoduše ověřit, pokud napíšeme číslo jedna v obecném formátu a změníme formát na typ datum, objeví se nám datum 1. 1. 1900. Jestli zapíšeme číslo dvě a provedeme stejný postupu jako výše, změní se nám datum a to pouze o jeden den, zobrazí se datum 2. 1. 1900.

Styles.xml

Část zdrojového kódu

```
<numFmts count="16">
  <numFmt numFmtId="166" formatCode="[$-F800]dddd\, \ mmmm\ dd\, \
  yyyy"/>
  <numFmt numFmtId="167" formatCode="d/m;@"/>
  <numFmt numFmtId="168" formatCode="d/m/yy;@"/>
  <numFmt numFmtId="169" formatCode="dd/mm/yy;@"/>
  <numFmt numFmtId="170" formatCode="[$-405]d\-mmm\.;@"/>
  <numFmt numFmtId="171" formatCode="[$-405]d/mmm/yy;@"/>
  <numFmt numFmtId="172" formatCode="[$-405]dd\-mmm\-yy;@"/>
  <numFmt numFmtId="173" formatCode="[$-405]mmm\-yy;@"/>
  <numFmt numFmtId="174" formatCode="[$-405]mmmm\ yy;@"/>
  <numFmt numFmtId="175" formatCode="[$-405]d\.\ mmmm\ yyyy;@"/>
  <numFmt numFmtId="176" formatCode="[$-409]d/m/yy\ h:mm\ AM/PM;@"/>
  <numFmt numFmtId="177" formatCode="d/m/yy\ h:mm;@"/>
  <numFmt numFmtId="178" formatCode="[$-405]mmmmm;@"/>
  <numFmt numFmtId="179" formatCode="[$-405]mmmmm\-yy;@"/>
  <numFmt numFmtId="180" formatCode="d/m/yyyy;@"/>
  <numFmt numFmtId="181" formatCode="[$-405]d\-mmm\-yyyy;@"/>
</numFmts>
```

Vysvětlení kódu

Tato část kódu nám popisuje, jak se má naformátovat datum. V tagu `<numFmt>` nám atribut `numFmtId` vyjadřuje identifikační čísla formátů, které nejsou základní a mají modifikovaný zápis. Malé písmeno `d` nám označuje den, `m` nám značí měsíc a `y` nám vyjadřuje rok. Proč je několik písmen za sebou, nám objasní následující tabulky.

Zápis v xml	Zobrazí se	Výsledek pro čísla menší než deset	Výsledek pro čísla větší než deset
<code>d</code>	Den v měsíci jako číslo	1	11
<code>dd</code>	den v měsíci jako číslo, ale v případě, že je toto číslo menší než deset vloží před něj nulu	01	11
<code>ddd</code>	První tři písmena názvu dne v týdnu (v českém jazyce se zobrazí pouze dvě písmena, a to díky vlivu národnostní zvyklosti)	tue (út)	tue (út)
<code>dddd</code>	Název dne ve slovním vyjádření	Úterý	Úterý

Tabulka 3: Den (11)

Zápis v xml	Zobrazí se	Výsledek pro čísla menší než deset	Výsledek pro čísla větší než deset
m	Měsíc jako číslo	1	11
mm	Měsíc jako číslo, ale v případě, že je toto číslo menší než deset vloží před něj nulu	01	11
mmm	První tři písmena názvu dne v týdnu (Ve verzipro český jazyk se bohužel vyskytují chyby a místo prvních třech písmen zobrazí pouze měsíc jako číslo)	leden (1)	leden (1)
mmmm	Název měsíce slovně	Leden	Leden
mmmmm	První písmeno z názvu měsíce	L	L

Tabulka 4: Měsíc (11)

Zápis v xml	Zobrazí se	Formát datumu menšího než 10
yy	Poslední dvě číslice roku	14
yyyy	Všechny čtyři číslice v roce	2014

Tabulka 5: Rok (11)

[[\\$-405](#)]toto číslo určuje jakým jazykem, má být datum zobrazené. Číslo [405](#) znamená český jazyk, pokud zapíšeme číslo [409](#), bude zvolen anglický jazyk a číslo [407](#) nám vypíše číslo v německém jazyce. Znak \. nebo – nám značí, jaký znak se má použít mezi dnem, měsícem a rokem, lze udělat i jejich kombinace. Znak _ má význam mezery. (14)

4.2.4 TYP ČAS

Stejně jako typ datum nám ukládá Excel typ čas jinak než je zobrazeno v aplikaci.

Popis příkladu

	A	B
1		
2		11:30:00
3		11:30
4		11:30 AM
5		11:30:00
6		11:30:00 AM
7		30:00,0
8		11:30:00
9		0.1.00 11:30 AM
10		0.1.00 11:30

Obrázek 15: Typ čas

Buňka B2 až B10 obsahuje čas různého formátu.

Složka s xml kódy: Typ_cas

Sheet1.xml

Část zdrojového kódu

```
<row r="2" spans="2:2" x14ac:dyDescent="0.25">
  <c r="B2" s="2">
    <v>0.479166666666666669</v>
  </c>
</row>
<row r="3" spans="2:2" x14ac:dyDescent="0.25">
  <c r="B3" s="3">
    <v>0.479166666666666669</v>
  </c>
</row>
```

Vysvětlení kódu

Excel neukládá čas ve formátu 11:30:00, ale jako desetinné číslo, které můžeme přibližně vypočítat v několika krocích. Jako první si musíme převést minuty na hodiny, v našem případě $30/60=0,5$ toto číslo přičteme k hodinám $11+0,5=11,5$. Na závěr toto číslo vydělíme 24. Výpočet bude $11,5/25=0.479166666666667$. Čas se v Excelu ukládá jako desetinná část dne. Jak se bude toto číslo zobrazovat, opět nalezneme v souboru styles.xml.

Styles.xml

Část zdrojového kódu

```
<numFmts count="9">
  <numFmt numFmtId="165" formatCode="[$-F400]h:mm:ss\ AM/PM"/>
  <numFmt numFmtId="166" formatCode="h:mm;@"/>
  <numFmt numFmtId="167" formatCode="[$-409]h:mm\ AM/PM;@"/>
  <numFmt numFmtId="168" formatCode="h:mm:ss;@"/>
  <numFmt numFmtId="169" formatCode="[$-409]h:mm:ss\ AM/PM;@"/>
  <numFmt numFmtId="170" formatCode="mm:ss.0;@"/>
  <numFmt numFmtId="171" formatCode="[h]:mm:ss;@"/>
  <numFmt numFmtId="172" formatCode="[$-409]d/m/yy\ h:mm\ AM/PM;@"/>
  <numFmt numFmtId="173" formatCode="d/m/yy\ h:mm;@"/>
</numFmts>
```

Vysvětlení kódu

Písmeno **h** nám značí hodiny, **mm** znamená minuty a malá písmena **ss** označují sekundy. Pro formát času je přednastavený anglický jazyk, a to z důvodu hodnoty **AM** a **PM**.

4.2.5 TYP ČÍSLO

Tento typ patří mezi základní typy.

Popis příkladu

	A	B
1		
2		-1250,00
3		-1 250,00
4		-1250,000
5		-1250
6		1250,00
7		-1250,00
8		-1250,00

Obrázek 16: Typ číslo

Do buňky B2 až B8 jsme vložili číslo -1250 a nastavili různý formát.

Složka s xml kódy: Typ_cislo

Styles.xml

Část zdrojového kódu

```
<numFmts count="4">
  <numFmt numFmtId="164" formatCode="0.000"/>
  <numFmt numFmtId="165" formatCode="0.00; [Red]0.00"/>
  <numFmt numFmtId="166" formatCode="0.00_ ;\ -0.00\ "/>
  <numFmt numFmtId="167" formatCode="0.00_ ; [Red]\ -0.00\ "/>
</numFmts>
```

Vysvětlení kódu

Desetinná místa určuje tečka mezi nulami. Číslo se zaokrouhlí na námi požadovaný počet desetinných míst. Pokud se má zobrazit záporné číslo červeně použijeme hodnotu atributu `[Red]`. Má-li být záporné číslo s mínusem a červeně musíme zapsat hodnotu atributu `[Red]` a za ní číslo s mínusem. První část označuje, jak se mají zobrazit kladná čísla, druhá pak formátuje záporná čísla, tyto úseky oddělujeme znakem „;“.

4.2.6 TYP MATEMATICKÝ

Tento typ se příliš často nepoužívá.

Popis příkladu

	A	B
1		
2		1,00E-01
3		3,00E-01
4		5,00E-01
5		1,10E+00

Obrázek 17: Typ matematický

Do buňky B2 až B5 jsme vložili různé čísla a navolili formát matematický.

Složka s xml kódy: Typ_matematicky

Styles.xml

Část zdrojového kódu

```
<cellXfs count="2">
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"/>
  <xf numFmtId="11" fontId="0" fillId="0" borderId="0" xfId="0"
    applyNumberFormat="1"/>
</cellXfs>
```

Vysvětlení kódu

Atribut `numFmtId` v tagu `<xf>` obsahuje hodnotu `0` a `11`, `0` nám označuje obecný typ, který je použitý na zbytek tabulky a `11` znamená, že se bude jednat o typ matematický, bude se aplikovat tento vzorec `0.00E+00`.

4.2.7 TYP MĚNA

Typ měna se využívá především v účetnictví.

Popis příkladů

	A	B
1		
2		-1 250,00 Kč
3		-1 250,000 Kč
4		-1 250 Kč
5		-₹ 1 250,00
6		1 250,00 Kč
7		-1 250,00 Kč
8		-1 250,00 Kč

Obrázek 18: Typ měna

Buňka B2 až B8 jsme vložili hodnotu `-1250` a použili jsme různé formátování.

Složka s xml kódy: Typ_mena

Styles.xml

Část zdrojového kódu

```
<numFmts count="7">
  <numFmt numFmtId="7" formatCode="#,##0.00\ &quot;Kč&quot;;\-
#,##0.00\ &quot;Kč&quot;"/>
  <numFmt numFmtId="8" formatCode="#,##0.00\ &quot;Kč&quot;; [Red]\-
#,##0.00\ &quot;Kč&quot;"/>
  <numFmt numFmtId="164" formatCode="#,##0.00\ &quot;Kč&quot;"/>
  <numFmt numFmtId="166" formatCode="#,##0.000\ &quot;Kč&quot;"/>
  <numFmt numFmtId="167" formatCode="#,##0\ &quot;Kč&quot;"/>
  <numFmt numFmtId="168" formatCode="[$₴-445]\ #,##0.00"/>
  <numFmt numFmtId="169" formatCode="#,##0.00\ &quot;Kč&quot;;
[Red]\ #,##0.00\ &quot;Kč&quot;"/>
</numFmts>
```

Vysvětlení kódu

Pokud chceme oddělit tisíce vizuální metodou, používáme podobný zápis jako při značení počtu desetinných míst, místo nul ovšem zvolíme znak # zastupující čísla (desítky, stovky...) a tam, kde se má vložit mezera, zapíšeme znak „,“. Jestli se má znak měny zobrazit ještě před číslem, musíme ho zapsat do hranatých závorek před označením jazyku, pokud se má zobrazit až za číslem napíšeme ho až za znak „\“. Pro druhý případ musíme použít před měnu a za měnu slovíčko " (tento znak je vysvětlený na straně 9), které slouží v zápisu k oddělení čísel od měny.

4.2.8 TYP PROCENTA

Typ procenta se ukládá data jako desetinná čísla.

Popis příkladu

	A	B
1		
2		10,00%
3		30,00%
4		50,00%
5		110,00%

Obrázek 19: Typ procenta

Do buněk B2 až B5 jsme vložili čísla v procentuálním tvaru.

Složka s xml kódy: Typ_procenta

Styles.xml

Část zdrojového kódu

```
<cellXfs count="2">
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0" xfId="0"/>
  <xf numFmtId="10" fontId="0" fillId="0" borderId="0" xfId="0"
    applyNumberFormat="1"/>
</cellXfs>
```

Vysvětlení kódu

Atribut `numFmtId` v tagu `<xf>` obsahuje číslo 10, což znamená, že se bude jednat o procenta, která mají dvě desetinná místa, bude se aplikovat tento vzorec 0.00%.

4.2.9 TYP SPECIÁLNÍ

Tento typ se využívá pouze ve specifických případech.

Popis příkladu

	A	B
1		
2		334 02
3		33402
4		33333 33 33
5		333 33 33 33

Obrázek 20: Typ speciální

Buňky B2 až B5 obsahují různá čísla.

Složka s xml kódy: Typ_speciální

Styles.xml

Část zdrojového kódu

```
<numFmts count="4">
  <numFmt numFmtId="169" formatCode="000\ 00"/>
  <numFmt numFmtId="170" formatCode="00000"/>
  <numFmt numFmtId="171" formatCode=" [&lt;=99999]###\
  ##;##\ ##\ ##"/>
  <numFmt numFmtId="172" formatCode=" [&lt;=99999999]###\
  ##\ ##;##\ ##\ ##\ ##"/>
</numFmts>
```

Vysvětlení kódu

```
<numFmt numFmtId="169" formatCode="000\ 00"/>
```

Atribut `formatCode` nám definuje, jak se má číslo zobrazit, v tomto případě se číslo zobrazí 334 02. Tento typ se používá pro poštovní směrovací čísla.

```
<numFmt numFmtId="170" formatCode="00000"/>
```

Tento typ je určen také pro zápis poštovního směrovacího čísla dohromady bez mezer.

```
<numFmt numFmtId="171" formatCode="[&lt;=99999]###\ ##;##\ ##\ ##"/>
```

Číslo se nám zobrazí 33333 33 33. Poslední čtyři čísla se rozdělí po dvou, zbytek čísel zůstane bez mezer.

```
<numFmtnumFmtId="172"formatCode="[&lt;=9999999]###\ ##\ ##;##\ ##\ ##\ ##"/>
```

Druhý zápis telefonních čísel, kdy se nám rozdělí poslední šestičíslí opět po dvojicích, zbytek čísla zůstane opět dohromady.

4.2.10 TYP TEXT

Typ text patří mezi základní typ.

Popis příkladu

	A	B	C	D	E
1	Produkt	cena/ks	prodáno/ks	reklamováno	ztráta
2	Jar				
3	persil				
4	mýdlo				
5	barva na vlasy				
6	celkem				

Obrázek 21: Typ text

Vytvořili jsme si záhlaví tabulky.

Složka s xml kódy: Typ_text

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
  <row r="1" spans="1:5" x14ac:dyDescent="0.25">
    <c r="A1" s="1" t="s">
      <v>0</v>
    </c>
    <c r="B1" s="1" t="s">
      <v>1</v>
    </c>
    <c r="C1" s="1" t="s">
      <v>2</v>
    </c>
    <c r="D1" s="1" t="s">
      <v>3</v>
    </c>
  </row>
```

```

<c r="E1" s="1" t="s">
  <v>4</v>
</c>
</row>
<row r="2" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A2" s="1" t="s">
    <v>5</v>
  </c>
</row>
<row r="3" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A3" s="1" t="s">
    <v>7</v>
  </c>
</row>
<row r="4" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A4" s="1" t="s">
    <v>6</v>
  </c>
</row>
<row r="5" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A5" s="1" t="s">
    <v>8</v>
  </c>
</row>
<row r="6" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A6" s="1" t="s">
    <v>9</v>
  </c>
</row>
</sheetData>

```

Vysvětlení kódu

```

<row r="1" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A1" s="1" t="s">
    <v>0</v>
  </c>
  <c r="B1" s="1" t="s">
    <v>1</v>
  </c>
  <c r="C1" s="1" t="s">
    <v>2</v>
  </c>
  <c r="D1" s="1" t="s">
    <v>3</v>
  </c>
  <c r="E1" s="1" t="s">
    <v>4</v>
  </c>
</row>

```

Atribut `t` obsahuje hodnotu `s`, což znamená, že se bude jednat o text, nikoliv o čísla. Do elementu `<v>0</v>` se pak píše pořadí, ve kterém se mají obsahy buňky zobrazit.

```

<row r="2" spans="1:5" x14ac:dyDescent="0.25">
  <c r="A2" s="1" t="s">
    <v>5</v>
  </c>
</row>
<row r="3" spans="1:5" x14ac:dyDescent="0.25">

```

```

    <c r="A3" s="1" t="s">
      <v>7</v>
    </c>
  </row>
  <row r="4" spans="1:5" x14ac:dyDescent="0.25">
    <c r="A4" s="1" t="s">
      <v>6</v>
    </c>
  </row>

```

Naši domněnku jsme si ověřili v tomto kódu. Napřed jsme zapsali text Jar do buňky A2 a potom do buňky A3 mýdlo. Protože jsme zapomněli na persil, tak jsme mezi buňku A2 a buňku A3 vložili novou řádku. Došlo k přejmenování buněk, ale hodnota v elementu `<v></v>` zůstala stejná. Hodnota tohoto elementu nám určuje pořadí, v kterém se má do buněk zapisovat, nebo jsme zapisovali. Tyto hodnoty jsou velice důležité.

SharedStrings.xml

Zdrojový kód

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sst xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
count="10" uniqueCount="10">
  <si>
    <t>Produkt</t>
  </si>
  <si>
    <t>cena/ks</t>
  </si>
  <si>
    <t>prodáno/ks</t>
  </si>
  <si>
    <t>reklamováno</t>
  </si>
  <si>
    <t>ztráta</t>
  </si>
  <si>
    <t>Jar</t>
  </si>
  <si>
    <t>mýdlo</t>
  </si>
  <si>
    <t>persil</t>
  </si>
  <si>
    <t>barva na vlasy</t>
  </si>
  <si>
    <t>celkem</t>
  </si>
</sst>

```

Vysvětlení kódu

Tento soubor obsahuje text, který se má vložit do buněk v souboru sheet1.xml. Obsah elementu `<v></v>` v souboru sheet1.xml. nám určuje v souboru SharedStrings.xml pořadí elementu `<si></si>`, ze kterého se má uložit obsah do buněk v souboru sheet1.xml. Například v souboru sheet1.xml má buňka A3 hodnotu v elementu `<v></v>` 7, ze souboru SharedStrings.xml se dá text elementu `<si></si>`, který je v pořadí osmý, protože hodnoty v elementu `<v></v>` v souboru sheet1.xml jsou řazené od nuly, vloží se do buňky A3 text persil.

4.3 SLUČOVÁNÍ BUNĚK

Může se nám stát, že budeme potřebovat buňku se stejným obsahem do hlavičky tabulky pro dva sousedící sloupce, abychom se vyhli zbytečnému opakování, můžeme sloučit sousedící buňky v hlavičce.

4.3.1 SLOUČENÍ DVOU BUNĚK

Sloučení dvou buněk se velmi často využívá v tabulkách.

Obsah dokumentu: V buňkách C2 a D2 je zapsaný a zarovnaný na střed text Slučování buněk.

Složka s xml kódy: Slucovani_bunek_1-2

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
  <row r="2" spans="3:4" x14ac:dyDescent="0.25">
    <c r="C2" s="1" t="s">
      <v>0</v>
    </c>
    <c r="D2" s="1"/>
  </row>
</sheetData>
<mergeCells count="1">
  <mergeCell ref="C2:D2"/>
</mergeCells>
```

Vysvětlení kódu

V tagu `<mergeCell>` nám hodnota atributu `ref` říká, že jsou sloučené buňky C2 až D2.

V našem případě pouze buňka C2 a D2.

4.3.2 SLOUČENÍ TŘÍ BUNĚK

Existují případy, kdy se neobejdeme bez sloučení tří sousedících buněk.

Obsah dokumentu: Sloučili jsme a zarovnali na střed C2, D2 a E2 s textem Slučování buněk.

Složka s xml kódy: Slucovani_bunek_1-3

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
  <row r="2" spans="3:5" x14ac:dyDescent="0.25">
    <c r="C2" s="1" t="s">
      <v>0</v>
    </c>
    <c r="D2" s="1"/>
    <c r="E2" s="1"/>
  </row>
</sheetData>
<mergeCells count="1">
  <mergeCell ref="C2:E2"/>
</mergeCells>
```

Vysvětlení kódu

Hodnota jedna v atributu `count` v tagu `<mergeCells>` nám říká, kolikrát dojde ke sloučení buněk. Atribut `ref` v tagu `<mergeCell>` nám pak říká rozsah buněk, které mají být sloučené.

4.3.3 DVOJITÉ SLOUČENÍ

Pro ověření některých poznatků jsme vytvořili příklad slucovani_bunek_1-5.

Obsah dokumentu: Tentokrát jsme sloučili buňky C2, D2 a G2, H2.

Složka s xml kódy: Slucovani_bunek_1-5

Sheet1.xml

Část zdrojového kódu

```
<sheetData>
  <row r="2" spans="3:8">
    <c r="C2" s="1" t="s">
      <v>0</v>
    </c>
    <c r="D2" s="1"/>
    <c r="G2" s="1" t="s">
      <v>0</v>
    </c>
    <c r="H2" s="1"/>
  </row>
</sheetData>
```

```

    </row>
</sheetData>
<mergeCells count="2">
    <mergeCell ref="C2:D2"/>
    <mergeCell ref="G2:H2"/>
</mergeCells>

```

Vysvětlení kódu

Nyní si ověříme naše domněnky o tagu `<mergeCells>` a tagu `<mergeCell>`. Protože dojde ke dvěma sloučením, tak hodnota atributu `count` v tagu `<mergeCells>` bude dva. Pro každé sloučení bude pak jedna řádka `<mergeCell>`.

4.4 VZORCE

Vzorce nám mohou usnadnit vyplňování tabulek.

Popis příkladu:

	A	B	C	D	E	F	G
1	produkt	cena/ks	prodáno ks	reklamováno	prodáno	ztráta	výdělek
2	jar	30	11	2	=C2-D2	9	=B2*E2 270
3	persil	139	5	1	=C3-D3	4	=B3*E3 556
4	mýdlo	9	120	0	=C4-D4	120	=B4*E4 1080
5	barva na vlasy	159	22	2	=C5-D5	20	=B5*E5 3180
6	celkem						=G2+G3+G4+G5 5086

Obrázek 22: Vzorce

Buňky E2 až E5, F2 až F5 a G2 až G6 obsahují vzorce.

Složka s xml kódy: vzorce

Sheet1.xml

Část zdrojového kódu

```

<row>
    <c r="E2">
        <f>C2-D2</f>
        <v>9</v>
    </c>
    <c r="F2">
        <f>B2*D2</f>
        <v>60</v>
    </c>
    <c r="G2">
        <f>B2*E2</f>
        <v>270</v>
    </c>
</row>
<row r="3" spans="1:7" x14ac:dyDescent="0.25">
    ...
    <f t="shared" ref="E3:E5" si="0">C3-D3</f>
    <v>4</v>

```

```

    </c>
    <c r="F3">
      <f t="shared" ref="F3:F5" si="1">B3*D3</f>
      <v>139</v>
    </c>
    <c r="G3">
      <f t="shared" ref="G3:G5" si="2">B3*E3</f>
      <v>556</v>
    </c>
  </row>
  ...
  <row r="6" spans="1:7" x14ac:dyDescent="0.25">
    <c r="A6" t="s">
      <v>9</v>
    </c>
    <c r="G6">
      <f>G2+G3+G4+G5</f>
      <v>5086</v>
    </c>
  </row>

```

Vysvětlení kódu

```

<c r="E2">
  <f>C2-D2</f>
  <v>9</v>
</c>

```

Tag `<f>` nám sděluje, že obsahem buňky je vzorec. Do nitř elementu `<f></f>` se zapisuje vzorec. Excel si výsledek ukládá do následujícího elementu `<v></v>`, kvůli urychlení načítání. Tato část kódu znamená: vezmi buňku E2 a ulož do ní vzorec C2-D2, který má výsledek 9. Při ruční editaci xml souboru bychom si tedy měli vypočítat výsledek vzorce ručně, ovšem můžeme element `<v>9</v>` vynechat a po změně koncovky souboru ze zip na xlsx otevřít soubor. Excel si sám tento element doplní a zobrazí výsledek. Pokud bychom, ale v zipu v souboru sheet1.xml přepsali například pouze buňku C2 na C3 a soubor uložili bez aktualizace výsledné hodnoty a změnili koncovku zip na xlsx a soubor otevřeli, tak se nám zobrazí starý výsledek, který se změní až po poklepání na buňku, která byla změněna. Pokud máme složitější vzorce, vyplatí se spíše element `<v></v>` úplně vynechat a nechat výsledek po otevření propočítat Excelem.

```

<c r="F2">
  <f>B2*D2</f>
  <v>60</v>
</c>

```

Tento kód znamená: vezmi buňku a ulož do ní vzorec B2*D2, který má výsledek 60.

```

<c r="E3">
  <f t="shared" ref="E3:E5" si="0">C3-D3</f>
  <v>4</v>
</c>

```

V tagu `<f>` atribut `t` obsahuje hodnotu `shared`, která označuje sdílení, atribut `ref` nám pak definuje, jaké buňky budou sdílet obsah elementu `<f></f>`. Atribut `si` nám pak vytváří odkaz na tento kód.

```
<c r="E5">
  <f t="shared" si="0"/>
  <v>20</v>
</c>
```

Protože jsme si již v předchozím kódu nadefinovali, jaký vzorec se má vložit do elementu `<f></f>` stačí se pouze odkázat pomocí atributu `si`, který má stejnou hodnotu jako atribut `si` v buňce E3 na vzorec, jenž se má aplikovat na tuto řádku, jelikož vzorec neobsahuje žádné absolutní odkazy, upraví vzorec pro aktuální řádku.

4.5 FUNKCE

Funkce jsou vlastně složitější vzorce, které jsou předdefinované.

Popis příkladů

	A	B
1		číslo
2		135
3		34
4		2
5		6
6		8
7		33
8		246
9		345
10	průměr	101,125
11	počet	8
12	maximum	345
13	minimum	2
14	suma	809

Obrázek 23: Funkce

Buňky B10 až B14 obsahují různé funkce.

Složka s xml kódy: funkce

Sheet1.xml

Část zdrojového kódu

```

<row r="10" spans="1:2" x14ac:dyDescent="0.25">
  <c r="A10" t="s">
    <v>1</v>
  </c>
  <c r="B10">
    <f>AVERAGE (B2:B9) </f>
    <v>101.125</v>
  </c>
</row>
<row r="11" spans="1:2" x14ac:dyDescent="0.25">
  <c r="A11" t="s">
    <v>5</v>
  </c>
  <c r="B11">
    <f>COUNT (B1:B9) </f>
    <v>8</v>
  </c>
</row>
<row r="12" spans="1:2" x14ac:dyDescent="0.25">
  <c r="A12" t="s">
    <v>2</v>
  </c>
  <c r="B12">
    <f>MAX (B2:B9) </f>
    <v>345</v>
  </c>
</row>
<row r="13" spans="1:2" x14ac:dyDescent="0.25">
  <c r="A13" t="s">
    <v>3</v>
  </c>
  <c r="B13">
    <f>MIN (B2:B9) </f>
    <v>2</v>
  </c>
</row>
<row r="14" spans="1:2" x14ac:dyDescent="0.25">
  <c r="A14" t="s">
    <v>4</v>
  </c>
  <c r="B14">
    <f>SUM (B2:B9) </f>
    <v>809</v>
  </c>
</row>

```

Vysvětlení kódu

Opět používáme pro označení funkce počáteční tag `<f>` a koncový tag `</f>`, stejně jako u vzorců. Do elementu `<v></v>` opět vkládáme výsledek vzorce funkce v elementu `<f></f>`.

```
<f>AVERAGE (B2:B9) </f>
```

Takto se funkce vypočítá průměr z rozsahu, v našem případě jsou to buňky B2 až B9.

```
<f>COUNT (B1 :B9) </f>
```

Slovo COUNT nám definuje funkci počet. Z rozsahu, který si zvolíme, spočítá buňky obsahující čísla.

```
<f>MAX (B2 :B9) </f>
```

Tato funkce má za úkol ze zvolené oblasti vybrat největší číslo.

```
<f>MIN (B2 :B9) </f>
```

Funkce MIN hledá ve vybraném rozsahu nejmenší číslo.

```
<f>SUM (B2 :B9) </f>
```

Funkce SUM sečítá všechny buňky obsahující číslo ve vybrané oblasti.

CalcChain.xml

Část zdrojového kódu

```
<calcChain
xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main">
  <c r="B11" i="1" l="1"/>
  <c r="B13" i="1"/>
  <c r="B12" i="1"/>
  <c r="B14" i="1"/>
  <c r="B10" i="1"/>
</calcChain>
```

Vysvětlení

V tomto souboru nalezneme buňky, které obsahují vzorec nebo funkci, těmto buňkám se přidělí hodnota atributu `i` v tagu `<c>` 1.

4.6 PODMÍNĚNÉ FORMÁTOVÁNÍ

U rozsáhlých tabulek můžeme zvýraznit část textu pomocí podmíněného formátování pro přehlednost.

Popis příkladu

produkt	cena/ks	prodáno ks	reklamováno	prodáno	ztráta	výdělek
jar	30	11	2	9	60	270
persil	139	5	1	4	139	556
mýdlo	9	120	0	120	0	1080
barva na vlasy	159	22	2	20	318	3180
celkem						5086

Obrázek 24: Podmíněné formátování

Na rozsah C2 až C5 jsme aplikovali ikony, na D2 až D5 datovou škálu, na F2 až F5 zvýraznění buněk větších než 100 a G2 až G5 barevnou škálu.

Složka s xml kódy: podmínene formatovani

Sheet1.xml

Část zdrojového kódu

```
<conditionalFormatting sqref="G2:G5">
  <cfRule type="colorScale" priority="5">
    <colorScale>
      <cfvo type="min" val="0"/>
      <cfvo type="max" val="0"/>
      <color rgb="FFFF0000"/>
      <color rgb="FF00B050"/>
    </colorScale>
  </cfRule>
</conditionalFormatting>
<conditionalFormatting sqref="F2:F5">
  <cfRule type="cellIs" dxflId="0" priority="4"
operator="greaterThan">
  <formula>100</formula>
</cfRule>
</conditionalFormatting>
<conditionalFormatting sqref="C2:C5">
  <cfRule type="iconSet" priority="2">
    <iconSet>
      <cfvo type="num" val="0"/>
      <cfvo type="num" val="50"/>
      <cfvo type="num" val="100"/>
    </iconSet>
  </cfRule>
</conditionalFormatting>
<conditionalFormatting sqref="D2:D5">
  <cfRule type="dataBar" priority="1">
    <dataBar>
      <cfvo type="min" val="0"/>
      <cfvo type="max" val="0"/>
      <color rgb="FF638EC6"/>
    </dataBar>
    <extLst>
      <ext uri="{B025F937-C7B1-47D3-B67F-A62EFF666E3E}"
xmlns:x14="http://schemas.microsoft.com/office/spreadsheetml/20
09/9/main">
      <x14:id>{134EDF93-4C08-4709-8F7B-395306567B41}</x14:id>
      </ext>
    </extLst>
  </cfRule>
</conditionalFormatting>
```

Vysvětlení kódu

Atribut `sqref` v tagu `<conditionalFormatting>` nám vybírá buňky, na které se bude aplikovat pravidlo podmíněného formátování. Element `<cfRule></cfRule>` obsahuje definici typu a pravidel pro formátování.

```
<conditionalFormatting sqref="G2:G5">
  <cfRule type="colorScale" priority="5">
    <colorScale>
      <cfvo type="min" val="0"/>
      <cfvo type="max" val="0"/>
      <color rgb="FFFF0000"/>
      <color rgb="FF00B050"/>
    </colorScale>
  </cfRule>
</conditionalFormatting>
```

`<colorScale>` znamená, že se bude aplikovat formát barevné škály. Atribut `type` v tagu `<cfvo/>` určuje pravidlo, které se má aplikovat, atribut `rgb` v tagu `<color/>` nám volí barvu pro nejnižší a nejvyšší hodnotu, v našem případě se bude aplikovat barevná škála, kde nejnižší číslo bude mít barvu `FFFF0000` a nejvyšší číslo bude mít barvu `FF00B050`, ostatní čísla se budou formátovat RGB přechodem mezi zvolenými barvami.

```
<conditionalFormatting sqref="F2:F5">
  <cfRule type="cellIs" dxflId="0" priority="4"
    operator="greaterThan">
    <formula>100</formula>
  </cfRule>
</conditionalFormatting>
```

Tato část kódu nám značí, že se budou automaticky formátovány buňky F2 až F5. V tagu `<cfRule>` nám atribut `type` popisuje, že se bude jednat o buňky, které podle atributu `operator` mají být větší než hodnota obsažená v elementu `<formula></formula>`. V našem případě se formát červená barva aplikuje na buňky s čísly větší než 100.

```
<conditionalFormatting sqref="C2:C5">
  <cfRule type="iconSet" priority="2">
    <iconSet>
      <cfvo type="num" val="0"/>
      <cfvo type="num" val="50"/>
      <cfvo type="num" val="100"/>
    </iconSet>
  </cfRule>
</conditionalFormatting>
```

Tento zápis nám říká: aplikuj pravidlo na buňky C2 až C5. Budou se vkládat ikony semaforu, což nastavujeme pomocí hodnoty `iconSet` v atributu `type` v tagu `<cfRule>`, kde intervalu `<0;50)` se přidá červené světlo, číslům z intervalu `<50;100)` se přiřadí světlo žluté barvy a čísla větší než 100 budou mít barvu ikony světla zelenou. Atribut `type` obsahuje hodnotu vyjadřující typ, který může být číslo, procenta, vzorec nebo percentil. Atribut `val` nám pak určuje první číslo intervalu, pro který se má vložit příslušná barva. Obecně lze říct, že prvnímu tagu `<cfvo>` se nastaví červená barva, druhému tagu pak oranžová a třetímu zelená.


```

<conditionalFormatting sqref="D2:D5">
  <cfRule type="dataBar" priority="1">
    <dataBar>
      <cfvo type="min" val="0"/>
      <cfvo type="max" val="0"/>
      <color rgb="FF638EC6"/>
    </dataBar>
    <extLst>
      <ext uri="{B025F937-C7B1-47D3-B67F-A62EFF666E3E}"
        xmlns:x14="http://schemas.microsoft.com/office/spreadsheetml/20
        09/9/main">
        <x14:id>{134EDF93-4C08-4709-8F7B-395306567B41}</x14:id>
      </ext>
    </extLst>
  </cfRule>
</conditionalFormatting>

```

Hodnota "dataBar" atributu `type` v tagu `<cfRule>` označuje, že se bude aplikovat datová čára, největší číslo bude mít nejdelší čáru, jejíž barva je definovaná v tagu `<color>` v atributu `rgb`, a nejmenšímu číslu bude přidělena nejkratší.

Následující tabulka poskytuje přehled hodnot atributu `type` v tagu `<cfRule>`.

Type	Výsledek
colorScale	Odstupňovaná barevná škála
dataBar	Datová čára
cells	Zvýrazní hodnoty splňující podmínku
Formula	Zvýrazní hodnoty splňující podmínku určenou vzorcem
iconSet	Nastaví ikonky

Tabulka 6: Atribut type (11)

4.7 TABULKA

Tabulka představuje určitý druh formátování a přednastavenými funkcemi.

Popis příkladu

	A	B	C	D	E	F	G
1	produkt	cena/ks	prodáno ks	reklamováno	prodáno	ztráta	výdělek
2	jar	30	11	2	9	60	270
3	persil	139	5	1	4	139	556
4	mýdlo	9	120	0	120	0	1080
5	barva na vlasy	159	22	2	20	318	3180
6	Celkem						5086

Na buňku z příkladu Vzorce jsme aplikovali formátování tabulky.

Složka s xml kódy: tabulka

Table1.xml

Část zdrojového kódu

```
<tableColumns count="7">
  <tableColumn id="1" name="produkt" totalsRowLabel="Celkem"/>
  <tableColumn id="2" name="cena/ks"/>
  <tableColumn id="3" name="prodáno ks"/>
  <tableColumn id="4" name="reklamováno"/>
  <tableColumn id="5" name="prodáno"/>
  <tableColumn id="6" name="ztráta"/>
  <tableColumn id="7" name="výdělek" totalsRowFunction="sum"/>
</tableColumns>
```

Vysvětlení

V tomto souboru nalezneme záhlaví tabulky a řádek souhrnů.

```
<tableColumn id="1" name="produkt" totalsRowLabel="Celkem"/>
```

První sloupec obsahuje záhlaví s textem produkt a souhrn s textem celkem. Atribut `id` nám značí číslo sloupce, v atributu `name` pak nalezneme text záhlaví a atribut `totalsRowLabel` nám definuje jméno, ke kterému se váže funkce ve sloupci 7.

```
<tableColumn id="7" name="výdělek" totalsRowFunction="sum"/>
```

V sedmém sloupci se nachází záhlaví s textem výdělek a souhrn s funkcí `sum`.

CalcChain.xml

Část zdrojového kódu

```
<calcChain
xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main">
  <c r="F3" i="1" l="1"/>
  <c r="F4" i="1"/>
  <c r="F5" i="1"/>
  <c r="F2" i="1"/>
  <c r="E3" i="1"/>
  <c r="G3" i="1" s="1"/>
  <c r="E4" i="1"/>
  <c r="G4" i="1" s="1"/>
  <c r="E5" i="1"/>
  <c r="G5" i="1" s="1"/>
  <c r="E2" i="1"/>
  <c r="G2" i="1" s="1"/>
  <c r="G6" i="1" s="1"/>
</calcChain>
```

Vysvětlení

V tomto souboru jsou nadefinované buňky obsahující vzorec či funkci.

Sheet1.xml

Část zdrojového kódu

```
<c r="G6">
  <f>SUBTOTAL(109, Tabulka1 [výdělek]) </f>
  <v>5086</v>
</c>
```

Vysvětlení kódu

Protože používáme tabulku, můžeme využít funkce, které tabulky nabízí. Tyto funkce voláme pomocí slovíčka SUBTOTAL, následuje závorka, do které se píše číslo funkce, v našem případě je to SUMA, jenž má číslo 109, následuje odkaz jedna a ve složených závorkách odkaz dvě. Vypočte se nám součet všech buněk ve sloupci výdělek.

4.8 GRAFY

Grafy slouží k obrazové interpretaci dat. Z důvodu délky zdrojových kódů naleznete celé toto kódy z kapitoly Grafy v přílohách na konci této práce.

Popis příkladů

Grafy: Použili jsme tabulku z příkladu Vzorce a vložili jsme sloupcový, výšečový a plošný graf.

Složka s xml kódy: grafy

Chart1.xml

V tomto souboru nalezeme obsah sloupcového grafu s názvem Výdělek.

```
<c:tx>
  <c:strRef>
    <c:f>List1!$A$2</c:f>
    <c:strCache>
      <c:ptCount val="1"/>
      <c:pt idx="0">
        <c:v>jar</c:v>
      </c:pt>
    </c:strCache>
  </c:strRef>
</c:tx>
```

Element `<c:tx></c:tx>` zahrnuje element `<c:f></c:f>`, kde se nachází odkaz na buňku, která obsahuje první popisek, jenž má být vložen do legendy grafu. V elementu `<c:v></c:v>` pak nalezeme přímo obsah této buňky.

```

<c:cat>
  <c:strRef>
    <c:f>List1!$G$1</c:f>
    <c:strCache>
      <c:ptCount val="1"/>
      <c:pt idx="0">
        <c:v>výdělek</c:v>
      </c:pt>
    </c:strCache>
  </c:strRef>
</c:cat>

```

V Elementu `<c:cat></c:cat>` nalezneme element `<c:f></c:f>`, kde je odkaz na buňku, která definuje název sloupce v grafu, v našem případě je to název grafu, protože v grafu figurují pouze hodnoty z jednoho sloupce. Element `<c:v></c:v>` pak v sobě zahrnuje přímo obsah této buňky.

```

<c:numRef>
  <c:f>List1!$G$2</c:f>
  <c:numCache>
    <c:formatCode>General</c:formatCode>
    <c:ptCount val="1"/>
    <c:pt idx="0">
      <c:v>270</c:v>
    </c:pt>
  </c:numCache>
</c:numRef>

```

Element `<c:numRef></c:numRef>` obsahuje element `<c:f></c:f>`, ve kterém se nachází odkaz na buňku obsahující první hodnotu, kterou má vyobrazit jeden sloupec. V elementu `<c:v>270</c:v>` pak nalezneme přímo obsah této buňky.

Všechny tři elementy, `<c:tx></c:tx>`, `<c:cat></c:cat>` a `<c:numRef></c:numRef>` se nám opakují tolikrát kolik je sloupců v grafu.

Chart2.xml

V tomto souboru nalezneme obsah výšečového grafu s názvem Reklamováno.

```

<c:tx>
  <c:strRef>
    <c:f>List1!$D$1</c:f>
    <c:strCache>
      <c:ptCount val="1"/>
      <c:pt idx="0">
        <c:v>reklamováno</c:v>
      </c:pt>
    </c:strCache>
  </c:strRef>
</c:tx>

```

Element `<c:tx></c:tx>` obsahuje element `<c:f></c:f>`, kde se nachází odkaz na buňku, ze které se má vložit titulek. V elementu `<c:v></c:v>` pak najdeme přímo obsah této buňky.

```
<c:cat>
  <c:strRef>
    <c:f>List1!$A$2:$A$5</c:f>
    <c:strCache>
      <c:ptCount val="4"/>
      <c:pt idx="0">
        <c:v>jar</c:v>
      </c:pt>
      <c:pt idx="1">
        <c:v>persil</c:v>
      </c:pt>
      <c:pt idx="2">
        <c:v>mýdlo</c:v>
      </c:pt>
      <c:pt idx="3">
        <c:v>barva na vlasy</c:v>
      </c:pt>
    </c:strCache>
  </c:strRef>
</c:cat>
```

V elementu `<c:cat></c:cat>` je pak definován rozsah buněk, ze kterého se mají vzít informace do legendy, které jsou slovně vyjádřeny v elementu `<c:v></c:v>`.

```
<c:val>
  <c:numRef>
    <c:f>List1!$D$2:$D$5</c:f>
    <c:numCache>
      <c:formatCode>General</c:formatCode>
      <c:ptCount val="4"/>
      <c:pt idx="0">
        <c:v>2</c:v>
      </c:pt>
      <c:pt idx="1">
        <c:v>1</c:v>
      </c:pt>
      <c:pt idx="2">
        <c:v>0</c:v>
      </c:pt>
      <c:pt idx="3">
        <c:v>2</c:v>
      </c:pt>
    </c:numCache>
  </c:numRef>
</c:val>
```

V elementu `<c:val></c:val>` pak nalezneme odkazy na data a jejich hodnoty, ze kterých se má vytvořit graf.

Chart3.xml

V tomto souboru nalezneme obsah plošného grafu s názvem Ztráta.

Zápis pro obsah grafu plošného je obdobný jako u grafu výsečového, proto není potřeba jednotlivé elementy a jejich obsah rozebírat.

Drawing1.xml

Tento soubor obsahuje informace o oblasti grafu.

```
<xdr:from>
  <xdr:col>8</xdr:col>
  <xdr:colOff>314325</xdr:colOff>
  <xdr:row>0</xdr:row>
  <xdr:rowOff>47625</xdr:rowOff>
</xdr:from>
```

Element `<xdr:from></xdr:from>` obsahuje vymezení počátečního bodu grafu. Z elementu `<xdr:col></xdr:col>` můžeme odvodit číslo sloupce, ve kterém se nachází počáteční bod. Abychom zjistili číslo sloupce, musíme vzít obsah elementu `<xdr:col></xdr:col>`, v našem případě číslo 8, a přičíst k němu jedničku, protože se v xml souboru začíná číslovat od čísla 0, ale v Excelu se začíná od čísla 1. Z obsahu elementu `<xdr:row></xdr:row>` pak můžeme odvodit číslo řádky, kde se nachází počáteční bod. Postup je podobný vezmeme obsah elementu `<xdr:row></xdr:row>` a přičteme k němu hodnotu jedna, ze stejného důvodu jako předtím.

```
<xdr:to>
  <xdr:col>16</xdr:col>
  <xdr:colOff>9525</xdr:colOff>
  <xdr:row>14</xdr:row>
  <xdr:rowOff>123825</xdr:rowOff>
</xdr:to>
```

Element `<xdr:to></xdr:to>` vymezuje koncový bodu grafu. Postup pro zjištění koncového řádku a sloupce naleznete v předchozím odstavci.

ZÁVĚR

Z první části, která obsahovala důvody vzniku tohoto standardu, si budoucí i stávající kodéři či programátoři uvědomili, jak důležité je, aby byla data tzv. „stálá“.

Další úsek nám pak poskytnul informace o základních pravidlech pro zápis XML kódů, bez kterých se při používání jazyka XML neobejdeme. Tyto pravidla je důležité dodržet, aby nám překladač kódu nehlásil chybu.

Následoval způsob řazení a organizace v souboru zip a popis souborů v něm obsažených, jenž kromě jiného popsal i význam jednotlivých složek a souborů zipu.

Poslední nejobsáhlejší kapitola, popisovala přehled základních operací v Excelu, díky nimž jsme se naučili vytvářet dokument Excelu pomocí jazyku XML.

Cílem této práce bylo poskytnout přehledný náhled do struktury a významných elementů Office OpenXML dokumentu pro MS Excel 2010. Myslíme, že se nám stanovený úkol podařilo splnit.

RESUMÉ

MS Excel 2010 use file format which is based on XML language called OpenXML. This Bachelor thesis contains informations about basics of XML code and how to use it. Basic rules for writing of XML code warning before possible fail. Hierarchical structure of the ZIP file can show you the content of each folders. There are a few basic XML elements and examples in my Bachelor thesis.

Keywords:

XML, tag, element, MS Office, Excel, atribut

POUŽITÁ LITERATURA

1. MATĚNA, Roman. *XML a značkovací jazyky, historie a vznik: Webfaq.cz*. Webfaq.cz [online]. 2008 [cit. 2014-02-21]. Dostupné z: <http://www.webfaq.cz/clanek/XML-a-znacikovaci-jazyky-historie-a-vznik>
2. KOSEK, Jiří. *XML: staronový formát. Domovská stránka Jirky Koska: VŠE O WWW* [online]. 1999 [cit. 2014-01-22]. Dostupné z: <http://www.kosek.cz/clanky/xml/xml-historie.html>
3. KOSEK, Jiří. *SGML: Standard Generalized Markup Language. Domovská stránka Jirky Koska: VŠE O WWW* [online]. 1999 [cit. 2014-01-23]. Dostupné z: <http://www.kosek.cz/clanky/cw/sgml.html>
4. BRADLEY, Neil. *XML kompletní průvodce*. Edinburg Gate: Addison Wesley, 2000, s. 18-27. ISBN 80-7169-949-7.
5. GRUSOVÁ, Lucie. *XML pro úplné začátečníky*. Praha 4: Computer Press, 2002, s. 1-71. ISBN 80-7226-697-7.
6. KOSEK, Jiří. *Sekce CDATA. Domovská stránka Jirky Koska: VŠE O WWW* [online]. 1999 [cit. 2014-04-5]. Dostupné z: <http://www.kosek.cz/clanky/swn-xml/ar03s43.html>
7. PECINOVSKÝ, Josef a Rudolf PECINOVSKÝ. *Office 2010 podrobný průvodce*. Praha 7: Grada Publishing, 2001. ISBN 978-80-247-3620-4.
8. Understanding [Content_Types].xml: Office Open XML I: Exploring the Office Open XML Formats. *Office* [online]. 2014 [cit. 2014-01-12]. Dostupné z: <http://office.microsoft.com/en-us/office-open-xml-i-exploring-the-office-open-xml-formats-RZ010243529.aspx?section=16>
9. [MS-OI29500]: Part 1 Section 22.2.2.7, DocSecurity (Document Security). *MSDN-the microsoft developer network* [online]. 2014 [cit. 2014-02-20]. Dostupné z: <http://msdn.microsoft.com/en-us/library/ff535413%28v=office.12%29.aspx>
10. ScaleCrop Class (DocumentFormat.OpenXml.ExtendedProperties). *MSDN-the microsoft developer network* [online]. 2014 [cit. 2014-02-11]. Dostupné z: <http://msdn.microsoft.com/en->

us/library/documentformat.openxml.extendedproperties.scalecrop%28v=office.14%29.aspx

11. Office Open XML *File Formats: Part 4* [online]. 2011 [cit. 2014-03-12]. Dostupné z: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-376,%20Third%20Edition,%20Part%204%20-%20Transitional%20Migration%20Features.zip>

12. LinksUpToDate Class: DocumentFormat.OpenXml.ExtendedProperties. *MSDN-the microsoft developer network* [online]. 2014 [cit. 2014-02-11]. Dostupné z: <http://msdn.microsoft.com/en-us/library/documentformat.openxml.extendedproperties.linksuptodate%28v=office.14%29.aspx>

13. Excel Versions. *Pearson Software Consulting, LLC, Comprehensive Excel Information* [online]. 2013 [cit. 2014-02-15]. Dostupné z: <http://www.cpearson.com/excel/versions.htm>

14. Format a date the way you want: Excel Office.com. *Office* [online]. 2013 [cit. 2014-02-11]. Dostupné z: <http://office.microsoft.com/en-us/excel-help/format-a-date-the-way-you-want-HA102809474.aspx>

15. SharedDocument Class (DocumentFormat.OpenXml.ExtendedProperties). *MSDN-the microsoft developer network* [online]. 2014 [cit. 2014-02-11]. Dostupné z: <http://msdn.microsoft.com/en-us/library/documentformat.openxml.extendedproperties.shareddocument%28v=office.14%29.aspx>

16. Properties.HyperlinksChanged Property (DocumentFormat.OpenXml.ExtendedProperties). *MSDN-the microsoft developer network* [online]. 2014 [cit. 2014-02-11]. Dostupné z: <http://msdn.microsoft.com/en-us/library/documentformat.openxml.extendedproperties.properties.hyperlinkschanged%28v=office.14%29.aspx>

SEZNAM OBRÁZKŮ

Obrázek 1: Vývoj	3
Obrázek 2: Základní problematika elementu.....	5
Obrázek 3: Chybné křížení elementů	7
Obrázek 4: Správný kód bez křížení elementů	7
Obrázek 5: Obsah souboru zip.....	10
Obrázek 6: Obsah složky Xl.....	11
Obrázek 7: Složka drawings	11
Obrázek 8: Složka media	12
Obrázek 9: Složka charts	12
Obrázek 10: Složka diagrams	12
Obrázek 11: Složka tables	13
Obrázek 12: Obsah složky worksheets	13
Obrázek 13: Obecný typ	23
Obrázek 14: Typ datum	25
Obrázek 15: Typ čas	27
Obrázek 16: Typ číslo.....	29
Obrázek 17: Typ matematický	30
Obrázek 18: Typ měna	30
Obrázek 19: Typ procenta	31
Obrázek 20: Typ speciální	32
Obrázek 21: Typ text	33
Obrázek 22: Vzorce	38
Obrázek 23: Funkce.....	40
Obrázek 24: Podmíněné formátování	42

SEZNAM TABULEK

Tabulka 1: Zakázané znaky	9
Tabulka 2: Číselné označení verze Excelu (13)	17
Tabulka 3: Den (11).....	26
Tabulka 4: Měsíc (11).....	27
Tabulka 5: Rok (11).....	27
Tabulka 6: Atribut type (11).....	45

PŘÍLOHY

Chart1.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:chartSpace
xmlns:c="http://schemas.openxmlformats.org/drawingml/2006/chart"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
  <c:date1904 val="0"/>
  <c:lang val="cs-CZ"/>
  <c:roundedCorners val="0"/>
  <mc:AlternateContent
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
    <mc:Choice Requires="c14"
xmlns:c14="http://schemas.microsoft.com/office/drawing/2007/8/2/chart">
      <c14:style val="102"/>
    </mc:Choice>
    <mc:Fallback>
      <c:style val="2"/>
    </mc:Fallback>
  </mc:AlternateContent>
  <c:chart>
    <c:autoTitleDeleted val="0"/>
    <c:plotArea>
      <c:layout/>
      <c:barChart>
        <c:barDir val="col"/>
        <c:grouping val="clustered"/>
        <c:varyColors val="0"/>
        <c:ser>
          <c:idx val="0"/>
          <c:order val="0"/>
          <c:tx>
            <c:strRef>
              <c:f>List1!$A$2</c:f>
              <c:strCache>
                <c:ptCount val="1"/>
                <c:pt idx="0">
                  <c:v>jar</c:v>
                </c:pt>
              </c:strCache>
            </c:strRef>
          </c:tx>
          <c:invertIfNegative val="0"/>
          <c:cat>
            <c:strRef>
              <c:f>List1!$G$1</c:f>
              <c:strCache>
                <c:ptCount val="1"/>
                <c:pt idx="0">
                  <c:v>výdělek</c:v>
                </c:pt>
              </c:strCache>
            </c:strRef>
          </c:cat>
          <c:val>
            <c:numRef>

```

```

    <c:f>List1!$G$2</c:f>
    <c:numCache>
      <c:formatCode>General</c:formatCode>
      <c:ptCount val="1"/>
      <c:pt idx="0">
        <c:v>270</c:v>
      </c:pt>
    </c:numCache>
  </c:numRef>
</c:val>
</c:ser>
<c:ser>
  <c:idx val="1"/>
  <c:order val="1"/>
  <c:tx>
    <c:strRef>
      <c:f>List1!$A$3</c:f>
      <c:strCache>
        <c:ptCount val="1"/>
        <c:pt idx="0">
          <c:v>persil</c:v>
        </c:pt>
      </c:strCache>
    </c:strRef>
  </c:tx>
  <c:invertIfNegative val="0"/>
  <c:cat>
    <c:strRef>
      <c:f>List1!$G$1</c:f>
      <c:strCache>
        <c:ptCount val="1"/>
        <c:pt idx="0">
          <c:v>výdělek</c:v>
        </c:pt>
      </c:strCache>
    </c:strRef>
  </c:cat>
  <c:val>
    <c:numRef>
      <c:f>List1!$G$3</c:f>
      <c:numCache>
        <c:formatCode>General</c:formatCode>
        <c:ptCount val="1"/>
        <c:pt idx="0">
          <c:v>556</c:v>
        </c:pt>
      </c:numCache>
    </c:numRef>
  </c:val>
</c:ser>
<c:ser>
  <c:idx val="2"/>
  <c:order val="2"/>
  <c:tx>
    <c:strRef>
      <c:f>List1!$A$4</c:f>
      <c:strCache>
        <c:ptCount val="1"/>
        <c:pt idx="0">
          <c:v>mýdlo</c:v>
        </c:pt>
      </c:strCache>
    </c:strRef>
  </c:tx>

```

```

        </c:pt>
    </c:strCache>
</c:strRef>
</c:tx>
<c:invertIfNegative val="0"/>
<c:cat>
    <c:strRef>
        <c:f>List1!$G$1</c:f>
        <c:strCache>
            <c:ptCount val="1"/>
            <c:pt idx="0">
                <c:v>výdělek</c:v>
            </c:pt>
        </c:strCache>
    </c:strRef>
</c:cat>
<c:val>
    <c:numRef>
        <c:f>List1!$G$4</c:f>
        <c:numCache>
            <c:formatCode>General</c:formatCode>
            <c:ptCount val="1"/>
            <c:pt idx="0">
                <c:v>1080</c:v>
            </c:pt>
        </c:numCache>
    </c:numRef>
</c:val>
</c:ser>
<c:ser>
    <c:idx val="3"/>
    <c:order val="3"/>
    <c:tx>
        <c:strRef>
            <c:f>List1!$A$5</c:f>
            <c:strCache>
                <c:ptCount val="1"/>
                <c:pt idx="0">
                    <c:v>barva na vlasy</c:v>
                </c:pt>
            </c:strCache>
        </c:strRef>
    </c:tx>
    <c:invertIfNegative val="0"/>
    <c:cat>
        <c:strRef>
            <c:f>List1!$G$1</c:f>
            <c:strCache>
                <c:ptCount val="1"/>
                <c:pt idx="0">
                    <c:v>výdělek</c:v>
                </c:pt>
            </c:strCache>
        </c:strRef>
    </c:cat>
    <c:val>
        <c:numRef>
            <c:f>List1!$G$5</c:f>
            <c:numCache>
                <c:formatCode>General</c:formatCode>

```

```

        <c:ptCount val="1"/>
        <c:pt idx="0">
            <c:v>3180</c:v>
        </c:pt>
    </c:numCache>
</c:numRef>
</c:val>
</c:ser>
<c:dLbls>
    <c:showLegendKey val="0"/>
    <c:showVal val="0"/>
    <c:showCatName val="0"/>
    <c:showSerName val="0"/>
    <c:showPercent val="0"/>
    <c:showBubbleSize val="0"/>
</c:dLbls>
<c:gapWidth val="150"/>
<c:axId val="74465280"/>
<c:axId val="74466816"/>
</c:barChart>
<c:catAx>
    <c:axId val="74465280"/>
    <c:scaling>
        <c:orientation val="minMax"/>
    </c:scaling>
    <c:delete val="0"/>
    <c:axPos val="b"/>
    <c:majorTickMark val="out"/>
    <c:minorTickMark val="none"/>
    <c:tickLblPos val="nextTo"/>
    <c:crossAx val="74466816"/>
    <c:crosses val="autoZero"/>
    <c:auto val="1"/>
    <c:lblAlgn val="ctr"/>
    <c:lblOffset val="100"/>
    <c:noMultiLvlLbl val="0"/>
</c:catAx>
<c:valAx>
    <c:axId val="74466816"/>
    <c:scaling>
        <c:orientation val="minMax"/>
    </c:scaling>
    <c:delete val="0"/>
    <c:axPos val="1"/>
    <c:majorGridlines/>
    <c:numFmt formatCode="General" sourceLinked="1"/>
    <c:majorTickMark val="out"/>
    <c:minorTickMark val="none"/>
    <c:tickLblPos val="nextTo"/>
    <c:crossAx val="74465280"/>
    <c:crosses val="autoZero"/>
    <c:crossBetween val="between"/>
</c:valAx>
</c:plotArea>
<c:legend>
    <c:legendPos val="r"/>
    <c:layout/>
    <c:overlay val="0"/>
</c:legend>
<c:plotVisOnly val="1"/>

```



```

    <c:dispBlanksAs val="gap"/>
    <c:showDLblsOverMax val="0"/>
  </c:chart>
  <c:printSettings>
    <c:headerFooter/>
    <c:pageMargins b="0.787401574999999996" l="0.7" r="0.7"
t="0.787401574999999996" header="0.3" footer="0.3"/>
    <c:pageSetup/>
  </c:printSettings>
</c:chartSpace>

```

Chart2.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:chartSpace
xmlns:c="http://schemas.openxmlformats.org/drawingml/2006/chart"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
>
  <c:date1904 val="0"/>
  <c:lang val="cs-CZ"/>
  <c:roundedCorners val="0"/>
  <mc:AlternateContent
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
    <mc:Choice Requires="c14"
xmlns:c14="http://schemas.microsoft.com/office/drawing/2007/8/2/chart">
      <c14:style val="102"/>
    </mc:Choice>
    <mc:Fallback>
      <c:style val="2"/>
    </mc:Fallback>
  </mc:AlternateContent>
  <c:chart>
    <c:title>
      <c:layout/>
      <c:overlay val="0"/>
    </c:title>
    <c:autoTitleDeleted val="0"/>
    <c:plotArea>
      <c:layout/>
      <c:pieChart>
        <c:varyColors val="1"/>
        <c:ser>
          <c:idx val="0"/>
          <c:order val="0"/>
          <c:tx>
            <c:strRef>
              <c:f>List1!$D$1</c:f>
              <c:strCache>
                <c:ptCount val="1"/>
                <c:pt idx="0">
                  <c:v>reklamováno</c:v>
                </c:pt>
              </c:strCache>
            </c:strRef>
          </c:tx>
          <c:cat>
            <c:strRef>
              <c:f>List1!$A$2:$A$5</c:f>
              <c:strCache>

```

```

        <c:ptCount val="4"/>
        <c:pt idx="0">
            <c:v>jar</c:v>
        </c:pt>
        <c:pt idx="1">
            <c:v>persil</c:v>
        </c:pt>
        <c:pt idx="2">
            <c:v>mýdlo</c:v>
        </c:pt>
        <c:pt idx="3">
            <c:v>barva na vlasy</c:v>
        </c:pt>
    </c:strCache>
</c:strRef>
</c:cat>
<c:val>
    <c:numRef>
        <c:f>List1!$D$2:$D$5</c:f>
        <c:numCache>
            <c:formatCode>General</c:formatCode>
            <c:ptCount val="4"/>
            <c:pt idx="0">
                <c:v>2</c:v>
            </c:pt>
            <c:pt idx="1">
                <c:v>1</c:v>
            </c:pt>
            <c:pt idx="2">
                <c:v>0</c:v>
            </c:pt>
            <c:pt idx="3">
                <c:v>2</c:v>
            </c:pt>
        </c:numCache>
    </c:numRef>
</c:val>
</c:ser>
<c:dLbls>
    <c:showLegendKey val="0"/>
    <c:showVal val="0"/>
    <c:showCatName val="0"/>
    <c:showSerName val="0"/>
    <c:showPercent val="0"/>
    <c:showBubbleSize val="0"/>
    <c:showLeaderLines val="1"/>
</c:dLbls>
    <c:firstSliceAng val="0"/>
</c:pieChart>
</c:plotArea>
<c:legend>
    <c:legendPos val="r"/>
    <c:layout/>
    <c:overlay val="0"/>
    <c:txPr>
        <a:bodyPr/>
        <a:lstStyle/>
        <a:p>
            <a:pPr rtl="0">
                <a:defRPr/>

```

```

        </a:pPr>
        <a:endParaRPr lang="cs-CZ"/>
    </a:p>
    </c:txPr>
</c:legend>
<c:plotVisOnly val="1"/>
<c:dispBlanksAs val="gap"/>
<c:showDLblsOverMax val="0"/>
</c:chart>
<c:printSettings>
    <c:headerFooter/>
    <c:pageMargins b="0.78740157499999996" l="0.7" r="0.7"
t="0.78740157499999996" header="0.3" footer="0.3"/>
    <c:pageSetup/>
</c:printSettings>
</c:chartSpace>

```

Chart3.xml

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:chartSpace
xmlns:c="http://schemas.openxmlformats.org/drawingml/2006/chart"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
>
    <c:date1904 val="0"/>
    <c:lang val="cs-CZ"/>
    <c:roundedCorners val="0"/>
    <mc:AlternateContent
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
        <mc:Choice Requires="c14"
xmlns:c14="http://schemas.microsoft.com/office/drawing/2007/8/2/chart">
            <c14:style val="102"/>
        </mc:Choice>
        <mc:Fallback>
            <c:style val="2"/>
        </mc:Fallback>
    </mc:AlternateContent>
    <c:chart>
        <c:title>
            <c:layout/>
            <c:overlay val="0"/>
        </c:title>
        <c:autoTitleDeleted val="0"/>
        <c:view3D>
            <c:rotX val="15"/>
            <c:rotY val="20"/>
            <c:rAngAx val="0"/>
            <c:perspective val="30"/>
        </c:view3D>
        <c:floor>
            <c:thickness val="0"/>
        </c:floor>
        <c:sideWall>
            <c:thickness val="0"/>
        </c:sideWall>
        <c:backWall>
            <c:thickness val="0"/>
        </c:backWall>
    </c:chart>
</c:plotArea>

```

```

<c:layout/>
<c:area3DChart>
  <c:grouping val="stacked"/>
  <c:varyColors val="0"/>
  <c:ser>
    <c:idx val="0"/>
    <c:order val="0"/>
    <c:tx>
      <c:strRef>
        <c:f>List1!$F$1</c:f>
        <c:strCache>
          <c:ptCount val="1"/>
          <c:pt idx="0">
            <c:v>ztráta</c:v>
          </c:pt>
        </c:strCache>
      </c:strRef>
    </c:tx>
    <c:cat>
      <c:strRef>
        <c:f>List1!$A$2:$A$5</c:f>
        <c:strCache>
          <c:ptCount val="4"/>
          <c:pt idx="0">
            <c:v>jar</c:v>
          </c:pt>
          <c:pt idx="1">
            <c:v>persil</c:v>
          </c:pt>
          <c:pt idx="2">
            <c:v>mýdlo</c:v>
          </c:pt>
          <c:pt idx="3">
            <c:v>barva na vlasy</c:v>
          </c:pt>
        </c:strCache>
      </c:strRef>
    </c:cat>
    <c:val>
      <c:numRef>
        <c:f>List1!$F$2:$F$5</c:f>
        <c:numCache>
          <c:formatCode>General</c:formatCode>
          <c:ptCount val="4"/>
          <c:pt idx="0">
            <c:v>60</c:v>
          </c:pt>
          <c:pt idx="1">
            <c:v>139</c:v>
          </c:pt>
          <c:pt idx="2">
            <c:v>0</c:v>
          </c:pt>
          <c:pt idx="3">
            <c:v>318</c:v>
          </c:pt>
        </c:numCache>
      </c:numRef>
    </c:val>
  </c:ser>

```

```

<c:dLbls>
  <c:showLegendKey val="0"/>
  <c:showVal val="0"/>
  <c:showCatName val="0"/>
  <c:showSerName val="0"/>
  <c:showPercent val="0"/>
  <c:showBubbleSize val="0"/>
</c:dLbls><c:axId val="74493952"/>
<c:axId val="74495488"/>
<c:axId val="0"/>
</c:area3DChart>
<c:catAx>
  <c:axId val="74493952"/>
  <c:scaling>
    <c:orientation val="minMax"/>
  </c:scaling>
  <c:delete val="0"/>
  <c:axPos val="b"/>
  <c:majorTickMark val="out"/>
  <c:minorTickMark val="none"/>
  <c:tickLblPos val="nextTo"/>
  <c:crossAx val="74495488"/>
  <c:crosses val="autoZero"/>
  <c:auto val="1"/>
  <c:lblAlgn val="ctr"/>
  <c:lblOffset val="100"/>
  <c:noMultiLvlLbl val="0"/>
</c:catAx>
<c:valAx>
  <c:axId val="74495488"/>
  <c:scaling>
    <c:orientation val="minMax"/>
  </c:scaling>
  <c:delete val="0"/>
  <c:axPos val="1"/>
  <c:majorGridlines/>
  <c:numFmt formatCode="General" sourceLinked="1"/>
  <c:majorTickMark val="out"/>
  <c:minorTickMark val="none"/>
  <c:tickLblPos val="nextTo"/>
  <c:crossAx val="74493952"/>
  <c:crosses val="autoZero"/>
  <c:crossBetween val="midCat"/>
</c:valAx>
</c:plotArea>
<c:legend>
  <c:legendPos val="r"/>
  <c:layout/>
  <c:overlay val="0"/>
</c:legend>
<c:plotVisOnly val="1"/>
<c:dispBlanksAs val="zero"/>
<c:showDLblsOverMax val="0"/>
</c:chart>
<c:printSettings>
  <c:headerFooter/>
  <c:pageMargins b="0.78740157499999996" l="0.7" r="0.7"
t="0.78740157499999996" header="0.3" footer="0.3"/>
  <c:pageSetup/>
</c:printSettings>

```

```
</c:chartSpace>
```

Drawing1.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xdr:wsDr
xmlns:xdr="http://schemas.openxmlformats.org/drawingml/2006/spreadsheetDr
awing"
xmlns:a="http://schemas.openxmlformats.org/drawingml/2006/main">
  <xdr:twoCellAnchor>
    <xdr:from>
      <xdr:col>8</xdr:col>
      <xdr:colOff>314325</xdr:colOff>
      <xdr:row>0</xdr:row>
      <xdr:rowOff>47625</xdr:rowOff>
    </xdr:from>
    <xdr:to>
      <xdr:col>16</xdr:col>
      <xdr:colOff>9525</xdr:colOff>
      <xdr:row>14</xdr:row>
      <xdr:rowOff>123825</xdr:rowOff>
    </xdr:to>
    <xdr:graphicFrame macro="">
      <xdr:nvGraphicFramePr>
        <xdr:cNvPr id="2" name="Graf 1"/>
        <xdr:cNvGraphicFramePr/>
      </xdr:nvGraphicFramePr>
      <xdr:xfrm>
        <a:off x="0" y="0"/>
        <a:ext cx="0" cy="0"/>
      </xdr:xfrm>
      <a:graphic>
        <a:graphicData
uri="http://schemas.openxmlformats.org/drawingml/2006/chart">
          <c:chart
xmlns:c="http://schemas.openxmlformats.org/drawingml/2006/chart"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationsh
ips" r:id="rId1"/>
            </a:graphicData>
          </a:graphic>
        </xdr:graphicFrame>
        <xdr:clientData/>
      </xdr:twoCellAnchor>
    <xdr:twoCellAnchor>
      <xdr:from>
        <xdr:col>0</xdr:col>
        <xdr:colOff>38100</xdr:colOff>
        <xdr:row>6</xdr:row>
        <xdr:rowOff>57150</xdr:rowOff>
      </xdr:from>
      <xdr:to>
        <xdr:col>5</xdr:col>
        <xdr:colOff>657225</xdr:colOff>
        <xdr:row>20</xdr:row>
        <xdr:rowOff>133350</xdr:rowOff>
      </xdr:to>
      <xdr:graphicFrame macro="">
        <xdr:nvGraphicFramePr>
          <xdr:cNvPr id="5" name="Graf 4"/>
          <xdr:cNvGraphicFramePr/>
        </xdr:nvGraphicFramePr>
      </xdr:graphicFrame>
    </xdr:twoCellAnchor>
  </xdr:twoCellAnchor>
</xdr:wsDr>
```

```

    </xdr:nvGraphicFramePr>
    <xdr:xfrm>
      <a:off x="0" y="0"/>
      <a:ext cx="0" cy="0"/>
    </xdr:xfrm>
    <a:graphic>
      <a:graphicData
uri="http://schemas.openxmlformats.org/drawingml/2006/chart">
        <c:chart
xmlns:c="http://schemas.openxmlformats.org/drawingml/2006/chart"

xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationsh
ips" r:id="rId2"/>
          </a:graphicData>
        </a:graphic>
      </xdr:graphicFrame>
      <xdr:clientData/>
    </xdr:twoCellAnchor>
    <xdr:twoCellAnchor>
      <xdr:from>
        <xdr:col>8</xdr:col>
        <xdr:colOff>333375</xdr:colOff>
        <xdr:row>15</xdr:row>
        <xdr:rowOff>38100</xdr:rowOff>
      </xdr:from>
      <xdr:to>
        <xdr:col>16</xdr:col>
        <xdr:colOff>28575</xdr:colOff>
        <xdr:row>29</xdr:row>
        <xdr:rowOff>114300</xdr:rowOff>
      </xdr:to>
      <xdr:graphicFrame macro="">
        <xdr:nvGraphicFramePr>
          <xdr:cNvPr id="6" name="Graf 5"/>
          <xdr:cNvGraphicFramePr/>
        </xdr:nvGraphicFramePr>
        <xdr:xfrm><a:off x="0" y="0"/>
        <a:ext cx="0" cy="0"/>
      </xdr:xfrm>
      <a:graphic>
        <a:graphicData
uri="http://schemas.openxmlformats.org/drawingml/2006/chart">
          <c:chart
xmlns:c="http://schemas.openxmlformats.org/drawingml/2006/chart"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationsh
ips" r:id="rId3"/>
            </a:graphicData>
          </a:graphic>
        </xdr:graphicFrame>
        <xdr:clientData/>
      </xdr:twoCellAnchor>
    </xdr:wsDr>

```