

Typografie a publikování

ITY 2016/2017

4. přednáška

Bohuslav Křena

`krena@fit.vutbr.cz`

Co nás dnes čeká

- **Sazba tabulek**
 - prostředí `tabbing`
 - sazba algoritmů
 - prostředí `tabular`
 - plovoucí prostředí `table`
- **Vkládání a vytváření obrázků**
 - vkládání obrázků – standardní balík `graphics`
 - kreslení obrázků v prostředí `picture`
 - vytváření obrázků pomocí **PGF** a **TikZ**
 - triky z balíčku `pstricks`
 - plovoucí prostředí `figure`

Sazba tabulek

- Pro pořadovou sazbu (pod sebe) slouží prostředí `tabbing`.
- Pro sazbu tabulek lze použít prostředí `tabular`.
- Sazbu tabulek jako plovoucích objektů zajišťuje prostředí `table`.
- \LaTeX není tabulkový procesor, ale sázecí systém.

Tabulky se v \LaTeX u sázejí pracněji než třeba v MS Office.

Se sazbou tabulek mohou pomoci specializované editory.

Například WinEdt umí vygenerovat kostru tabulky zadané velikosti (3×3).

```
\begin{tabular}{|*|*|*|}  
  \hline  
  % after \\\: \hline or \cline{col1-col2} \cline{col3-col4} ...  
  * & * & * \\  
  * & * & * \\  
  * & * & * \\  
  \hline  
\end{tabular}
```

Prostředí *tabbing*

<code>\=</code>	nastavení zarážky
<code>\></code>	přechod k další zarážce
<code>\\</code>	nový řádek
<code>\kill</code>	Řádek netiskne, jen definuje zarážky.
<code>\+</code>	posun levého okraje doprava
<code>\-</code>	posun levého okraje doleva
<code>\pushtabs</code>	Uloží nastavení zarážek.
<code>\poptabs</code>	Obnoví nastavení zarážek.

```
\begin{tabbing}
  \verb|\pushtabs|\qqquad \= Řádek netiskne, jen definuje zarážky. \kill
  \verb|\=|           \> nastavení zarážky \\
  \verb|\>|           \> přechod k další zarážce \\
  \verb|\\|           \> nový řádek \\
  \verb|\kill|        \> Řádek netiskne, jen definuje zarážky. \\
  \verb|\+|           \> posun levého okraje doprava \\
  \verb|\-|           \> posun levého okraje doleva \\
  \verb|\pushtabs|    \> Uloží nastavení zarážek. \\
  \verb|\poptabs|     \> Obnoví nastavení zarážek. \\
\end{tabbing}
```

Zdrojový text v prostředí *tabbing*

```
begin Init(S);  
    if Prsi then begin                                { Leje jako z konve. }  
        Get(S, Holinky);  
        Get(S, Destnik);  
    end else begin                                       { Svítí sluníčko. }  
        Get(S, Tenisky);  
        Get(S, SlunecniBryle);  
    end  
end.
```

```
\textbf{begin} \= Init(S); \+ \\  
  \textbf{if} \= Prsi \textbf{then begin}  
  \' \{ \emph{Leje jako z konve.} \} \+ \\  
    Get(S, Holinky); \ Get(S, Destnik); \\  
  \< \textbf{end else begin}  
  \' \{ \emph{Svítí sluníčko.} \} \\  
    Get(S, Tenisky); \ Get(S, SlunecniBryle); \-\\  
  \textbf{end} \-\\  
\textbf{end}.
```

Zdrojový text s pevným odsazením 4

```
begin Init(S);  
    if Prsi then begin                                { Leje jako z konve. }  
        Get(S, Holinky);  
        Get(S, Destnik);  
    end else begin                                       { Svítí sluníčko. }  
        Get(S, Tenisky);  
        Get(S, SlunecniBryle);  
    end  
end.
```

```
{ }{ }{ }{ }\={ }{ }{ }{ }\= \kill  
\textbf{begin} Init(S); \+ \\  
    \textbf{if} Prsi \textbf{then begin}  
    \' \{ \emph{Leje jako z konve.} \} \+ \\  
        Get(S, Holinky); \ Get(S, Destnik); \\  
    \< \textbf{end else begin}  
    \' \{ \emph{Svítí sluníčko.} \} \\  
        Get(S, Tenisky); \ Get(S, SlunecniBryle); \-\\  
    \textbf{end} \-\\  
\textbf{end}.
```

Zdrojový text s pevným odsazením 2

```
begin Init(S);  
  if Prsi then begin                                { Leje jako z konve. }  
    Get(S, Holinky);  
    Get(S, Destnik);  
  end else begin                                       { Svítí sluníčko. }  
    Get(S, Tenisky);  
    Get(S, SlunecniBryle);  
  end  
end.
```

```
{ }{ }\\={ }{ }\\= \\kill  
\\textbf{begin} Init(S); \\+ \\  
  \\textbf{if} Prsi \\textbf{then begin}  
  \\' \\{ \\emph{Leje jako z konve.} \\} \\+ \\  
    Get(S, Holinky); \\ Get(S, Destnik); \\  
  \\< \\textbf{end else begin}  
  \\' \\{ \\emph{Svítí sluníčko.} \\} \\  
    Get(S, Tenisky); \\ Get(S, SlunecniBryle); \\-\\  
  \\textbf{end} \\-\\  
\\textbf{end}.
```

Další možnosti sazby algoritmů

- prostředí `verbatim`
- balíček `listings`
- balíček `algorithms`

`http://algorithms.berlios.de/`

`http://www.ctan.org/tex-archive/macros/latex/contrib/algorithms`

- balíček `algorithm2e`

`http://www.ctan.org/tex-archive/macros/latex/contrib/algorithm2e`

- balíček `algorithmicx`

`http://www.ctan.org/tex-archive/macros/latex/contrib/algorithmicx`

Zdrojový kód je určen pro komunikaci mezi člověkem a počítačem.
Pro komunikaci mezi lidmi příliš vhodný není.

Ukázka použití balíčku *listings*

```
1  procedure PostInsert (var L:TList; El:TEl);  
2  var PomUk:TUk;  
3  begin  
4      if L.Act <> nil then begin (* je-li seznam aktivní *)  
5          new(PomUk);  
6          PomUk^.Data:=El;  
7          PomUk^.UkNasl:=L.Act^.UkNasl;  
8          L.Act^.UkNasl:=PomUk  
9      end;  
10 end;
```

```
\usepackage{listings}  
...  
\lstset{language=Pascal, numbers=left,  
        frame=shadowbox, rulesepcolor=\fymalightblue}  
\begin{lstlisting}  
procedure PostInsert (var L:TList; El:TEl);  
...  
\end{lstlisting}
```

Ukázka použití balíčku *algorithms*

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

if $n < 0$ **then**

$X \leftarrow 1/x$

$N \leftarrow -n$

else

$X \leftarrow x$

$N \leftarrow n$

end if

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow N/2$

else $\{N$ is odd $\}$

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

Ukázka použití balíčku `algorithms`

```
\begin{algorithmic}
\REQUIRE $n \geq 0 \vee x \neq 0$
\ENSURE $y = x^n$
\STATE $y \leftarrow 1$
\IF{$n < 0$}
\STATE $X \leftarrow 1 / x$
\STATE $N \leftarrow -n$
\ELSE
\STATE $X \leftarrow x$
\STATE $N \leftarrow n$
\ENDIF
\WHILE{$N \neq 0$}
\IF{$N$ is even}
\STATE $X \leftarrow X \times X$
\STATE $N \leftarrow N / 2$
\ELSE[$N$ is odd]
\STATE $y \leftarrow y \times X$
\STATE $N \leftarrow N - 1$
\ENDIF
\ENDWHILE
\end{algorithmic}
```

Prostředí *tabular*

```
\begin{tabular}[p]{sloupce}
  řádek \\
  ...
\end{tabular}
```

- Volitelný parametr `p` určuje připojení tabulky k okolnímu textu.
Implicitně středem, `t/b` (`top/bottom`) – horním/dolním okrajem.
- Parametr `sloupce` určuje počet, zarovnání a oddělení sloupců.

<code>l</code>	<code>r</code>	<code>c</code>	<code>p{šířka}</code>
<i>left</i>	<i>right</i>	<i>center</i>	<i>paragraph</i>
doleva	doprava	na střed	do bloku

- `|` – svislá čára mezi sloupci
- `@...` – jiná mezisloupcová výplň
- Buňky v řádku jsou odděleny znakem `&`.
- Vodorovná čára – `\hline` (na konci řádku)
- Částečná vodorovná čára – `\cline{x-y}` (od - do)

Příklad tabulky z IUS

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

Tabulka 1: Překračování nákladů SW projektů

```
\begin{table}[ht]
\begin{center}
\begin{tabular}{| c | c |} \hline
Překročení nákladů o & Projektů \\ \hline
méně než 20\,\% & 15,5\,\% \\
21-50\,\% & 31,5\,\% \\
51-100\,\% & 29,6\,\% \\
101-200\,\% & 10,2\,\% \\
201-400\,\% & 8,8\,\% \\
více než 400\,\% & 4,4\,\% \\ \hline
\end{tabular}
\caption{Překračování nákladů SW projektů}
\label{tabPrekroceniNakladu}
\end{center}
\end{table}
```

Úprava šířky sloupců tabulky

Velikost mezer mezi sloupci určuje délkový registr `tabcolsep`.

`\tabcolsep=6pt` (implicitní nastavení)

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

`\tabcolsep=12pt`

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

`\tabcolsep=1pt`

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

Úprava výšky řádků tabulky

- Individuálním nastavením mezery u každého řádku

`\[0.3em]`

- Vložením rozpěry (angl. *strut*), např. obdélníku nulové šířky

`\rule{0pt}{1.3em}`

- Předefinováním příkazu pro meziřádkovou vzdálenost

`\renewcommand{\arraystretch}{1.5}`

Úprava výšky řádků tabulky

`\\[0.3em]`

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

`\\rule{0pt}{1.3em}`

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

`\\renewcommand{\\arraystretch}{1.5}`

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21-50 %	31,5 %
51-100 %	29,6 %
101-200 %	10,2 %
201-400 %	8,8 %
více než 400 %	4,4 %

Další možnosti sazby tabulek

- Balík `array` nabízí různé typy sloupců.
- Balík `supertab` nebo `longtable` umožňuje sazbu na více stran.
- Balík `tabularx` slouží pro sazbu na danou šířku.
- Balík `colortbl` řeší podbarvení tabulky (*užívat s mírou*).

l	r	c	p{šířka}
<i>left</i>	<i>right</i>	<i>center</i>	<i>paragraph</i>
doleva	doprava	na střed	do bloku

```
\begin{tabular}[t]{ | l | r | c | p{15mm} | } \hline
\rowcolor[rgb]{0.7,0.8,0.9}
\texttt{l} & \texttt{r} & \texttt{c} & \verb|p{šířka}|\\ \hline
\emph{left} & \emph{right} & \emph{center} & \emph{paragraph} \\ \hline
doleva & doprava & na střed & do bloku \\ \hline
\end{tabular}
```

Plovoucí prostředí *table*

- Uprostřed tabulky nemůže dojít ke zlomu stránky.
Jenže co když se tabulka na aktuální stránku nevejde?
 1. Tabulka bude na nové stránce.
 2. Na předchozí straně zůstane nevyužitý prostor.

- Řešení nabízí plovoucí prostředí **table**:

```
\begin{table}[umístění]
  \begin{tabular}
    ...
  \end{tabular}
  \caption Název (popis) tabulky
\end{table}
```

- Požadované (přednostní) umístění
 - h (*here*) – do místa zdrojového textu
 - t (*top*) – na horní část stránky
 - b (*bottom*) – a dolní část stránky
 - p (*page*) – na samostatnou stránku (s dalšími plovoucími objekty)

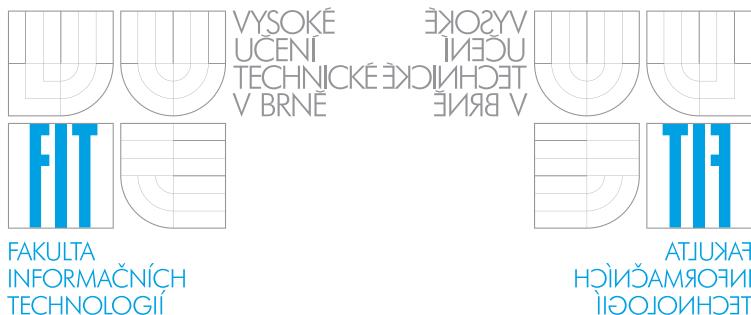
Obrázky v \LaTeX u

- Kategorie obrázků
 - **Perokresby** – jednotlivé tahy perem
 - **Autotypie** – různé odstíny a barvy
- Formáty obrázků
 - **Vektorový formát** – soubor grafických primitiv
 - **Rastrový formát** – informace o každém bodu (*pixelu*)
- Vkládání obrázků
 - Balík **graphics** – vložení již hotových obrázků
 - Prostředí **picture** – přímé kreslení vektorových obrázků
 - Systémy **PGF** a **TikZ** – přímé kreslení vektorových obrázků
 - METAFONT – součást \TeX u pro vektorové obrázky (písmo)
 - Využití vlastností programu pro zpracování souboru `.dvi`
 - ...

Standardní balík *graphics*

- Pro vkládání obrázků slouží příkaz `\includegraphics{obrázek}`.
- Pro vložení obdélníkového výřezu z obrázku je příkaz `\includegraphics*[x1,y1][x2,y2]{obrázek}`.
- `scalebox` – změna velikosti
- `resizebox` – vyplnění do připravené oblasti
- `rotatebox` – otočení o zadaný úhel
- `reflectbox` – zrcadlový obraz

```
\scalebox{0.33}{\includegraphics{logoFIT.eps}}  
\reflectbox{\includegraphics{logoFIT.eps}} }
```



Standardní balík *picture*

- Umožňuje přímé *kreslení* vektorových obrázků.
 - + výborné sladění s celým dokumentem
 - značně pracné a náročné na představivost
- Obrázek je realizován jako prostředí
 - `\begin{picture}(šířka,výška)(x-posun,y-posun)`
definice obrázku ...
`\end{picture}`
 - Souřadný systém začíná v bodě (0,0), který je vlevo dole.
 - Posun obrázku vůči bodu (0,0) je nepovinný.
- Objekty se do obrázku vkládají příkazem `\put`
`\put(x,y){objekt}`
- Opakované vkládání objektu provede příkaz `\multiput`
`\multiput(x,y)(dx,dy){počet}{objekt}`

Objekty z balíku *picture*


- **Text** – s formátováním pro jeden řádek
- **Úsečky** – $\backslash\text{line}(x,y)\{\text{délka}\}$ kde $x,y \in \langle -6;6 \rangle \cap \mathbb{Z}$

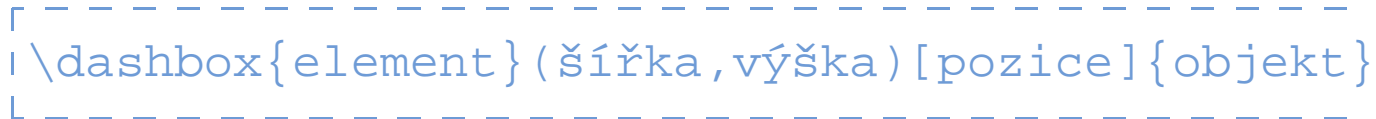

 $\backslash\text{put}(0,0)\{\backslash\text{line}(1,0)\{100\}\}$

- **Vektory** – $\backslash\text{vector}(x,y)\{\text{délka}\}$ (úsečka zakončená šipkou)


 $\backslash\text{put}(0,0)\{\backslash\text{vector}(1,0)\{100\}\}$

- **Rámečky**

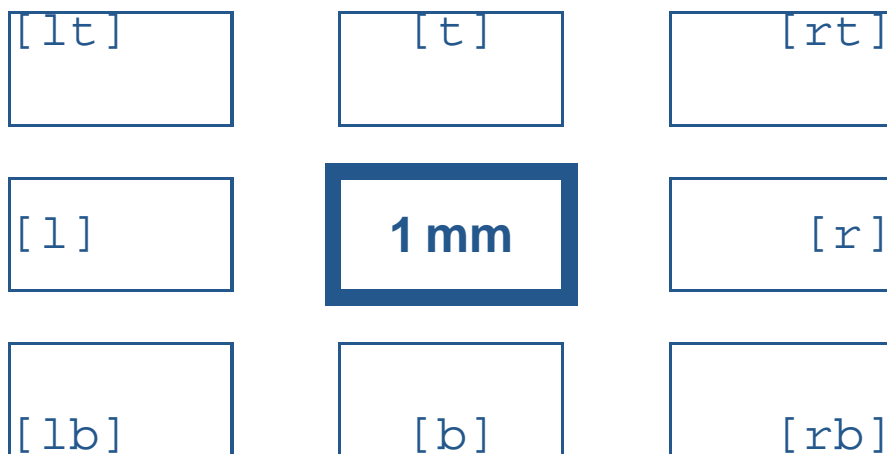

 $\backslash\text{framebox}(\text{šířka},\text{výška})[\text{pozice}]\{\text{objekt}\}$


 $\backslash\text{dashbox}\{\text{element}\}(\text{šířka},\text{výška})[\text{pozice}]\{\text{objekt}\}$

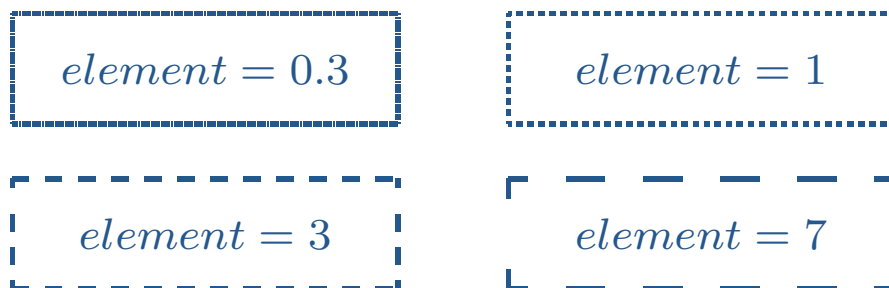
$\backslash\text{makebox}(\text{šířka},\text{výška})[\text{pozice}]\{\text{objekt}\}$

Možnosti balíku *picture*

- Nastavení síly čáry `\linethickness{míra}`,
`\thinlines` (0.4 pt), `\thicklines` (0.8 pt)
- Umístění vnitřního objektu v rámečku



- Možnosti čárkovaných rámečků



Objekty z balíku *picture*

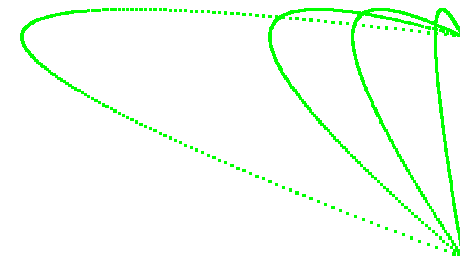
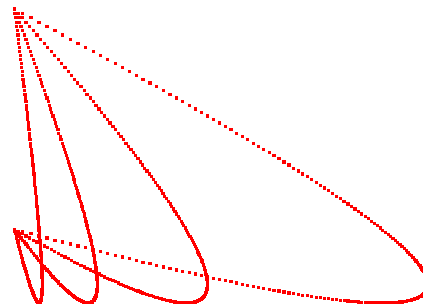
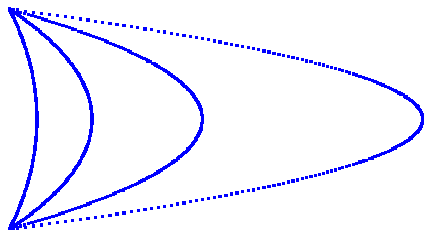
- **Kroužky** – `\circle{průměr}` nebo `\circle*{průměr}`



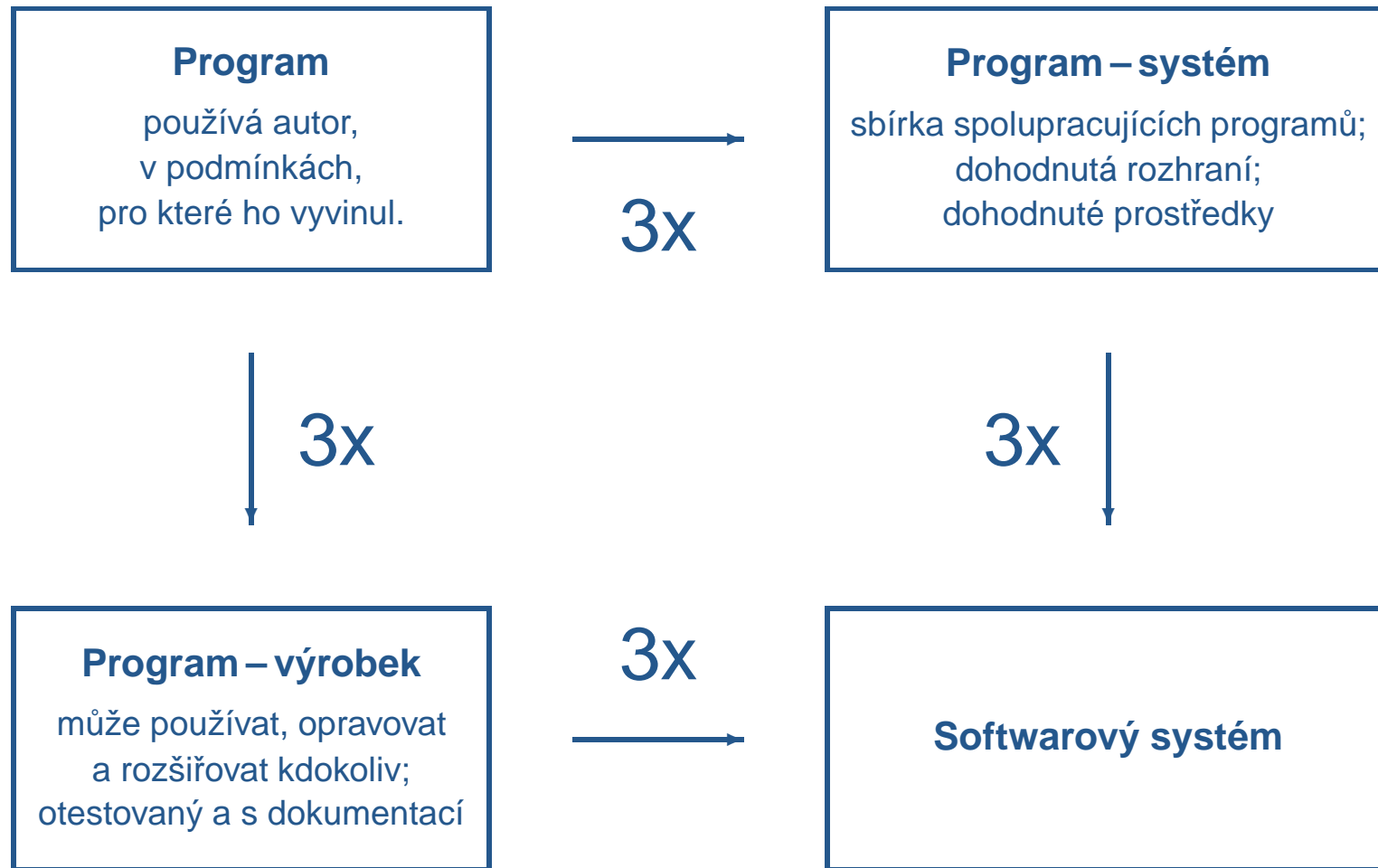
- **Ovály** – `\oval(šířka,výška)[část]` (l-left, r-right, t-top, b-bottom)



- **Béziérový křivky** – `\qbezier[N](Ax,Ay)(Bx,By)(Cx,Cy)`
... křivka z A do C , řídicí bod B , složená z N bodů
... pozor na časovou náročnost vykreslování



Obrázek z IUS



Obrázek 1: Program × SW systém

Obrázek z IUS – 1. část

```
\begin{figure}[ht]
\begin{picture}(320,200)
  \linethickness{1pt}
  \put(0,140){\framebox(110,60){
    \shortstack{\textbf{Program} \ \ [1mm]
      \tiny používá autor, \
      \tiny v podmínkách, \
      \tiny pro které ho vyvinul.}}}
  \put(0,0){\framebox(110,60){
    \shortstack{\textbf{Program\,--\,výrobek} \ \ [1mm]
      \tiny může používat, opravovat \
      \tiny a rozšiřovat kdokoliv; \
      \tiny otestovaný a s dokumentací}}}
  \put(190,140){\framebox(130,60){
    \shortstack{\textbf{Program\,--\,systém} \ \ [1mm]
      \tiny sbírka spolupracujících programů; \
      \tiny dohodnutá rozhraní; \
      \tiny dohodnuté prostředky}}}
  \put(190,0){\framebox(130,60){
    \shortstack{\textbf{Softwarový systém}}}}
```

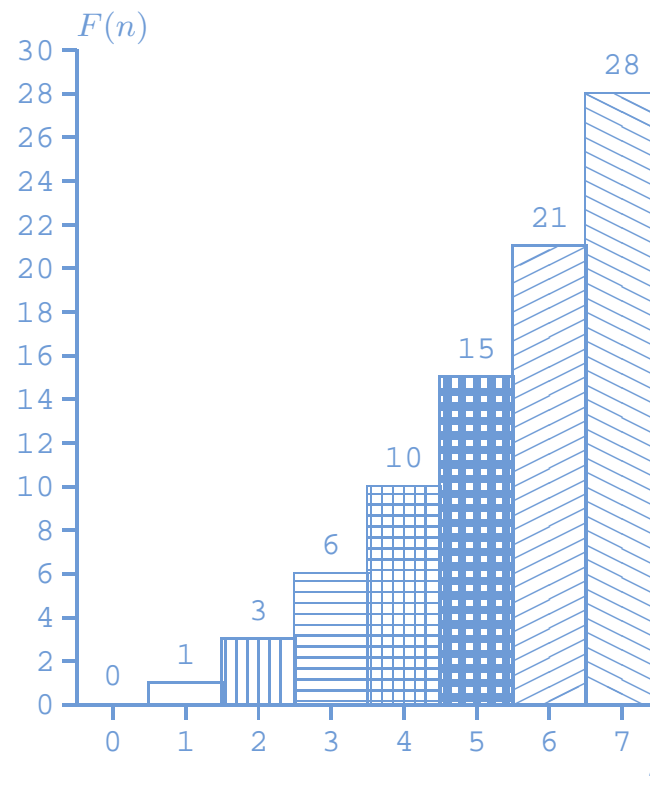
Obrázek z IUS – 2. část

```
% Šipka mezi "Program" a "Program\,--\,výrobek"
\put(55,120){\vector(0,-1){40}}
\put(75,100){\makebox(0,0){\large 3x}}
% Šipka mezi "Program" a "Program\,--\,systém"
\put(130,170){\vector(1,0){40}}
\put(150,150){\makebox(0,0){\large 3x}}
% Šipka mezi "Program\,--\,systém" a "Softwarový systém"
\put(255,120){\vector(0,-1){40}}
\put(235,100){\makebox(0,0){\large 3x}}
% Šipka mezi "Program\,--\,výrobek" a "Softwarový\,--\,systém"
\put(130,30){\vector(1,0){40}}
\put(150,50){\makebox(0,0){\large 3x}}
\end{picture}

\caption{Program $\times$ SW systém}
\label{picProgramVSSoftware}
\end{figure}
```

Rozšíření balíku *picture*

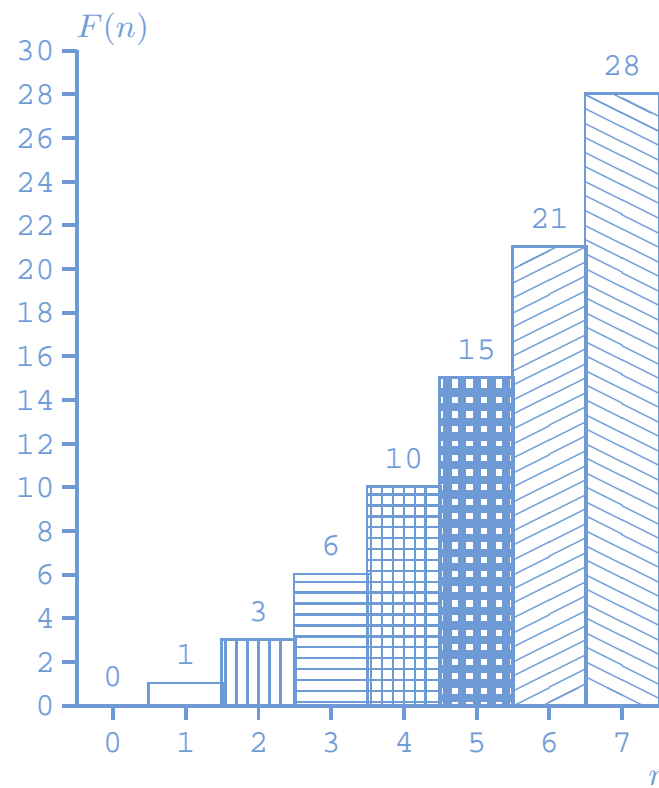
- **pict2e** – odstraňuje omezení (šikmé čáry, kroužky, tloušťky ...)
- **epic** – lomené, čárkované a tečkované úsečky
- **eepic** – oblouky, elipsy, tloušťky šikmých čar
- **trees** – stromy
- **curves** – křivky
- **bar** – sloupcové grafy



Obrázek 2: Ukázka sloupcového grafu

Ukázka sloupcového grafu

```
\begin{barendv}  
\setwidth{20}  
\setdepth{0}  
\setstretch{6}  
\setnumberpos{up}  
\setxaxis{0}{7}{1}  
\setxname{$n$}  
\setyaxis{0}{30}{2}  
\setyname{$F(n)$}  
\bar{0}{0}  
\bar{1}{1}  
\bar{3}{2}  
\bar{6}{3}  
\bar{10}{4}  
\bar{15}{5}  
\bar{21}{6}  
\bar{28}{7}  
\end{barendv}
```



Obrázek 3: Ukázka sloupcového grafu

Grafické systémy PGF a TikZ ●

- PGF (**P**ortable **G**raphics **F**ormat) – balíček maker pro T_EX
- TikZ – rozhraní pro použití PGF v L^AT_EXu
- výrazně větší možnosti kreslení než prostředí `picture`
- **Manuál k verzi 2.00 má 560 stran!**
- Vytvořeno příkazem
`\tikz \fill[red!70!blue] (0,0) circle (0.25em);`

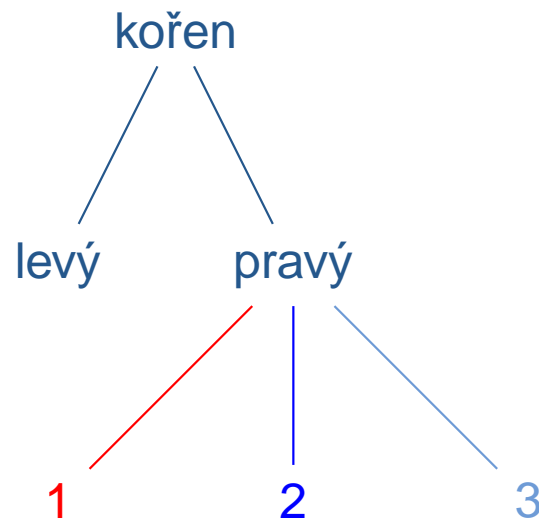
TikZ – ukázka

```
\usepackage{tikz}
...
\begin{tikzpicture}
  \draw[thick,rounded corners=8pt]
    (0,0) -- (0,2) -- (1,3.25) -- (2,2) -- (2,0)
    -- (0,2) -- (2,2) -- (0,0) -- (2,0);
\end{tikzpicture}
```



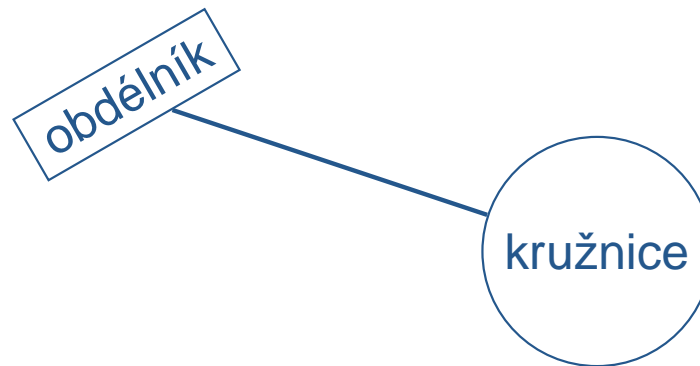
TikZ – speciální podpora pro stromy

```
\begin{tikzpicture}
  \node {kořen}
    child {node {levý}}
    child {node {pravý}}
      child[red] {node {1}}
      child[blue] {node {2}}
      child[fymalightblue] {node {3}}
  };
\end{tikzpicture}
```



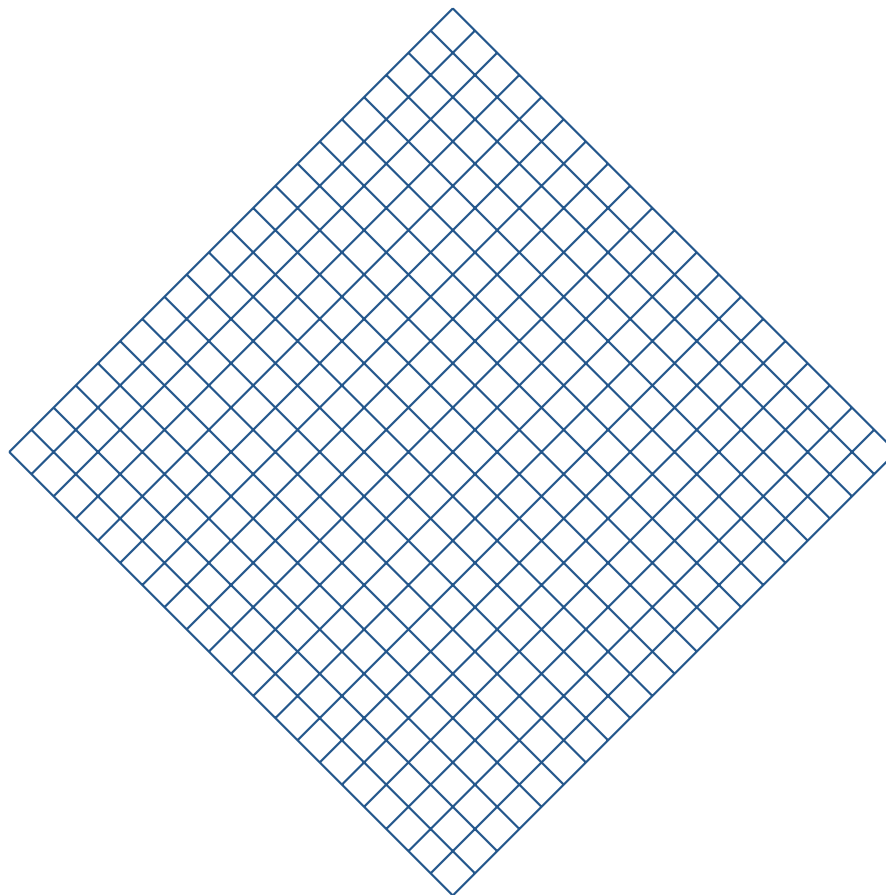
TikZ – chytré nalezení hranic objektů

```
\begin{tikzpicture}  
  \path (0,0) node(a) [rectangle,rotate=45,draw] {obdélník}  
        (3,-1) node(b) [circle,draw] {kružnice};  
  \draw[thick] (node cs:name=a) -- (node cs:name=b);  
\end{tikzpicture}
```



TikZ–mřížka

```
\tikz[rotate=45] \draw[step=2mm] (0,0) grid (4,4);
```



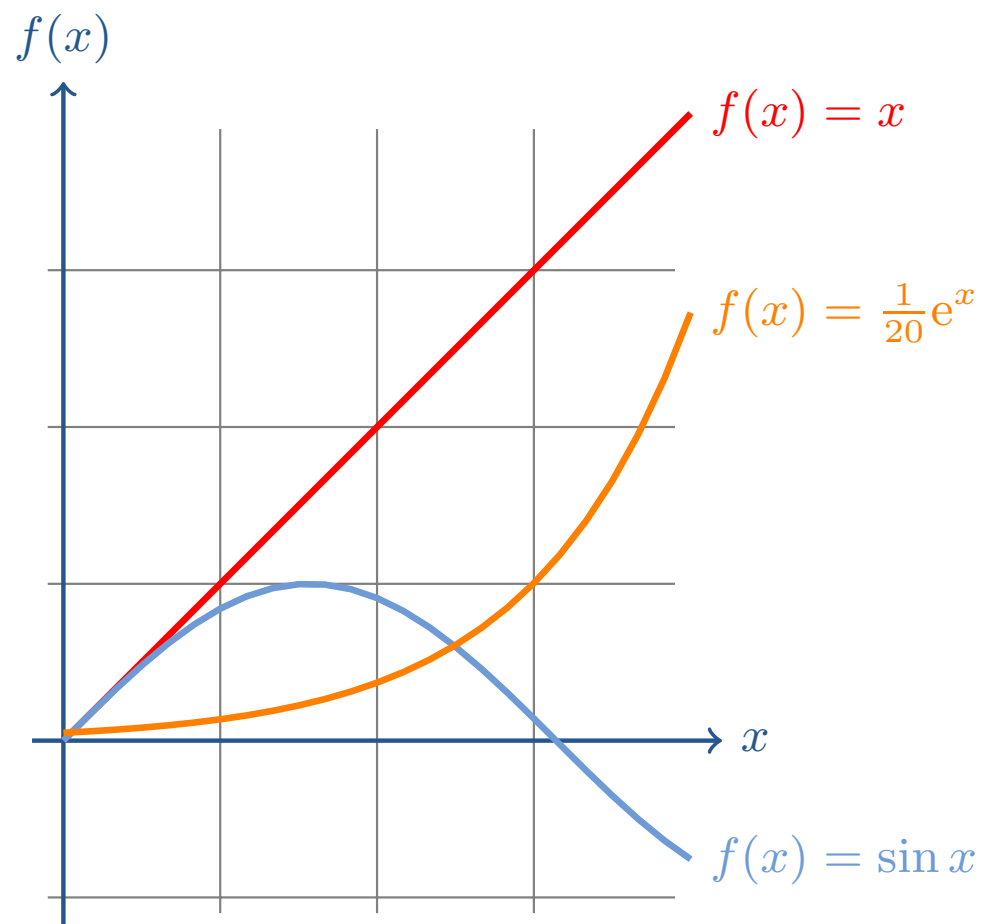
TikZ – vykreslování funkcí

```
\begin{tikzpicture}[domain=0:4]
  \draw[thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);

  \draw[thick,->] (-0.2,0) -- (4.2,0) node[right] {$x$};
  \draw[thick,->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};

  \draw[very thick,color=red]
    plot (\x,\x)
    node[right] {$f(x) = x$};
  \draw[very thick,color=fymalightblue]
    plot (\x,{sin(\x r)})
    node[right] {$f(x) = \sin x$};
  \draw[very thick,color=orange]
    plot (\x,{0.05*exp(\x)})
    node[right] {$f(x) = \frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}
```

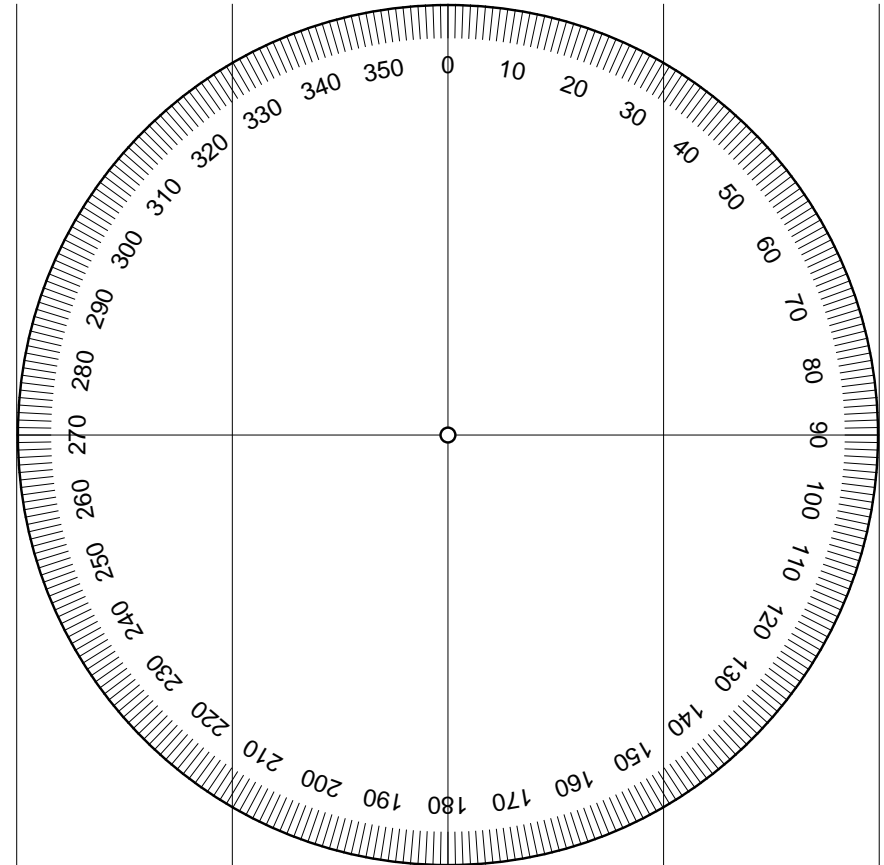
TikZ – vykreslování funkcí



Balík *pstricks*

- Využívá jazyk PostScript a příkaz `\special`.
- Zpracování při převodu z `dvi` do `ps`.
- Příklad nade vše.

```
% Herbert Voss
\documentclass[a4paper]{article}
\usepackage{pstricks}
\usepackage{multido}
\SpecialCoor
\begin{document}
\begin{pspicture*}(-5,-5)(5,5)
  \pscircle{5}
  \multido{\iA=0+1}{360}{\psline[linewidth=0.1pt](4.6;\iA)(5;\iA)}
  \multido{\iA=90+-10,\iB=0+10}{36}{\rput{-\iB}(4.3;\iA){\iB}}
  \multido{\rA=-5+2.5}{5}{\psline[linewidth=0.2pt](\rA,-5)(\rA,5)}
  \psline[linewidth=0.2pt](-5,0)(5,0)
  \pscircle[fillcolor=white,fillstyle=solid]{0.1}
\end{pspicture*}
\end{document}
```



Plovoucí prostředí *figure*

Je obdobou plovoucího prostředí `table` používaného pro tabulky.

```
\begin{figure}[ht]  
  \begin{picture}(320,200)  
    ...  
  \end{picture}  
  \caption{Program $\times$ SW systém}  
  \label{picProgramVSSoftware}  
\end{figure}
```

- Požadované (přednostní) umístění obrázku
 - `h` (*here*) – do místa zdrojového textu
 - `t` (*top*) – na horní část stránky
 - `b` (*bottom*) – na dolní část stránky
 - `p` (*page*) – na samostatnou stránku (s dalšími plovoucími objekty)
- Odkazovat se lze příkazem `\ref{picProgramVSSoftware}`.
- Seznam obrázků vygeneruje příkaz `\listoffigures`.

Použité zdroje

- J. Rybička: \LaTeX pro začátečníky
- \LaTeX Graphics
<http://www.ursoswald.ch/LaTeXGraphics/overview/overview.html>
- Ichimusai LaTeX Page
<http://www.ichimusai.org/latex/>
- PGF and TikZ – Graphic systems for \TeX
<http://pgf.sourceforge.net/>
- PSTricks web site
<http://tug.org/PSTricks/main.cgi/>
- B. Křena, R. Kočí: Úvod do softwarového inženýrství, studijní opora
- J. M. Honzík: Algoritmy, studijní opora, 2011