## PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO KATEDRA INFORMATIKY

# BAKALÁŘSKÁ PRÁCE

Editor tabulek pro LaTeX



Tomáš Jurča

2011

## Anotace

Tato práce se zabývá tvorbou tabulkového editoru pro typografický systém  $\not ET_EX$ . Úvodem popisuje prostředí  $T_EX$  a jeho nádstavby. Dále se zabývá popisem implementace tříd a dialogů použitých k vytvoření tabulkového editoru. Obsahem je také manuál určený pro uživatele vytvořeného editoru, který popisuje základní práci s programem.

<sup>&</sup>quot;Touto cestou bych rád poděkoval vedoucímu mé bakalářské práce panu Mgr. Tomáši Kührovi za jeho odborné vedení a připomínky. Děkuji také panu RNDr. Miloslavu Závodnému za jeho cenné rady, které mi byly při zpracování mé bakalářské práce přínosem. Také prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a uvedl jsem všechny literární zdroje, ze kterých jsem čerpal."

## Obsah

1	Úvo	d 8
т.	11	TrX 8
	1.1.	$111  \text{TeV program} \qquad \qquad$
		$112 \text{ Try system} \qquad 0$
	19	$IAT_{DY} \qquad \qquad$
	1.2.	$\begin{array}{c} \mathbf{L}_{1} \mathbf{L}_{2} \mathbf{L}_{3} \mathbf{L}_$
		$1.2.1.  OBETEA \dots \dots$
	12	$1.2.2.  \text{Forma structura dokumentu}  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $
	1.5.	Sazba bezheno textu $\dots$
		1.5.1. Rezy pisma $\dots$ 11 1.2.2. Valikast pieme 12
	1 /	$1.5.2.  \text{venkost pisma} \qquad 13$
	1.4.	Sazba matematiky
2.	Tab	ulka 15
	2.1.	Typy tabulek v IATEXu
		2.11 Tabbing 15
		2.1.2 Tabular 16
		21.3 Array 17
	2.2	Plovoucí prostředí 17
3.	Imp	lementace 19
	3.1.	Třídy
		3.1.1. Bunka
		3.1.2. Border
		3.1.3. Ohraniceni
		3.1.4. Tabulka
		3.1.5. Dokument
	3.2.	Formuláře dokumentu
		3.2.1. Form1
		3.2.2. Nova tabulka
		3.2.3. OhraniceniOkno
		3.2.4. OhraniceniVice
	3.3.	Vybrané části implementace
		3.3.1. Ukládání projektů
		3.3.2. Načtení projektů
		3.3.3. Generování tabulky
		3.3.4. Načtení tabulek z dokumentu T <sub>F</sub> X
4.	Mai	nuál 23
	4.1.	Hlavní okno
	4.2.	Hlavní menu
		4.2.1. Soubor

	4.2.2. Úpravy	26						
	4.2.3. Formát	28						
	4.2.4. Nápověda	28						
4.3.	Panel nástrojů	29						
4.4.	Zobrazení ohraničení buňky	29						
4.5.	Seznam vytvořených tabulek	29						
4.6.	Tlačítka pro editaci řádků a sloupců tabulky	30						
Závěr		31						
Conclu	Conclusions 3							
Reference								
5. Obs	sah přiloženého CD	34						

## Seznam obrázků

1.	Hlavní okno	23
2.	Rozbalená záložka "Soubor"	24
3.	Upozornění na neuložený dokument	25
4.	Rozbalená záložka "Úpravy"	26
5.	Dialogové okno přidání tabulky a úpravy názvu tabulky	26
6.	Dialogové okno pro ohraničení jedné buňky	27
7.	Dialogové okno pro ohraničení více buňek	27
8.	Rozbalená záložka "Formát"	28
9.	Rozbalená záložka "Nápověda"	29
10.	Panel nástrojů	29
11.	Zobrazení ohraničení aktuální buňky	30

## Seznam tabulek

1.	Třídy dokumentů			•		•	 	•	•	•			10
2.	Příkazy upravující formát písma .					•	 	•	•				12
3.	Příkazy upravující velikost písma					•	 	•	•				13
4.	Akcenty v matematice					•	 	•	•				14
5.	Znaky povolení umístění objektů	•	•	•		•	 	•	•				18

## 1. Úvod

"TEX je možná nejdůležitějším objevem v typografii tohoto století. Zavádí standardní jazyk počítačové typografie a lze ho významově přirovnat ke Gutenbergově vynálezu knihtisku."

Gordon Bell<sup>1</sup>

V dubnu 1977 zahájil Donald Ervin Knuth, profesor Stanfordské univerzity, práci na typografickém systému, jenž nazval T<sub>E</sub>X (čteno "tech"). Podnětem mu byl nevalný výsledek počítačem pořízené sazby druhého vydání jeho knihy, který se vůbec nedal srovnat s vydáním připraveným ručně. Cílem tehdy vzniklého projektu T<sub>E</sub>X bylo vytvoření typografického systému, "jímž připravené dokumenty budou nejen krásné, ale přímo nejkrásnější<sup>"2</sup>, a který bude nezávislý na změnách v tiskařské technologii ("z archivovaného souboru obsahujícího popis knihy získáme stejný výsledek dnes i za sto let<sup>"3</sup>). [4]

## 1.1. T<sub>E</sub>X

TEX je volně dostupný počítačový program určený pro pořizování vysoce kvalitní elektronické sazby. K nejpodstatnějším odlišnostem od komerčních "kolegů" patří otevřenost systému. Uživateli se tak otevírá možnost vytvořit si nadstavbu podle vlastní představy a potřeby. K tomu je však potřeba o vlastnostech systému něco vědět. Sazbu jednoduššího textu lze zvládnout pomocí LATEXových příruček. Při náročnějších požadavcích na sazbu se ale myšlenka, že vše lze zvládnout právě pomocí těchto příruček, rozplyne. [3]

## 1.1.1. T<sub>E</sub>X program

Slovy Donalda Knutha:  $T_EX$  je program "určený pro tvorbu hezkých knih a obzvlášť knih obsahujících spoustu matematiky". Algoritmy řádkového zlomu optimalizují zlom v rámci celého odstavce a stránky jsou lámány jednotlivě. Algoritmus dělení slov umožňuje použít v dokumentu až 256 jazyků. Samozřejmostí je zpracování výjimek dělení slov tam, kde je to zapotřebí. Zvláštní pozornost byla také věnována algoritmu řídícímu sazbu matematických výrazů. Ten byl sestaven na základě rozborů nejzdařilejších sazeb matematických publikací. [4]

Na rozdíl od systémů WYSIWYG (co vidíš, to dostaneš), jako je např. MS Office, patří systém  $T_EX$  mezi mark-up (značkové) systémy. Přímo v textu předkládaném  $T_EX$ u ke zpracování (zdrojový text) jsou zapsány tzv. formátovací značky, které

<sup>&</sup>lt;sup>1</sup>Gordon Bell byl 23 let viceprezidentem odpovědným za výzkum a vývoj ve společnosti Digital Equipment.

 $<sup>^2</sup>Z$ ávodný, Miloslav. Typografický systém  $T_{\!E\!}X$ . Matematika fyzika informatika 1999, č.9.

 $<sup>^{3}</sup>$ Závodný, Miloslav. Typografický systém  $T_{E\!X}$ . Matematika fyzika informatika 1999, č.9.

ovlivňují vzhled výsledné sazby (optické značky), resp. zjednodušují zápis textu (logické značky). V samotném programu T<sub>E</sub>X je zabudováno asi tři sta základních značek, tzv. primitivů. Ty postačují k dokonalému zformátování textu (úprava dokumentu, jeho stránek, odstavců a řádků), k přímé práci se však nehodí. Další, uživatelsky přítulnější, značky (řídící sekvence, "příkazy") jsou z těchto primitivů sestavovány pomocí speciálního programovacího jazyka, jímž je T<sub>E</sub>X vybaven. Logika T<sub>E</sub>Xu vyžaduje nejprve vytvoření souboru vhodných (základních) značek a jeho následné zpracování inicializační verzí programu T<sub>E</sub>X (iniT<sub>E</sub>X) do binární podoby, tzv. formátu - soubor s příponou \*.fmt. Text určený k sazbě (obvykle soubor s příponou \*.tex) je pak předkládán programu T<sub>E</sub>X (virT<sub>E</sub>X), který při startu vyžaduje načtení formátu. Často se pak tato "pracující" verze T<sub>E</sub>Xu pojmenovává podle jména formátu. [4]

### 1.1.2. T<sub>E</sub>X systém

 $T_EX$  je volně a zdarma šiřitelný publikační systém. Samotný program  $T_EX$  je pouze jeho základní součástí. V systému jsou zařazeny různé standardní formáty (vytvořené podle obvyklých potřeb uživatelů  $T_EXu$ ), další knihovny značek (často nazývané "styly", sloužící např. pro podporu národní sazby, tvorbu rejstříků, snadné zařazení postscriptových obrázků, ale třeba i pro sazbu šachových diagramů, notových partitur atd.), fonty, program METAFONT (pro generování znakových fontů i obrázků) a související programy, ovladače výstupních zařízení, rozsáhlá dokumentace aj. [4]

Balík T<sub>E</sub>Xovského softwaru však není nutné instalovat celý. Ne vše je pro dobrou funkci systému potřebné. Instalace na pevném disku může zabírat 7MB, ale i 170 MB. [4]

## 1.2. I₄T<sub>E</sub>X

IAT<sub>E</sub>X je rozšířením T<sub>E</sub>Xu o nové balíky maker. Nejnovější verzí je IAT<sub>E</sub>X  $2_{\varepsilon}$  Je to formát přívětivější pro začátečníky, protože pro sázení je využito různých prostředí, která začínají \begin{...} a končí \end{...}. Uživatel se tak stará pouze o to, co chce vysázet a IAT<sub>E</sub>X sám řeší, jak to vysází, tak aby výsledný soubor vypadal dobře. Tento formát také podporuje přidávání obrázků v zdrojovém souboru. Jeho drobnými nevýhodami mohou být velká podobnost souborů vzniklých v IAT<sub>E</sub>Xu a také nemožnost užití nových příkazů ve starších verzích. Podobnosti souborů lze však předejít použitím dodatečných stylů. [1]

## 1.2.1. CSI≱T<sub>E</sub>X

CSET<sub>E</sub>X je počeštěný formát ET<sub>E</sub>Xu, který tak umožňuje uživatelům sázet české znaky, které by jinak musely být sázeny pomocí speciálních příkazů.

## 1.2.2. Povinná struktura dokumentu

Jelikož je I<sup>A</sup>T<sub>E</sub>X strukturovaný, musíme mu na začátku specifikovat typ, začátek a konec dokumentu. Povinná minimální struktura dokumentu proto vypadá [1][2]:

```
\documentclass[11pt]{article}
\usepackage{czech,a4}
\begin{document}
\end{document}
```

Předchozí text je pouze ukázka prázdného dokumentu, na němž lze objasnit základní příkazy [1][2]:

• \documentclass[options]{class},

kde parametr *class* určuje formátování textu např. velikost stránek, číslování, záhlaví atd. Možné třídy dokumentů shrnuje tabulka č.1. Parametr *options* je volitelný a určuje doplňkové vlastnosti stylu. Pokud není uveden, tak se implicitně vkládá velikost písma 10pt a velikost stránky letterpaper. Zvolená velikost písma pak určuje velikost příkazu \normalsize, od kterého IATEX dopočítává ostatní velikosti.

Třída	Nejčastější příklad použití
article	Pro odborné články, prezentace, krátké zprávy, doku-
	mentaci k programům, pozvánky apod.
report	Pro delší zprávy obsahující několik kapitol, menší knihy,
	diplomové práce apod.
book	Pro skutečné knihy
slide	Pro blány (slides). Tato třída užívá velkého bezpatko-
	vého písma, jež je pro zpětnou projekci nejvhodnější.

Tabulka 1. Třídy dokumentů

### • \usepackage[options]{package}

načte určité "styly", které přidávají např. nové fonty, podporu grafiky, nové definice nebo mění základní nastavení LATEX. Tento příkaz lze použít např. pro doplnění češtiny do LATEXu.

• \begin{document}

určuje začátek dokumentu. Teprve od tohoto místa, začíná  ${\rm L\!AT}_{\rm E\!X}$ sázet dokument.

• \end{document}

určuje konec dokumentu. Tímto místem končí sazba dokumentu. To, co je za tímto příkazem, není vysázeno a ETEX na tento text nebere ohled.

## 1.3. Sazba běžného textu

Pokud chceme vytvořit text bez formátování, stačí pouze psát a není třeba používat příkazů  $PT_EXu$ . Kdybychom však chtěli např. změnit typ písma, řezy písma či velikost písma, je třeba příkazů použít. [1]

Při psaní textu je také nutno dbát na některá omezení [1]:

- na nový řádek přejdeme pomocí příkazu \newline
- nový odstavec lze vytvořit buď vynecháním alespoň jednoho prázdného řádku (při vynechání více řádků, se vytvoří pouze nový odstavec a nezvětší se mezera mezi odstavci), nebo příkazem \par
- je respektována pouze jedna mezera (při použití více mezer nebo tabulátoru se nezvětší mezera mezi textem)
- některé speciální znaky (#, \$, %, &, ; ,, \, {, }) se v textu nevysází pokud stojí samostatně. Před takovým znakem (který chceme samostatně vysázet) je třeba vložit \. Výjimkou je \, které by při použití \\ znamenalo jiný příkaz způsobí přechod sázení na nový řádek a volitelný argument stanový velikost mezery mezi odstavci. \ lze vložit pouze pomocí příkazu \$\backslash\$.
- pro elipsu (...) je třeba použít příkaz \ldots nebo \dots
- pro české uvozovky (dole začínají a nahoře končí) se musí použít příkaz \uv{text}, který je definován v balíku czech.sty
- anglické uvozovky se začínají pomocí (') a končí(')

## 1.3.1. Řezy písma

Systém TEX používá svá vlastní písma nazvaná Computer Modern (CM). Jejich předlohou se staly rytiny z období klasicismu. K běžné sazbě se používá antikva, což je stojaté netučné písmo. Pro zvýraznění některých pasáží lze použít *italiku (kurzívu)*. Toto písmo má šikmou osu a proto je třeba po skončení příkazu použít \/, abychom zabránili slití (prolnutí) písmen. Pro zdůraznění krátkých textů se používá **tučný řez** a pro zvýraznění názvů a jmen lze použít KAPITÁ-LEK (malá písmena jsou stejného tvaru jako velká, jen jsou zmenšena v měřítku). [1]

Ve starší verzi LATEXu se vybírali základní řezy písma pomocí příkazů \rm, \it, \bf, \sf, \sl, \sc, \tt (což postupně znamená antikva, *italika*, **tučná antikva**, grotesk, *skloněná antikva*, KAPITÁLKY a strojopis). Tyto příkazy jsou kompatibilní s plainTEXem a uvádí se před textem. Není je možné mezi sebou (na rozdíl od novější verze) kombinovat. Lze mezi nimi pouze přepínat, tzn. pokud chceme tučné použijeme příkaz \bf a jakmile skončíme vrátíme se příkazem \rm k normálnímu písmu. Častěji byly používány bloky, které usnadnily přehled textu a pro výše zmíněný tučný text vložený mezi normálním vypadaly takto:

\rm normální text {\bf tučný text} pokračuje normální text, což vysází

"normální text tučný text pokračuje normální text". [1]

Jelikož uvedené příkazy přepínají pouze na konkrétní řez písma, nelze s nimi dosáhnout kombinací a v některých případech nastanou problémy, pokud použijeme jiné písmo než Computer Modern. Z těchto důvodů bylo zavedeno nové schéma označované jako NFSS, v němž se místo konkrétních příkazů používají přepínače atributů, které pak LATEX sám vyhodnotí a podle nich určí písmo.

Pro všechny kombinace nemusí existovat písmo, proto LATEX sám zvolí podobné písmo a vypíše varování, že takové písmo, které jsme chtěli neexistuje. V tabulce č. 2. jsou tyto příkazy uvedeny. Např. tučný skloněný text se zapíše jako:

\textbf{\textit{tučný skloněný text}}, což vysází
,,tučný skloněný text". [1]

Příkazy nastavující rodinu písma

\textrm	Nastaví antikvu
\textsf	Nastaví grotesk
\texttt	Nastaví strojové písmo

Příkazy nastavující váhu písma

\textbf	Nastaví tučné písmo
\textmd	Nastaví normální písmo (antikvu)

Příkazy nastavující tvar písma

\textsc	Nastaví kapitálky
\textsl	Nastaví skloněné písmo
\textit	Nastaví kurzívu (italiku)
\textup	Nastaví vzpřímené písmo

Další příkazy

\textnormal	Nastaví normální písmo dokumentu (tj. zruší všechny
	ostatní atributy)
\emph	Nastaví nebo zruší naklonění písma podle toho jaký je
	text v jeho okolí
$\text$	Podtrhne uvedený text

Tabulka 2. Příkazy upravující formát písma

#### 1.3.2. Velikost písma

Základní velikost písma (implicitně daná) je 10pt. Tuto základní velikost lze změnit na velikosti 11pt a 12pt příkazem \documentclass (viz kapitola 1.2.2.). Velikost písma \normalsize pak určuje stanovená (či implicitně daná) základní velikost písma. Od této velikosti se pak odvíjejí ostatní velikosti dle příkazů uvedených v tabulce č. 3. [1]

Příkaz	Ukázka
\tiny	písmo
\scriptsize	písmo
\footnotesize	písmo
\small	písmo
\normalsize	písmo
\large	písmo
\Large	písmo
\LARGE	písmo
\huge	písmo
\Huge	písmo

Tabulka 3. Příkazy upravující velikost písma

## 1.4. Sazba matematiky

Nespornou výhodou TEXu je vynikající sazba všech typů matematických výrazů. Pro větší počet matematických symbolů, příkazů atd. používáme např.  $A_{MS}$ -LATEX. [1]

Existují dva typy matematického prostředí. Jedno slouží pro vkládání matematiky uvnitř textu a nehodí se pro vkládání složitých matematických výrazů. Zápis vypadá takto: **\$rovnice\$**. Druhé slouží k sázení textu na nový řádek a je určeno právě pro složité výrazy jako rovnice apod. Zápis vypadá takto: **\$\$rovnice\$\$**. [1]

V matematice (při použití matematického fontu) nefungují akcenty tak, jako v běžném textu, proto musíme použít příkazy \akcent{znak} viz tabulka č. 4. Samozřejmě se lze přepnout do textového módu \mbox. Jako příklad můžeme uvést: \$a+b=c,\quad\mbox{pro všechny}...\$.

Před vložením tučného skloněného matematického symbolu musíme použít příkaz \boldmath a po skončení \unboldmath. [1]

Aby nám LATEX vysázel v matematickém textu desetinou čárku, je třeba použít {,}, jinak by LATEX vysázel malou mezeru za desetinou čárku. [1]

Pro matematický text také nefungují stejné příkazy pro řezy písma jako v běžném textu. Zde se používají příkazy [1]:

- \mathbf ... pro tučný řez písma
- \mathit ... pro textovou italiku
- \mathnormal ... pro matematickou italiku

Příkaz	Ukázka	Popis
\acute	á	Čárka nad písmenem
$\check$	ă	Háček nad písmenem
\breve	$\breve{a}$	Půlkruh nad písmenem
\dot	$\dot{a}$	Tečka nad písmenem
\ddot	$\ddot{a}$	Dvojtečka nad písmenem
\grave	à	Obrácená čárka nad písmenem
\hat	$\hat{a}$	Stříška nad písmenem
\tilde	$\tilde{a}$	Vlnovka nad písmenem
\vec	$\vec{a}$	Šipka doprava nad písmenem

Tabulka 4. Akcenty v matematice

## 2. Tabulka

Tabulkou je jakýkoli organizovaný a graficky členěný soupis údajů uspořádaný do sloupců a řádků. Každý řádek či sloupec je složen z několika buněk. Každá buňka má svou jedinečnou pozici danou číslem řádku a sloupce. Uchovává v sobě informace o obsahu a jeho formátování. K tvorbě tabulky lze v IATEXu použít prostředí tabbing, tabular a array, která jsou popsána níže.

## 2.1. Typy tabulek v LATEXu

## 2.1.1. Tabbing

Tento typ tabulky je určen k vytváření zarovnaných textů bez produkce čar (obdobně jako u psacího stroje) pomocí tabelačních zarážek. Umožňuje tedy vkládat text do sloupců. Při psaní textu se vkládají příkazy [1]:

- \= ... pro nastavení zarážky
- $\bullet$   $\backslash > \ldots$  přesune text na pozici další zarážky
- \+ a \- ... posouvají místo první zarážky na novém řádku
- \kill ... je možno uvést na nastavovacím řádku, v němž označíme pozice zarážek, ale nechceme vysázet
- $\bullet$  <br/>  $\cdot$  . . . posune všechno zapsané co nejdále doprava v nynějším sloupci
- $\bullet$  \' ... umožňuje umístit text zarovnaný vpravo oproti poslední zarážce

Například k vytvoření této tabulky:

1.sloupec2.sloupec3.sloupec1.sloupec3.sloupec

musíme napsat

```
\begin{tabbing}
1.sloupec \= 2.sloupec \= 3.sloupec \\
1.sloupec \>\>3.sloupec\\
\end{tabbing}
```

#### 2.1.2. Tabular

Toto prostředí se používá k sazbě tabulek, které chceme s ohraničením. Šířku a výšku buněk určuje LATEX. Tabulku začneme příkazem \begin{tabular}{specifikace} a ukončíme \end{tabular}. Jednotlivé sloupce tabulky oddělujeme znakem & a na nový řádek se dostaneme pomocí \\. Vodorovné čáry vkládáme na samostatném řádku před nebo za každým řádkem, přičemž vložení před znamená horní ohraničení řádku a vložení za znamená spodní ohraničení řádku. [1][2]

Oddělení sloupců se vkládá do argumentu specifikace [1]:

- | ... způsobí vytištění vysázení svislé čáry mezi sloupce
- || ... způsobí vytištění vysázení dvojité čáry mezi sloupce
- @{symbol} ... způsobí vysázení symbolu symbol mezi sloupce
- bez znaku ... sloupec nebude oddělen od předchozího žádným znakem, pouze mezerou

Do povinného argumentu specifikace je třeba také zadat zarovnání jednotlivých sloupců a to následujícím způsobem [1]:

- c ... zarovnání textu doprostřed buňky
- r ... zarovnání textu na pravý okraj buňky
- 1 ... zarovnání textu na levý okraj buňky
- p{vel} ... vytvoří sloupec o dané šířce, který se případně zalomí
- \*{n}{def} ... použití pro zkrácení definice. Definice def (sestavená z c, r, l, p{vel}) se nakopíruje n krát.

Pro sloučení několika buněk na řádku je určen příkaz \multicolum{počet sloupců}{zarovnání}{text}. Pro vysázení čar slouží tyto příkazy [1]:

- \hline ... vysází vodorovnou čáru přes celou šíři tabulky
- \cline{n-m} ... vysází vodorovnou čáru jen mezi n-tým a m-tým sloupcem
- \vline ... vysází svislou čáru v místě uvedení přes jeden řádek

Abychom zvětšili výšku řádku v tabulce musíme před prostředí tabular uvést \def\arraystretch{1.5}, kde 1.5 znamená násobek původní výšky řádku. Pokud již nechceme mít výšku řádku v dalších tabulkách větší, vrátíme nastavení výšky řádku zpět pomocí \def\arraystretch{1.0}, což uvedeme za tabulkou. Například k vytvoření tabulky č. 4. musíme napsat:

```
\begin{tabular}{lcl}
Příkaz &Ukázka & Popis \\
\hline\hline
\verb|\acute| & $\acute{a}$ & Čárka nad písmenem\\
\verb|\check| & $\check{a}$ & Háček nad písmenem\\
\verb|\breve| & $\breve{a}$ & Půlkruh nad písmenem\\
\verb|\dot| & $\dot{a}$ & Tečka nad písmenem\\
\verb|\dot| & $\dot{a}$ & Dvojtečka nad písmenem\\
\verb|\ddot| & $\dot{a}$ & Obrácená čárka nad písmenem\\
\verb|\grave| & $\grave{a}$ & Obrácená čárka nad písmenem\\
\verb|\hat| & $\hat{a}$ & Stříška nad písmenem\\
\verb|\tilde| & $\tilde{a}$ & Vlnovka nad písmenem\\
\verb|\vec| & $\vec{a}$ & Šipka doprava nad písmenem\\
\hline
\end{tabular}
```

#### 2.1.3. Array

Array je prostředí určené pro tvorbu matematických tabulek jako jsou např. matice. Oddělení sloupců funguje stejně jako v předchozím prostředí tabular. Odlišné je pouze zadání začátku a konce tabulky, kdy začínáme příkazem \begin{array}{specifikace} a končíme \end{array}.

V tomto prostředí je třeba dbát na správnou sazbu matematiky, jelikož se jednotlivé buňky sází v matematickém režimu. Sazba matematiky je popsána v kapitole 1.4. na straně 13.

## 2.2. Plovoucí prostředí

Při vkládání více obrázků a tabulek vyvstává komplikace s umisťováním těchto objektů na stránku. Tyto objekty není možné většinou rozdělovat, a proto je třeba nastolený problém řešit. [2]

Jedním z řešení by bylo vysázet tyto objekty vždy na nové stránce, což by ovšem vedlo k vynechání místa na stránce a mohlo by dojít ke vzniku poloprázdných stran, které ovšem nejsou žádoucí. [2]

Dalším řešením je použití plovoucího prostředí, které zajistí vysázení objektu hned, jakmile je to možné. Tzn., že na rozepsanou stránku je vysázen text následující po objektu a jakmile se uvolní potřebný prostor (většinou při přechodu na novou stránku) je vysázen vkládaný objekt. U tohoto sázení však může dojít k odsunutí sázení na poslední stranu dokumentu a to proto, že je tabulka či obrázek příliš velký a nevleze se na stránku. LATEX tvoří z objektů frontu a důsledně dodržuje pořadí. Pokud se v pořadí první tabulka ve frontě nevysází, nevysází se ani ty za ní. LATEX nabízí dvě prostředí - jedno pro obrázky (figure) a druhé pro tabulky (table). [2] Objekt umístěný v tomto prostředí je považován za plovoucí objekt. Obě prostředí mají ještě nepovinný parametr specifikace prostředí. Tento parametr řekne  $I_{\rm TE}$ Xu, kam na stránce lze daný plovoucí objekt umístit, tedy kam se má pokusit daný objekt umístit. Znaky pro povolení umístění jsou uvedeny v tabulce č. 5. [2]

znak	povolené místo, kam umístit objekt
h	Na tom samém místě, kde se tabulka objevuje ve zdro-
	jovém textu. Vhodné pro menší objekty.
t	Nahoře na stránce.
b	Dole na stránce.
р	Na zvláštní straně, kde se objeví jen plovoucí objekty.
!	Není brán ohled na většinu vnitřní parametrů prostředí
	(maximální počet plovoucích objektů na stránce a po-
	dobně), které by jinak nedovolili vysázet danou tabulku
	na stránce.

Tabulka 5. Znaky povolení umístění objektů

## 3. Implementace

## 3.1. Třídy

### 3.1.1. Bunka

Tato třída slouží k uchování formátování buňky. Je v ní obsažen obsah buňky, její zarovnání a ohraničení a také velikost a řez písma.

#### 3.1.2. Border

"Border" je třída určená k uchování ohraničení jedné konkrétní buňky.

## 3.1.3. Ohraniceni

Třída "ohraniceni" v sobě uchovává informace o ohraničení všech buňek. Jejím hlavním využitím je uchování a editace ohraničení aktuální tabulky v dokumentu tak, aby se uložilo až při přechodu na jinou tabulku. Obsahuje také metody sloužící k přidání a odebírání řádku a sloupce.

#### 3.1.4. Tabulka

Tato třída v sobě nese informace o jednotlivých buňkách, počtu řádků a sloupců, názvu a typu tabulky a uchovává také textový obsah dokumentu před a za tabulkou. Toto uchování textu slouží k následnému generování textové části dokumentu tak, jak vypadala před zásahem do tabulek.

## 3.1.5. Dokument

Třída "dokument" obsahuje seznam tabulek, číslo a ohraničení aktuální tabulky. Součástí je také proměnná, která určuje, zda byl, či nebyl dokument změněn.

## 3.2. Formuláře dokumentu

#### 3.2.1. Form1

Hlavní formulářové okno je Form1. V něm se řeší téměř celá funkčnost programu. K hlavním metodám patří přidání/odebrání tabulky, načtení tabulek z T<sub>E</sub>Xovského dokumentu, vygenerování T<sub>E</sub>Xovského dokumentu z aktuálního dokumentu a uložení a načtení tabulky.

#### 3.2.2. Nova tabulka

Toto dialogové okno slouží k přidání nové tabulky do dokumentu, kdy uživatel má možnost zvolit si název a typ tabulky.

#### 3.2.3. OhraniceniOkno

Dialogové okno "ohraniceniOkno" je určeno k nastavení ohraničení při výběru jedné buňky. Nejdůležitější metodou je kresliHrany, která má za úkol vykreslit vybrané ohraničení tak, aby ho uživatel viděl.

#### 3.2.4. OhraniceniVice

Toto dialogové okno slouží k nastavení ohraničení více vybraných buňek, kde lze zvolit vnější či vnitřní ohraničení. Nejdůležitější metodou je stejně jako u OhraniceniOkna kresliHrany, která má stejnou funkci.

## 3.3. Vybrané části implementace

#### 3.3.1. Ukládání projektů

K ukládání projektu jsem si zvolil serializaci dat, která umožňuje ukládat celou instanci objektu Dokument. K tomu, aby bylo možno instanci nějakého objektu uložit (v mém případě třída Dokument), musíme před deklaraci jeho třídy i použitých tříd v ní umístit atribut Serializable.

Dále musíme do třídy či okna, která obsahuje metodu pro ukládání objektu přidat bázovou třídu System.IO, pro proudy dat a bázovou třídu System.Runtime.Serialization.Formatters.Binary, obsahující metodu k serializaci. Hlavní část kódu serializace,která již ukládá instanci objektu, je obsažena v metodě uloz v okně Form1. Tato metoda obsahuje:

```
1 BinaryFormatter formatovac = new BinaryFormatter();
2 FileStream parametry = new FileStream(adresa, FileMode.Create,
FileAccess.Write, FileShare.None);
3 formatovac.Serialize(parametry, document);
4 parametry.Close();
```

Na prvním řádku vytváříme novou instanci třídy BinaryFormatter, která obsahuje metodu serializace. Na dalším řádku vytvoříme ukládaný soubor s parametry:

- adresa . . . název souboru (případně i s cestou k souboru jako v mém případě)
- FileMode.Create ... vytvoří soubor. Pokud již daný soubor existuje, bude přepsán.
- FileAccess.Write ... přístup k souboru pro zápis
- FileShare.None . . . jakýkoli přístup k souboru z venku je při ukládání zakázán

Pomocí dalšího řádku do vytvořeného souboru **parametry** zapíšeme data ukládané třídy **document**. Na posledním řádku už je jen ukončení práce se souborem.

### 3.3.2. Načtení projektů

Pro načtení uloženého dokumentu je zvolena deserializace, která načte data do jednotlivých tříd, tak jak byly uloženy. Výkonná část kódu je obsažena v metodě nacti v okně Form1 a obsahuje:

```
1 BinaryFormatter formatovac = new BinaryFormatter();
2 FileStream parametry = File.OpenRead(adresa);
3 Dokument docP = (Dokument)formatovac.Deserialize(parametry);
4 parametry.Close();
5 return docP;
```

Nejdůležitější částí je 3.řádek, kde se z souboru **parametry** vytváří dokument **docP**, který je předán k načtení do vizuální podoby volající metodou (viz poslední řádek).

#### 3.3.3. Generování tabulky

Ke generování tabulky je určena metoda vytvorTabulkuTeX v okně Form1. Nejdříve se v ní vygeneruje záhlaví. Zarovnání jednotlivých sloupců se určuje nejvyšším počtem výskytů ve sloupci. Pro ohraničení se volí také nejvyšší počet výskytů. Poté se začnou generovat jednotlivé řádky.

Při tabulce tabular se převádí tučně na příkaz \textbf, kurzíva na příkaz \textit, podtrženě na \underline a velikosti se převádí na příkazy viz tabulka č. 3. Při tabulce array se převádí tučně na příkaz \mathbf, kurzíva na příkaz \mathit a podtrženě na \underline. Tyto příkazy se do sebe postupně zanořují. Jiná volba levého či pravého ohraničení je převedena na příkaz \multicolumn{1}{vlevo zarovnani vpravo}{text}. Odlišná volba zarovnání je řešena podobně jako ohraničení. Ohraničení řádku dole (nahoře u prvního řádku) je převedeno na sekvenci \cline{od-do}, nebo \hline v závislosti na tom, zda je všude ohraničení.

## 3.3.4. Načtení tabulek z dokumentu TEX

Tato metoda je v okně Form1. Načtení probíhá v cyklu, kdy čekáme až do znaku \, který v LATEXu znamená začátek příkazu. Jestliže na znak narazíme, přečteme název příkazu. Pokud je příkazem začátek prostředí, zjistíme, o které prostředí se jedná, a podle toho se zachováme. Pokud se objevil příkaz verb nebo komentář, přeskočíme text buď do konce verb, nebo do konce řádku. Jestliže příkaz byl tabular (metoda ctiTabular) či array (metoda ctiArray) začneme číst tabulku až do jejího ukončení. Prostředí verbatim přeskakujeme stejně jako verb jen je k tomu určena metoda ctiVerbatim.

Při čtení tabulky nejdříve přečteme záhlaví a potom čteme jednotlivé buňky, které jsou mezi sebou odděleny znakem & nebo  $\$  na konci řádku tabulky. Při čtení není podporováno načítání prostředí či příkazy, které používají znak &, kromě příkazu  $\$  a speciálních znaků.

## 4. Manuál

Tato kapitola se zabývá popisem uživatelského rozhraní tabulkového editoru pro LATEX. LATEX editor je určen k vizuální tvorbě a editaci tabulek vytvořených přímo v tomto programu. Lze také načíst již existující tabulky ze souborů uživatele a následně je editovat.

## 4.1. Hlavní okno

Po spuštění programu se zobrazí prázdné okno (viz obrázek č. 1.), které se skládá z hlavního menu a panelu nástrojů. Na pravé straně okna se dále zobrazuje zvolené ohraničení aktuální buňky, seznam všech vytvořených tabulek a jsou zde také umístěna tlačítka určená k editaci sloupců a řádků tabulky.



Obrázek 1. Hlavní okno

## 4.2. Hlavní menu

Slouží k ovládání celé aplikace, je rozděleno na čtyři základní části popsané níže. Jednotlivé akce v "Menu" jsou aktivní pokud obrázky před akcí jsou modré. Neaktivní jsou, když jsou obrázky šedé.

Hlavní menu obsahuje:

- soubor ... slouží k vytvoření nového dokumentu, otevření dokumentu, uložení rozpracovaného dokumentu a ukončení programu a je určeno k převedení jedné či všech tabulek do formátu  $T_{\rm E} X$
- úpravy ... je určeno k práci s tabulkou
- formát ... je určeno k práci s řezy písma a zarovnáním textu v buňce
- nápověda ... stručně popisuje práci s editorem

## 4.2.1. Soubor

🔲 Ta	bulkový editor pro L	aTeX		
Sou	ibor Úpravy Fo	rmát Nápověda		
	Nový	Ctrl+N		
	Otevřít	Ctrl+O	Tabulky	
	Uložit jako	Ctrl+S	Ohraničení buňky A1:	
	Načti tabulky z dok	umentu TeX	· · · · · · · · · · · · · · · · · · ·	
-	Převeď tabulku do Vygeneruj TeX dok	TeXu ument		
0	Ukončit	Ctrl+E		
	\verb \large	\large{písmo}	Seznam všech tabulek:	

Obrázek 2. Rozbalená záložka "Soubor"

Obrázek č. 2. ukazuje rozbalenou záložku "Soubor", kterou dělíme na:

- Volba "Nový" je určena k vytvoření nového dokumentu. K ulehčení činnosti je zde klávesová zkratka (CTRL + N), která vyvolá událost vytvoření nového souboru. Po spuštění programu je nový dokument vytvořen automaticky tzn., že je možno rovnou tvořit a následně upravovat tabulky. V případě, že máme rozpracovaný neuložený dokument a tuto nabídku použijeme, program se zeptá, zda chceme změny uložit (viz obrázek č. 3.) a teprve poté vytvoří nový dokument.
- Nabídka "Uložit …" slouží k uložení rozpracovaného dokumentu. K ulehčení činnosti je zde klávesová zkratka (CTRL + S), která vyvolá otevření okna, kde je třeba nejprve zadat cestu, kam soubor uložit a název souboru. Následně kliknutím na tlačítko "Uložit" provedeme uložení. Při stisknutí tlačítka "Storno" se zavře okno a neprovede se nic. Tuto akci lze provést jen, když je aktivní, tedy když byla v dokumentu provedena nějaká změna.

Volba "Otevřít …" slouží k otevření již existujícího uloženého dokumentu s příponou \*.tbl, který byl vytvořen touto aplikací. K ulehčení činnosti je zde klávesová zkratka (CTRL + O), která vyvolá otevření okna, kde je třeba zadat cestu k souboru a kliknout na tlačítko "Otevřít". Při stisknutí tlačítka "Storno" se zavře okno a neprovede se nic. Tato volba je dostupná kdykoli a v případě, že ji spustíme při práci s neuloženým dokumentem, program se zeptá, zda chceme uložit změny stejně jako při volbě "Nový" (viz obrázek č. 3.).

Uložit?
Chcete uložit rozpracovaný dokument?
Ano Ne Storno

Obrázek 3. Upozornění na neuložený dokument

- Volba "Načti tabulky z dokumentu TeX" je určena k načtení všech tabulek T<sub>E</sub>Xovského dokumentu. Tato funkce podporuje načtení prostředí tabular a array. Není podporováno načtení vnořených tabulek v tabulce a prostředí, která obsahují &. Tabulky dokumentu se načtou a očíslují tak, jak po sobě následují v dokumentu. Načtené tabulky není možné z dokumentu odstranit. Editor si pamatuje umístění tabulek v původním dokumentu a je schopný vygenerovat nový dokument v původním rozvržení se změněnými tabulkami.
- "Převed' tabulku do TeX" slouží k převedení aktuální tabulky dokumentu do T<sub>E</sub>Xu. Její součástí je volba názvu nově vytvořeného souboru s příponou \*.tex na disku.
- Volba "Vygeneruj TeX dokument" je určena ke generování T<sub>E</sub>Xovského dokumentu ze všech tabulek. V případě, že uživetel tvoří tabulky přímo v dokumentu je vygenerován dokument obsahující pouze tyto tabulky. Pokud byl načten T<sub>E</sub>Xovský dokument, který není složený pouze s prostředí tabular a array, je po užití této volby opět mezi tabulky vloženo vše, co obsahoval dokument mezi tabulkami stejně jako v originálním dokumentu a nově vytvořené tabulky jsou přidány na konec dokumentu.
- "Ukončit" slouží k zavření celé aplikace, pokud je aktuální dokument neuložený, program nabídne výzvu k uložení (viz obrázek č. 3.). Při stisknutí tlačítka "Storno" se neprovede nic. K ulehčení činnosti je zde klávesová zkratka (CTRL + E), která zavře aplikaci.

## 4.2.2. Úpravy

🔲 Ta	] Tabulkový editor pro LaTeX							
Sou	bor	Úpravy Formát Nápověda						
B	I	Přídat tabulku 🕴 📰 🏢 🙀 🗊						
	Α	Nastavení tabulky	Tabulky					
•	Pří	Ohraničení buňky / buněk	Ohraničení buňky A1:					
	\ve	Přidat řádek						
	\ve	Přídat sloupec						
	\ve	Odebrat řádek / řádky						
	\ve	Odebrat sloupec / sloupce						
	\ve	Odsheetteluller						
	\ve	Odebrat tabuiku	Seznam všech tabulek:					

Obrázek 4. Rozbalená záložka "Úpravy"

Obrázek č. 4. zobrazuje rozbalenou záložku "Úpravy", kterou dělíme na:

• "**Přidat tabulku**" ... při použití se zobrazí dialogové okno (viz varianta a obrázku č. 5.), kam lze vložit název tabulky a typ tabulky (výběr z obyčejná a v matematice). Následně se vloží nová tabulka na konec dokumentu.

Nová tabulka		Vlastnosti tabulk	y	
Název tabulky:		Název tabulky:	Tabulka č.1	
Typ tabulky:	<ul> <li>obyčejná (tabular)</li> <li>v matematice (array)</li> </ul>	Typ tabulky:	<ul> <li>obyčejná (tabular)</li> <li>v matematice (array)</li> </ul>	
	OK storno		OK storno	
	a)		b)	

Obrázek 5. Dialogové okno přidání tabulky a úpravy názvu tabulky

- "Nastavení tabulky" ... zobrazí dialogové okno (viz varianta b obrázku č. 5.), které umožňuje změnit název tabulky v dokumentu během práce s tabulkou.
- "Ohraničení buňky/buněk" … lze použít dvěma způsoby pro ohraničení jedné buňky nebo pro ohraničení více buněk. Při volbě ohraničení jedné buňky se zobrazí dialogové okno (viz obrázek č. 6.), v němž lze volit ohraničení nahoře, dole, vlevo a vpravo ze tří možností - žádné, jednoduché a dvojité. Při volbě ohraničení více buněk se zobrazí rozšířené dialogové okno

a můžeme kromě vnějšího ohraničení vybírat také vnitřní ohraničení - vodorovně a svisle (viz obrázek č. 7.). Během volby v pravé části dialogových oken vidíme, jak se ohraničení mění.

Ohraničeni	í buňek	
		Výsledný vzhled ohraničení buňky:
Nahoře	žádné	▼
Dole	jednoduché	•
Vlevo	žádné	•
Vpravo	žádné	▼
		OK Storno

Obrázek 6. Dialogové okno pro ohraničení jedné buňky

Ohraničení buňek			
- Vnější ohraniče	ní	_	Výsledné ohraničení buňek:
Nahoře	dvojité	•	
Dole	dvojité	•	
Vlevo	dvojité	•	
Vpravo	dvojité	•	
- Vnitřní ohraniče	ní		
Vodorovně	jednoduché	•	
Svisle	žádné	•	
			OK Storno

Obrázek 7. Dialogové okno pro ohraničení více buňek

- "Přidat řádek" ... přidá jeden řádek na konec aktuální tabulky
- "Přidat sloupec" ... přidá jeden sloupec na konec aktuální tabulky
- "Odebrat řádek/rádky" ...<br/>odebere jeden či více řádků z aktuální tabulky
- "Odebrat sloupec/sloupce" ... odebere jeden či více sloupců z aktuální tabulky

• "Odebrat tabulku" ... odstraní aktuální tabulku. Před odstraněním se aplikace nezeptá zda chceme provedené změny uložit a automaticky tabulku z dokumentu odstraní.

### 4.2.3. Formát

Obrázek č. 8. ukazuje rozbalenou záložku "Formát", kterou dělíme na:

- "Tučně" ... změní řez písma na tučné
- "Kurzíva" ... změní řez písma na kurzíva
- "Podtrženě" ... změní řez písma na podtržené
- "Zarovnat vlevo" ... zarovná text v buňce/buňkách vlevo
- "Zarovnat na střed" ... zarovná text v buňce/buňkách na střed
- "Zarovnat vpravo" ... zarovná text v buňce/buňkách vpravo

🔲 Tal	bulkový editor pro L	.aTeX			
Sou	bor Úpravy Fo	rmát Nápověda			
в	I U norm	Tučně Ctrl+B			
	A	Kurzíva Ctrl+I	1	Tabulky	
•	Příkaz	Podtrženě Ctrl+U		Ohraničení buňky A1:	
	\verb \tiny	Zarovnat vlevo			
	\verb \script	Zarovnat na střed			
	\verb \footn(	Zarovnat vpravo			
	\verb \small	\small{písmo}	-		
	\verb \norma	\normalsize			
	\verb \large	\large{písmo}		Seznam všech tabulek:	

Obrázek 8. Rozbalená záložka "Formát"

Pokud aktuální buňka obsahuje i příkazy, program upozorní na možné riziko při kompilaci vygenerovaného dokumentu T<sub>E</sub>X.

#### 4.2.4. Nápověda

Nápověda se skládá ze dvou částí (viz obrázek č. 9.):

- "Nápověda"
- "O programu"

První je stručný popis práce s editorem a je možno ji vyvolat klávesovou zkratkou F1 a druhá jsou základní údaje o verzi, názvu a autorovi programu, copyright a důvod vzniku aplikace.

🔲 Tak	oulkový editor pro L	aTeX			
Soul	bor Úpravy Fo	rmát Nápověd	a		
в	I U normalsiz	e 🕜 Náp	ověda F1	5 🔜 🙀 🗊	
	A	В 🍫 Ор	ogramu		Tabulky
•	Příkaz	Ukázka			Ohraničení buňky A1:
	\verb \tiny	\tiny{písmo}			
	\verb \script				
	\verb \footno	\footnotesiz			
	\verb \small	\small{písmo}			
	\verb \norma	\normalsize			
	\verb \large	\large{písmo}			Seznam všech tabulek:

Obrázek 9. Rozbalená záložka "Nápověda"

## 4.3. Panel nástrojů

Ikony použité na tomto panelu jsou velmi podobné jako ikony u MS Office. Jak je vidět na obrázku č. 10., z leva tento panel obsahuje tlačítko pro tučné písmo, kurzívu a podtržené písmo, dále rozbalovací seznam pro volbu velikosti písma. Za ní následují tlačítka zarovnání textu v buňce - vlevo, doprostřed a vpravo. Dále jsou čtyři ikonky - pro přidání nové tabulky, odebrání tabulky, změnu nastavení tabulky a změnu ohraničení buňky.

Tabulkový editor	pro LaTeX		
Soubor Úpravy	Formát Nápověda		
B I U norr	malsize 🔻 📰 📰 📰		
A tiny		Tabulky	
▶ Příkaz foot	notesize	Ohraničení buňky A1:	
\verb \t	noteste II o}		L
\verb \s <sub>larg</sub>	e }{		
\verb \f Larg	ie siz		
\verb \s hug	e mo}		
\verb \r Hug	je ze		

Obrázek 10. Panel nástrojů

## 4.4. Zobrazení ohraničení buňky

V pravé horní části hlavního okna (viz obrázek č. 11.) se zobrazuje ohraničení aktuální buňky. Při výběru více buněk se zobrazuje pouze ohraničení první zvolené buňky.

## 4.5. Seznam vytvořených tabulek

V této části hlavního okna se zobrazují všechny tabulky aktuálního dokumentu. Aktuální zobrazená tabulka je označena modře a při přechodu na jinou tabulku si aplikace automaticky uloží provedené změny. Tabulky jsou řazeny tak, jak si je uživatel postupně vytváří. Při načtení z  $T_E$ Xovského dokumentu jsou tabulky řazeny dle pořadí ve zdrojovém dokumentu.

🔲 Tab	oulkový editor pro L	aTeX			- 0 <b>X</b>
Soul	bor Úpravy Fo	rmát Nápověda			
B	<i>I</i> <u>U</u>   normalsiz	e 🔹 🗐	🔳 🗐 🕴 📑 📑		
	A	В		Tabulky	
•	Příkaz	Ukázka		Ohraničení buňky A1:	
	\verb \tiny	\tiny{písmo}			
	\verb \script				
	\verb \footno	\footnotesiz			
	\verb \small	\small{písmo}			
	\verb \norma	\normalsize			
	AugebNarged	Marga (píama)			

Obrázek 11. Zobrazení ohraničení aktuální buňky

## 4.6. Tlačítka pro editaci řádků a sloupců tabulky

V pravé dolní části hlavního okna jsou tlačítka pro přidání řádku nebo sloupce a pro odebrání řádků či sloupců z aktuální tabulky. Přidávat lze pouze po jednom řádku či sloupci. Odebírat lze po označení více řádků či sloupců najednou.

## Závěr

Cílem této práce bylo vytvořit jednoduchý tabulkový editor pro pohodlnou tvorbu tabulek typografického systému  $I\!\!A T_{\rm E} X$ , který slouží pro vizuální návrh tabulky a její následný export do podporovaného formátu. Výsledkem je editor, který umožňuje vytvářet, editovat a ukládat tabulky do  $T_{\rm E} X u$ .

Editor je napsán v programovacím jazyku C# s balíčkem Net.framework 3.5.

V první části se práce zabývá teorií o TEX a  ${\rm \ensuremath{E}TEX}$ . Dále následuje implementace a uživatelský manuál.

Vzniklý tabulkový editor slouží především uživatelům, kteří si chtějí usnadnit práci při tvorbě dokumentů s více tabulkami v  $T_EXu$ . Díky tomuto editoru je možné snadno tvořit a editovat tabulky a také je převést do formátu  $T_EX$ .

Dalším vývojem práce by mohlo být především upravení ohraničení tabulky. Také bych se zaměřil na možnost vkládání řádků a sloupců kamkoli do tabulky. V této fázi práce je to možné pouze na konec tabulky. Možným rozšířením funkcí by mohlo být také načítání tabulek z jiných formátů.

## Conclusions

The aim of this work was to create simple table editor for comfortable creating of tables of the typographical system LATEX, which is used for visual table design, and its subsequent export to a supported format. The result of this work is an editor that lets you read, edit and save tables in TEX.

Editor is written in programming language C# with package Net.framework 3.5.

In the first part of the work deals with theories of  $T_EX$  and  $ET_EX$ . This is followed by implementation and user manual.

The resulting table editor is used mainly for users who want to facilitate the job of creating documents with multiple tables in  $T_EX$ . With this editor you can easily create and edit tables and convert them into a format  $T_FX$ .

Further development of work would be mainly a border adjustment table. I also focused on the possibility of inserting rows and columns anywhere in the table. In this phase of work it is possible only at the end of the table. A possible extension of the functions could also be reading tables from other formats.

## Reference

- [1] Autor neuveden. Drsnýúvod do  ${\not\!\!\! DT}_{E\!X} u.$  Elektronická publikace, 2005
- [2] Oetiker, T. Ne příliš stručný úvod do  $argstyle{2.5} T_E X2\epsilon$ . Elektronická publikace, 1998.
- [3] Olšák, Petr.  $T_{\!E\!}\!Xbook$ naruby. Konvoj, Brno, 2001.
- [4] Závodný, Miloslav. *Typografický systém T<sub>E</sub>X*. Matematika fyzika informatika 1999, č.9.

## 5. Obsah přiloženého CD

V této části práce je uveden stručný popis obsahu přiloženého CD.

bin/

Instalátor programu EDITOR TABULEK PRO LATEX spustitelný přímo z CD.

doc/

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

src/

Kompletní zdrojové texty programu EDITOR TABULEK PRO LATEX pro bezproblémové vytvoření spustitelných verzí programu.

### readme.txt

Instrukce pro instalaci a spuštění programu EDITOR TABULEK PRO LA-TEX, včetně požadavků pro jeho provoz.

Dále CD obsahuje:

### data/

Ukázková a testovací data pro potřeby obhajoby práce.