

La classe `guidatematica.dtx`*

GuIT[†]

12 dicembre 2012

Sommario

Questa classe serve per comporre testi da pubblicare nella collana Guide Tematiche del GuIT. Si appoggia alla classe `memoir` ma include molte personalizzazioni e i pacchetti necessari.

1 Caratteristiche della classe

La classe `guidatematica.cls` serve per comporre libri in italiano con le seguenti specifiche:

- La classe di supporto è la classe `memoir.cls` facente parte del sistema TeX in qualsiasi distribuzione completa.
- Tutte le opzioni inserite nell'istruzione `\documentclass` vengono trasferite integralmente alla classe `memoir`. Solo l'opzione `ipertesto` è specifica della classe; serve all'utente per caricare il pacchetto `hyperref` con alcune personalizzazioni, ritardandone il caricamento al momento di eseguire il comando `\begin{document}`; questo dipende dal fatto che il pacchetto `hyperref` deve sempre venire caricato per ultimo. Non è attivata di default, perché non tutte le guide, benché previste per essere lette prevalentemente sullo schermo del proprio PC, hanno bisogno di collegamenti ipertestuali interni ed esterni. Quindi caricare questo pacchetto è una scelta esclusiva dell'utente.
- La classe funziona sia con pdfLaTeX sia con XeLaTeX; le impostazioni sono prefissate e i margini per ulteriori personalizzazioni sono ridotti al minimo; lo scopo della classe è quello di dare una veste comune alle guide tematiche della collana, facendo sì che siano riconoscibili a prima vista grazie alla loro uniformità di composizione.
- La classe riconosce da sola il motore di composizione e imposta direttamente quanto occorre nella forma richiesta dal programma di composizione.
- La codifica di entrata deve essere `utf8`; per XeLaTeX essa è implicita; con pdfLaTeX essa viene preimpostata e non è possibile con questa classe specificare una codifica differente.

*Version number v.0.6c; ultima revisione 2012/12/12.

[†]e-mail: www.guitex.org

- La codifica dei font di uscita con pdfLaTeX è T1 e non è possibile specificare una codifica differente, cosa che sarebbe quanto mai sgradevole per comporre in italiano dove sono presenti molte lettere accentate. Per altro se fosse necessario comporre qualcosa in una lingua che usa un alfabeto diverso, basta caricare i suoi pacchetti specifici; per scrivere una parola o due basta caricare l'encoding dei font specifici in modo diretto: per esempio: se si deve scrivere una parola in russo, basta caricare esplicitamente il file del suo encoding con `\input{t2enc.def}` e poi specificare la codifica T2 con `\fontencoding{T2}` seguito dai comandi per scegliere famiglia, serie, forma e corpo del font cirillico che si vuole usare.
- Con pdfLaTeX il pacchetto `babel` viene caricato con la sola opzione `italian`; nei rari casi in cui sia necessario usare anche altre lingue, esse vanno specificate come opzione alla classe, perché non si può chiamare `babel` due volte con opzioni diverse. Le opzioni della classe sono globali e vengono passate automaticamente a tutti i pacchetti che sanno che cosa farne.
- Con XeLaTeX viene caricato il pacchetto `polyglossia` e l'italiano viene impostato come la lingua principale; altre lingue possono venire specificate mediante il comando `\setotherlanguages` o con lingue particolari con `\setotherlanguage` con le opzioni specifiche per la lingua. Se questa richiede un alfabeto particolare bisogna ricordarsi di specificarne una famiglia di font che contenga l'alfabeto desiderato.
- Sono stati introdotti molti nuovi comandi e sono stati definiti molti nuovi ambienti. Per maggiori informazioni si rimanda al documento *guidatematica-doc.pdf* distribuito con il “kit” `GuidaTematica.zip`.

2 Codice commentato

Le due righe contenenti il formato e l'identificazione della classe sono già state inserite prima; ora possiamo passare alla gestione delle opzioni che vengono tutte passate alla classe `memoir`

```
1 \DeclareOption{ipertesto}{%
2 \AtBeginDocument{\RequirePackage[colorlinks,linkcolor={blue},
3 citecolor={blue!80!black},urlcolor={blue}]{hyperref}}
4 }
5 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
6 \ProcessOptions*\relax
7
8 \LoadClassWithOptions{memoir}
```

Poi si specificano le informazioni per il formato di carta che sono richieste da `memoir`.

```
9 \pdfpagewidth=\paperwidth
10 \pdfpageheight=\paperheight
11
12 \stockwidth=\paperwidth
13 \stockheight=\paperheight
```

Si carica subito il pacchetto `ifxetex` per poter distinguere nelle impostazioni quelle da eseguire con il programma `xetex` rispetto a quelle da eseguire col programma `pdftex`.

```
14 \RequirePackage{ifxetex}
```

Con `xetex` si carica il pacchetto `fontspec` e si impostano le varie forme dei font Latin Modern OpenType, generando anche i comandi necessari per la compatibilità con i comandi da usare con `pdftex` quando si invoca il pacchetto `cfr-lm` per la gestione estesa dei font Latin Modern Type 1. Si provvede nei due casi di caricare i pacchetti necessari per la gestione della lingua italiana.

```
15 \ifxetex
16 \RequirePackage{fontspec}
17 \setmainfont[Ligatures=TeX,Numbers=OldStyle]{Latin Modern Roman}
18 \setsansfont[Ligatures=TeX,Numbers=OldStyle]{Latin Modern Sans}
19 \setmonofont[Numbers=Lining]{Latin Modern Mono}
20 \newfontfamily{\tvfamily}[Numbers=Lining]{Latin Modern Mono Prop}
21 \newfontfamily{\scfamily}%
22   [Ligatures=TeX,Numbers=OldStyle]{Latin Modern Roman Caps}
23 \DeclareRobustCommand\scshape{\scfamily}
24 \DeclareTextFontCommand{\textttm}{\ttfamily}
25 \DeclareTextFontCommand{\textttv}{\tvfamily}
26 \DeclareTextFontCommand{\textsc}{\scshape}
27 \def\textl#1{{\addfontfeature{Numbers=Lining}#1}}
28 \RequirePackage{unicode-math}
29 \setmathfont{Latin Modern Math}
30 %
31 \RequirePackage{polyglossia}
32 \setmainlanguage{italian}
33 \else
34 \RequirePackage[utf8]{inputenc}
35 \RequirePackage[T1]{fontenc}
36 \RequirePackage[italian]{babel}
37 \RequirePackage[tt={oldstyle=false,tabular,monowidth}]{cfr-lm}
38 \fi
```

Si caricano poi i pacchetti ritenuti necessari; non si è caricato espressamente il pacchetto `hyperref` perché va caricato per ultimo. L'utente potrebbe avere bisogno di caricare ulteriori pacchetti nel preambolo, quindi è meglio che `hyperref` venga caricato dall'utente dopo i pacchetti di cui ha bisogno. Per fare ciò basta specificare l'opzione `ipertesto` alla classe che automaticamente carica il pacchetto `hyperref` come ultima cosa al momento di iniziare la composizione della guida tematica che si desidera comporre.

In questa versione non viene caricato il pacchetto `guit.sty` perché è incompatibile con `XeLaTeX`; certo lo si sarebbe potuto caricare nel ramo `\else` del test `\ifxetex` e la definizione del logo del gruppo in forma grafica (immagine pdf) la si sarebbe potuta limitare all'uso con `XeLaTeX` siccome alla data attuale il pacchetto `guit.sty` è sottoposto a revisione per l'uso con `XeLaTeX` è molto meglio usare temporaneamente l'immagine in entrambi i casi e modificare questa classe solo con l'aggiunta di un nuovo pacchetto quando `guit.sty` sarà diventato compatibile con entrambi i programmi di composizione. Il logo tondo, il “timbro”, del GuIT viene invece allegato come immagine, visto che si tratta di una variante nella quale le lettere della scritta attorno al bordo non appaiono mai capovolte.

```
39 \RequirePackage{graphicx}
40 \RequirePackage{metalogo}
41 \RequirePackage{pict2e}[2009/06/01]
42 \RequirePackage{microtype}
```

```

43 \RequirePackage{etoolbox}
44 \RequirePackage{xcolor}
45 \RequirePackage{natbib}
46 \RequirePackage{multicol}
47 \RequirePackage{imakeidx}
48 \RequirePackage{fancyvrb}
49 \RequirePackage{afterpage}
50 \RequirePackage{enumitem}
51 \RequirePackage{listings}
52 \lstloadlanguages{[LaTeX]TeX}

```

Il disegno della pagina viene fatto con il pacchetto `canoniclayout` piuttosto classico, ma adattissimo sia al formato A4, sia a quello B5.

```

53 \RequirePackage{canoniclayout}
54 \marginparwidth=0.8\dimexpr\evensidemargin+1in-\marginparsep\relax

```

Viene ora recuperato il comando `\GetFileInfo` dal pacchetto `doc.sty` in modo da sfruttare la possibilità di recuperare il dati di un file che contenga in testa uno dei comandi `\ProvidesClass` oppure `\ProvidesPackage` oppure `\ProvidesFile`, completo di argomento obbligatorio e facoltativo nel formato previsto da questi comandi. L'argomento di `\GetFileInfo` è il nome del file di cui si vogliono estrarre le informazioni, specificato completo di estensione. Il file deve essere già stato caricato prima di usare questo comando; usandolo si recuperano nelle macro `\filename`, `\filedate` e `\fileversion` le informazioni specifiche, che poi possono venire usate come meglio si crede.

```

55 \def\GetFileInfo#1{%
56   \def\filename{#1}%
57   \def\@tempb##1 ##2 ##3\relax##4\relax{%
58     \def\filedate{##1}%
59     \def\fileversion{##2}%
60     \def\fileinfo{##3}}%
61   \edef\@tempa{\csname ver@#1\endcsname}%
62   \expandafter\@tempb\@tempa\relax? ? \relax\relax}

```

Segue ora il comando `\setcopymark` che può venire usato in diversi modi; lo si può usare per inserire il marchio di copyright e il nome del detentore del suo diritto. Ma lo si può usare anche per marcare lo stato di avanzamento di bozze successive. Il suo output viene scritto nel margine esterno della pagina. Per altri usi di dichiarazioni di diritti si ritiene che sia meglio usare l'Introduzione invece del retrofrontespizio, visto che queste guide vengono per lo più composte solo fronte (almeno così si suggerisce) per una lettura confortevole a monitor.

```

63 \let\@copymark\@empty % Di default è vuoto
64 \newcommand*\setcopymark[1]{\gdef\@copymark{#1}}
65 \newcommand*\insertcopymark{%
66 \begin{picture}(0,0)\unitlength=1mm\relax\if@openright
67 \put(10,9.5){\rotatebox{90}{\makebox(0,0)[1]{\@copymark}}}%
68 \else
69 \ifodd\value{page}\put(10,9.5){\rotatebox{90}{\@copymark}}{%
70   \makebox(0,0)[1]{\@copymark}}}\else
71 \put(-10,9.5){\rotatebox{90}{\makebox(0,0)[1]{\@copymark}}}\fi\fi
72 \end{picture}%
73 }

```

Si definisce il comando per inserire il layout della pagina sotto il testo di una pagina specifica. Vengono specificati i comandi sia per la pagina destra sia per

quella sinistra, anche se si raccomanda di comporre con l'opzione `oneside`. Questi comandi sono completamente parametrizzati alle dimensioni del foglio rifilato e valgono per qualunque formato.

```

74 \def\cblayoutsinistro{%
75 \dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
76 \dimen258=\dimexpr\evensidemargin+1in\relax
77 \dimen262=1mm\relax
78 \dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
79 \dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
80     -\headsep-\textheight\relax
81 \begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
82 \put(0,0){\unitlength=\p@
83 \put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
84 \color{red}%
85 \put(\strip@pt\dimen258,\strip@pt\dimen264){%
86     \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
87 \Line(0,0)(\strip@pt\paperwidth,\strip@pt\paperheight)%
88 \put(\strip@pt\dimexpr\paperwidth/2\relax,%
89     \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
90     {\circle{\strip@pt\paperwidth}}}%
91 }
92 \end{picture}}
93
94 \def\cblayoutdestro{%
95 \dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
96 \dimen258=\dimexpr\oddsidemargin+1in\relax
97 \dimen262=1mm\relax
98 \dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
99 \dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
100     -\headsep-\textheight\relax
101 \begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
102 \put(0,0){\unitlength=\p@
103 \put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
104 \color{red}%
105 \put(\strip@pt\dimen258,\strip@pt\dimen264){%
106     \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
107 \Line(0,\strip@pt\paperheight)(\strip@pt\paperwidth,0)%
108 \put(\strip@pt\dimexpr\paperwidth/2\relax,%
109     \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
110     {\circle{\strip@pt\paperwidth}}}%
111 }
112 \end{picture}}

```

Si specificano ora i comandi di configurazione per la classe `memoir`. Non si vuole il maiuscolo nelle testatine, perché verrà usato il maiuscoletto; non si vogliono filetti per separare la testatina e il piedino dal testo. Si vuole però che il `\@copymark` sia sempre presente nel piedino: vuol dire che se questa macro è vuota non viene stampato niente.

```

113 \nouppercasseheads
114 %\renewcommand{\footruleheight}{\normalrulethickness}
115 \renewcommand{\footruleskip}{0pt}
116 %\makefootrule{empty}{0pt}{0pt}{0pt}
117 \makeevenfoot{plain}{\@insertcopymark}{\textsc{thepage}}{}
118 \makeoddfoot{plain}{}{\textsc{thepage}}{\@insertcopymark}

```

```

119 \makeevenfoot{headings}{\@insertcopymark}{-}{-}
120 \makeoddfoot{headings}{-}{-}{\@insertcopymark}
121 \makeevenfoot{myheadings}{\@insertcopymark}{-}{-}
122 \makeoddfoot{myheadings}{-}{-}{\@insertcopymark}
123 \makeheadrule{headings}{\textwidth}{0pt}
124 \makeheadrule{myheadings}{\textwidth}{0pt}

```

Nelle testatine il numero di pagina viene composto in maiuscoletto: la cosa non è importantissima, ma all'occorrenza i numeri di pagina romani minuscoli vengono composti col minuscolo del maiuscoletto e non hanno quell'aspetto orribile dei numeri romani composti in tondo minuscolo.

```

125 \makeevenhead{headings}%
126   {{\textsc{\thepage}\quad}}{\textsc{\small\leftmark}}{-}{-}
127 \makeoddhead{headings}%
128   {-}{\textsc{\small\rightmark}}{\quad\textsc{\thepage}}{-}
129 \makeevenhead{myheadings}%
130   {{\textsc{\thepage}\quad}}{\textsc{\small\leftmark}}{-}{-}
131 \makeoddhead{myheadings}%
132   {-}{\textsc{\small\rightmark}}{\quad\textsc{\thepage}}{-}

```

Vengono definiti i comandi per “decorare” le intestazioni delle testatine.

```

133 \makesmarks{headings}{%
134 \createmark{chapter}{both}{shownumber}{\@chapapp\space}{.\quad}
135 \createmark{section}{right}{shownumber}{\mathsection$,}{\quad}
136 \renewcommand*\indexmark{\markboth{\indexname}{\indexname}}}

```

Vengono definiti gli stili di pagina modificati con il layout disegnato sotto. Non sono da usare sistematicamente, ma probabilmente come argomenti di `\thispagestyle`.

```

137 \makepagestyle{headingslayout}
138 \makeevenhead{headingslayout}{\cblayoutsinistro
139   \textsc{\thepage}\quad\textsc{\small\leftmark}}{-}{-}
140 \makeevenfoot{headingslayout}{\@insertcopymark}{-}{-}
141 \makeoddhead{headingslayout}{\cblayoutdestro
142   \hfill\textsc{\small\rightmark}\quad\textsc{\thepage}}{-}{-}
143 \makeoddfoot{headingslayout}{-}{-}{\@insertcopymark}

```

Viene definito lo stile per comporre la pagina iniziale dei capitoli; si compone il titolo del capitolo in maiuscoletto e il numero del capitolo molto grande fuori nel margine esterno. Si imposta questo stile al momento dell'inizio del documento per contrastare eventuali ulteriori modifiche inserite nel preambolo.

```

144 \makechapterstyle{guidatematica}{%
145   \renewcommand*\chapnumfont{%
146     \fontshape{it}\fontsize{40}{40}\selectfont}
147   \renewcommand*\printchaptername{-}
148   \renewcommand*\chapternamenumb{-}
149   \renewcommand*\chaptitelfont{%
150     \fontsize{18}{16}\scshape}% sterlineato
151   \renewcommand*\printchapternum{%
152     \noindent\rlap{\makebox[\textwidth]{r}}{\rlap{\hskip10pt
153       \chapnumfont \thechapter}}}\printchaptertitle}
154   \renewcommand*\afterchapternum{-}
155 }
156 \AtBeginDocument{\chapterstyle{guidatematica}}

```

Vengono ora fissati alcuni parametri stilistici per le pagine che iniziano una “parte”, anche se si ritiene che le guide tematiche debbano essere sufficientemente succinte da non richiedere di essere divise in parti.

```
157 \renewcommand*\partnamefont{\normalfont\large\scshape}
158 \renewcommand*\partnumfont{\normalfont\large\scshape}
159 \renewcommand*\parttitlefont{\normalfont\huge\scshape}
```

Ora le spaziature prima, dopo, a destra e a sinistra dei titoli dei comandi di sezionamento minori. Inoltre si definiscono gli stili dei titolini e altre informazioni stilistiche per le parti e i capitoli.

```
160 \setbeforesecskip{-3.5ex plus-1ex minus-0.2ex}
161 \setbeforesubsecskip{-3ex plus-1ex minus-0.2ex}
162 \setbeforesubsubsecskip{-2.5ex plus-1ex minus-0.2ex}
163 \setbeforeparaskip{1\onelineskip plus1ex minus0.2ex}
164 \setbeforesubparaskip{1\onelineskip plus1ex minus0.2ex}
165 \setaftersecskip{1.5ex plus0.2ex}
166 \setaftersubsecskip{1.5ex plus0.2ex}
167 \setaftersubsubsecskip{1.5ex plus0.2ex}
168 \setafterparaskip{-1em}
169 \setaftersubparaskip{-1em}
170 \setsubparaindent{\parindent}
171
172 \setsecheadstyle{\large\scshape\raggedright}
173 \setsubseheadstyle{\large\scshape\raggedright}
174 \setsubsubseheadstyle{\large\scshape\raggedright}
175 \setparaheadstyle{\small\scshape}
176 \setsubparaheadstyle{\small\scshape}
177
178 \aliaspagestyle{part}{empty}
179 \aliaspagestyle{chapter}{empty}
180 \copypagestyle{titlepage}{headings}
181 \renewcommand\cftpartpagefont{\scshape}
182 \renewcommand\cftpartfont{\large\scshape}
183 \renewcommand\cftchapterfont{\large\scshape}
184 \renewcommand\cftchapterpagefont{\scshape}
185 \renewcommand*\cftchapterfillnum}[1]{%
186     {\cftchapterleader}\nobreak
187     \cftchapterformatpnum{#1}%
188     \cftchapterafterpnum\par\nobreak}
189 \setlength{\cftbeforechapterskip}{1.0em \@plus 2\p@}
```

Grazie alle funzionalità di memoir si ridefiniscono anche le modalità di comporre le didascalie.

```
190 \captiondelim{\quad}
191 \captionnamefont{\small\scshape}
192 \captiontitlefont{\small}
193 \captionstyle[centering]{ }
194 \hangcaption
195 \captionwidth{\dimexpr\textwidth-2\parindent\relax}\changepcaptionwidth
```

Qui ora grazie al pacchetto `enumitem` si definiscono, o ridefiniscono, comunque si personalizzano i vari ambienti che formano liste di tipo diverso. Intanto in tutte le liste descrittive l’etichetta viene sempre composta in maiuscolo; se si vuole cambiare font o stile basta usare gli appositi comandi nella forma *<chiave>* = *<valore>* forniti dal pacchetto `enumitem`. Solo per il nuovo ambiente descrittivo

`plaindescription` la didascalia di default è composta in `\normalfont`. L'ambiente `blockdescription` viene definito senza le etichette sporgenti e senza rientranza del margine sinistro. I nuovi ambienti `compactenumerate`, `compactitemize` e `compactdescription` sono definiti in modo da annullare tutti gli spazi verticali interni e in modo da ridurre gli spazi prima e dopo la lista.

```

196 \renewcommand\descriptionlabel[1]{\hspace\labelsep\normalfont\scshape #1}
197 \renewcommand\blockdescriptionlabel[1]{\normalfont\scshape #1}
198 \providecommand\plaindescriptionlabel[1]{\hspace\labelsep\normalfont #1}
199
200 \renewlist{blockdescription}{description}{1}
201 \setlist[blockdescription]{before={\let\makelabel\blockdescriptionlabel},
202 leftmargin=\z@,labelsep*=0.5em,labelindent=\z@,labelwidth=\z@}
203
204 \newlist{plaindescription}{description}{1}
205 \setlist[plaindescription]{before={\let\makelabel\plaindescriptionlabel}}
206
207 \newlist{compactenumerate}{enumerate}{1}
208 \setlist[compactenumerate,1]{label=\arabic*.,
209 noitemsep, partopsep=\z@, topsep=.25\onelineskip}
210
211 \newlist{compactitemize}{itemize}{4}
212 \setlist[compactitemize]{label=*,
213 noitemsep, partopsep=\z@, topsep=.25\onelineskip}
214
215 \newlist{compactdescription}{description}{1}
216 \setlist[compactdescription]{%
217 style=sameline, noitemsep, partopsep=\z@, topsep=.25\onelineskip}

```

Vengono poi impostati e differiti all'inizio del documento i contatori che regolano la profondità di numerazione delle sezioni e della loro inclusione nell'indice generale.

```

218 \AtBeginDocument{\setsecnumdepth{subsection}
219 \settocdepth{subsection}\maxsecnumdepth{subsection}
220 \maxtocdepth{subsection}}

```

Viene ora definita la virgola intelligente; si veda una descrizione più dettagliata della sua utilità in matematica. Qui si commenta solo il codice.

La prima parte del codice serve per rendere attiva la virgola solo in modo matematico e solo quando la lingua principale *non* è l'inglese. Si fornisce anche un comando `\m@thcomma` come cuore della definizione della virgola attiva.

```

221 \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
222 \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}
223 \ifxetex
224 \AtBeginDocument{\ifcsstring{xpg@main@language}{english}{\relax}{%
225 \mathcode'\,=\string"8000}}
226 \else
227 \AtBeginDocument{\ifcsstring{language}{english}{\relax}{%
228 \mathcode'\,=\string"8000}}
229 \fi
230 {\catcode '\,=\active \gdef,{\futurelet\let@token\m@thcomma}}%

```

Poi, ritardando l'esecuzione alla fine del preambolo, si definisce `\m@thcomma` in modo che svolga il suo compito: `\let@token` contiene il token che segue `\m@thcomma` nel file sorgente; precisamente il token che segue la virgola nel file sorgente. Perciò

esso può essere uno spazio, un carattere implicito, un carattere esplicito, una macro, un qualunque token di categoria diversa da 11 e da 12. Il comando `\m@thcomma` assorbe anche il primo token che la segue, purché non sia uno spazio, eventualmente già memorizzato in `\let@token`. Lo scopo della virgola intelligente è quello di sapere se il carattere che viene immediatamente dopo sia una cifra, che è di categoria 12, come l'asterisco. Quindi per prima cosa si controlla se `\let@token` ha categoria 12. Se non lo è allora non si tratta sicuramente di una cifra e la virgola intelligente inserisce la virgola interpuntiva. Ma se lo è potrebbe essere un carattere esplicito alfabeticamente oppure un carattere implicito di categoria 12. In questo caso non lo si può usare per verificare se il carattere implicito sia una cifra; ma in matematica nessuno si sognerebbe mai di indicare con un carattere implicito una qualunque delle 10 cifre decimali; dunque se si tratta di un carattere implicito esso rappresenta qualche carattere diverso da una cifra e ci vuole la virgola interpuntiva. Solo che per sapere se si tratta di un carattere implicito bisogna verificare se esso ha la forma di una sequenza di controllo. Per fare questo bisogna prendere la stringa che costituisce l'argomento di `\m@thcomma`, togliergli il primo carattere (eventualmente un backslash) e vedere se quel che resta costituisce il nome di una sequenza di controllo a cui è stato assegnato un significato; dunque se il token fosse per esempio `\infty` la stringa sarebbe formata dai caratteri `\`, `i`, `n`, `f`, `t`, `e` `y`, togliendo il primo dei quali resta la “parola” `infty`; il test `\ifcsname infty\endcsname` restituisce il valore “vero” se `\infty` ha un significato; nel nostro caso quindi si tratterebbe di un carattere implicito e ci vuole la virgola interpuntiva. Se invece il primo argomento della macro `\m@thcomma` fosse un carattere esplicito di categoria 12 (non potrebbe essere altro a questo punto dei test), questa operazione di togliere il primo carattere, lascerebbe una stringa vuota cosicché il test `\ifcsname\endcsname` darebbe luogo ad un valore “falso”, quindi abbiamo la certezza che si tratti di un carattere esplicito di categoria 12; bisogna ora testare se si tratta di una cifra. I due test `\ifnum'#1<'0` e `\ifnum'#1>'9` controllano se il carattere in questione è estero o interno all'intervallo ASCII delle cifre; se esso è esterno ci vuole la virgola interpuntiva, altrimenti la virgola decimale.

Non tanto semplice ma efficace sia con pdfLaTeX sia con XeLaTeX.

```

231 \AtEndPreamble{%
232 \gdef\m@thcomma#1{%
233 \unless\ifcat\noexpand\let@token%
234   \virgola
235 \else
236 \expandafter\expandafter\expandafter
237   \ifcsname\expandafter\@gobble\string#1\endcsname
238   \virgola
239 \else
240   \ifnum'#1<'0\relax
241     \virgola
242   \else
243     \ifnum'#1>'9\relax
244       \virgola
245     \else
246       \virgoladecimale
247     \fi
248   \fi
249 \fi

```

```

250 \fi#1}%
251 }

```

Si ridefiniscono l'ambiente `thebibliography` e `theindex` per poterne inserire il titolo nell'indice e per poter avere il “target” nel punto giusto quando si clicca su un link ipertestuale che porti all'inizio di queste “sezioni”. Per giunta queste parti dovrebbero cadere nella back matter, quindi non verrebbero nemmeno numerati.

Per la bibliografia si sfruttano i comandi interni di `memoir` per ottenere quello che si desidera e nello stesso tempo per consentire di mettere un prologo prima dell'elenco bibliografico. La documentazione di `memoir` dice che bisogna ridefinire il comando interno `\bibsection`.

```

252 \renewcommand{\bibsection}{\chapter{\bibname}\prebibhook}

```

Per l'indice o gli indici analitici non dovrebbero esserci problemi, nel senso che il pacchetto `imakeidx` li ridefinisce a sua volta per poterli gestire a modo suo. Probabilmente è del tutto inutile la ridefinizione fatta in questa classe, ma non dà nessun fastidio, quindi la si lascia lo stesso.

```

253 \let\imki@idxprologue\empty
254 \def\imki@columns{2}
255 \renewenvironment{theindex}
256 {%
257 \clearforchapter
258 \csname phantomsection\endcsname
259 \chapter{\indexname}%
260 \indexmark%
261 \parindent\z@
262 \parskip\z@ \@plus .3\p@\relax
263 \let\item\@idxitem
264 \begin{multicols}{\imki@columns}[\imki@idxprologue]
265 }
266 {%
267 \end{multicols}\gdef\imki@idxprologue{}\clearpage
268 }

```

All'inizio del documento si inserisce l'indicazione per lo stile bibliografico. Avendo caricato il pacchetto `natbib` e lo stile bibliografico `guidatematica.bst` le citazioni e l'elenco dei riferimenti bibliografici vengono eseguiti con lo stile “autore-anno”. `natbib` da parte sua mette a disposizione tanti comandi della famiglia `\cite`, che si possono usare più di una mezza dozzina di tali comandi per avere lo stile predefinito, fra parentesi, solo l'autore, solo l'anno, eccetera.

```

269 \AtBeginDocument{\bibliographystyle{guidatematica}}%

```

Come configurazione generale con `XeLaTeX` si impone lo stile senza il rientro del primo capoverso di un capitolo o di ogni sezione. Con `pdfLaTeX` non è necessario, perché questa funzionalità è predefinita. Tuttavia ogni autore di guide tematiche è padronissimo di impostarsi il pacchetto `indentfirst` se vuole rientrare tutti i capoversi, anche quelli che seguono ogni comando di sezionamento.

```

270 \ifxetex
271 \ifcsstring{xpg@main@language}{italian}%
272 {\csgappto{init@extras@italian}{\nofrench@indent}}{ }%
273 \fi}

```

L'ambiente per la pagina del titolo, `titlepage` non è definita con la classe `memoir`; qui la si definisce apposta e funzionalmente per questa classe, non è un

comando da usare da parte dell'utente; egli deve invece limitarsi a dare i comandi `\author` un solo autore, o una lista di autori separati da virgole o altri spaziatori; `\title` con il titolo della guida; `\subtitle` per un eventuale sottotitolo; `\date` con le informazioni che ritiene più opportune.

```

274 \newenvironment{titlepage}{\clearpage
275 \pagestyle{titlepage}
276 \oddsidemargin=\dimexpr(\oddsidemargin+\evensidemargin)/2\relax
277 \makeoddhead{titlepage}{}{\smash{\Large\scshape\@author}}{}
278 \makeoddfoot{titlepage}{}{\smash{\large\scshape\@date}}{}
279 }{\clearpage}
280
281 \providecommand\subtitle[1]{\gdef\@subtitle{#1}}
282 \let\@subtitle\empty
283
284 \renewcommand\maketitle{
285 \begin{titlepage}
286 \vspace*{\stretch{.7}}
287 {\centering\Large\scshape\@title\par}
288 \ifvoid\@subtitle{}{\vspace{2\onelineskip}}
289 {\centering\scshape\Large\@subtitle\par}}
290 \vspace{\stretch{.7}}
291 {\centering\includegraphics[width=40mm]{LogoGuITtondo}\par}
292 \vspace*{\stretch{.7}}
293 \end{titlepage}
294 }

```

Invece sono importanti questi due ambienti: `medaglione` e `sintassi`. Il primo serve per incorniciare un testo; per la larghezza l'ambiente `medaglione` accetta un argomento facoltativo, predefinito alla giustezza corrente (quindi una giustezza che cambia all'interno delle liste); determina la giustezza di composizione all'interno del medaglione tenendo conto dello spessore del filetto della cornice e dello spazio di separazione fra la cornice e il testo in essa contenuto.

Il secondo ambiente, `sintassi` è un medaglione adatto per incorniciare i comandi e quindi per descrivere la loro sintassi. Spesso l'ambiente `sintassi` contiene la descrizione della sintassi di un solo comando; dovendo indicare la sintassi di molti comandi è opportuno sfruttare la composizione sbandierata con l'allineamento a sinistra (predefinito) ma con un "a capo" automatico ogni volta che nel sorgente si incontra un "a capo" nel testo sorgente; in questi casi sta all'utente di specificare il comando `\obeylines` all'inizio dell'ambiente cosicché non dovrà più ricordarsi di mettere esplicitamente un comando `\\` alla fine di ogni riga.

```

295 \newenvironment{medaglione}[1][\linewidth]{%
296   \begin{lrbox}{0}%
297   \begin{minipage}{\dimexpr#1-2\fbboxsep-2\fbboxrule}
298 }{%
299   \end{minipage}\end{lrbox}\fbbox{\usebox{0}}\relax
300 }
301
302 \newenvironment{sintassi}{\flushleft\medaglione}%
303   {\endmedaglione\endflushleft}

```

Per descrivere la sintassi sono utili i comandi seguenti al fine di semplificare la scrittura delle parti di testo corrente e differenziare gli argomenti generici; quelli obbligatori e quelli facoltativi; dovendo descrivere delle cose riguardanti il, disegno

programmato, le coordinate spesso vengono distinte con due valori, separati da una virgola e racchiusi fra parentesi tonde; anche in questo caso siamo di fronte alla descrizione di una sintassi e si dispone di un apposito comando per descrivere le coordinate geometriche.

I comandi sono `\meta` per un *<argomento generico>*; `\marg` per un *<argomento obbligatorio>*; `\oarg` per un *<argomento facoltativo>*; `\garg` per le *<coordinate>*, *<geometriche>*; `\comando` per un *\comando* completa del suo backslash; `\cs` per il nome (senza backslash) di un *\comando* al quale viene anteposto il suo necessario backslash quando esso viene composto; `\Sambiente` per descrivere la sintassi di uno specifico ambiente che accetta un argomento come in `\begin{<ambiente>}[<argomento>] ... \end{<ambiente>}`; `\Dambiente` per descrivere la sintassi di un ambiente che accetta un argomento facoltativo per il quale bisogna indicare il valore predefinito, come in `\begin{<ambiente>}[<argomento>][<default>] ... \end{<ambiente>}`; `Bambiente` per indicare l'apertura di un ambiente specifico; `Eambiente` per indicare la chiusura di un ambiente specifico; `\Arg` per indicare fra graffe un argomento specifico {123}.

```

304 \newcommand*\hz{\nobreak\hskip\z@}
305 \renewcommand*\meta[1]{\langle$\textit{\hz#1}$\rangle$}
306 \renewcommand*\marg[1]{\texttt{\char123}\meta{#1}\texttt{\char125}}
307 \renewcommand*\oarg[1]{\texttt{[]}\meta{#1}\texttt{[]}}
308 \newcommand*\Arg[1]{\texttt{\char123}\relax#1\char125}}
309 \def\GT@splitargs#1,#2!\def\@tempA{#1}\def\@tempB{#2}}
310 \newcommand\garg[1]{\GT@splitargs#1!\texttt{()}\meta{\@tempA}%
311   \texttt{,}\meta{\@tempB}\texttt{)}}
312 \newcommand*\comando[1]{\texttt{\string#1}}
313 \renewcommand*\cs[1]{\texttt{\char92#1}\index{#1@\texttt{\char92#1}}}
314 \let\csindex\cs
315 \newcommand*\Sambiente[2]{\cs{begin}\marg{#1}\oarg{#2}\,,\dots
316   \cs{end}\marg{#1}}
317 \newcommand*\Dambiente[3]{\cs{begin}\marg{#1}\oarg{#2}\oarg{#3}\,,\dots
318   \cs{end}\marg{#1}}
319 \newcommand*\Bambiente[1]{\cs{begin}\texttt{\char123#1\char125}}
320 \newcommand*\Eambiente[1]{\cs{end}\texttt{\char123#1\char125}}

```

Invece i comandi che seguono servono per comporre con un certo stile grafico certi elementi del linguaggio dagli ambienti alle classi fino ai codici di allineamento di certi oggetti con gli oggetti circostanti; ci si può riferire alla documentazione d'uso della classe *guidatematica-doc.pdf* distribuita con il kit di composizione delle guide tematiche.

Si noti che molti di questi oggetti possono venire inseriti in uno o più indici analitici; sta al compositore stabilire se la Guida tematica abbia necessità di uno o più indici analitici; la documentazione *guidatematica-doc.pdf* contiene maggiori dettagli. Qui vale la pena di sottolineare che i comandi il cui nome termina con `style` servono solo per gestire lo stile di composizione; quelli invece che cominciano con la stessa “parola” ma non terminano con `style` compongono nel testo la locuzione specifica con lo stile che le si addice ma, se è stato specificato il comando `\makeindex`, eventualmente con le sue opzioni, inviano anche le apposite informazioni al file di servizio `\jobname.idx` per essere poi elaborati con il programma `makeidx` per avere le varie voci in ordine alfabetico gerarchico, al fine di poter comporre l'indice analitico come prescritto.

```

321 \DeclareRobustCommand*\ambstyle[1]{\textsf{\slshape#1}}
322 \DeclareRobustCommand*\classstyle[1]{\texttt{\itshape#1}}
323 \DeclareRobustCommand*\filestyle[1]{\texttm{\textl{#1}}}
324 \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape\textl{#1}}}
325 \DeclareRobustCommand*\progstyle[1]{\textsf{#1}}
326 \DeclareRobustCommand*\prog[1]{\progstyle{#1}%
327   \index{programma!#1@\progstyle{#1}}}
328 \DeclareRobustCommand*\pack[1]{\packstyle{#1}%
329   \index{pacchetto!#1@\packstyle{#1}}}
330 \DeclareRobustCommand*\class[1]{\classstyle{#1}%
331   \index{classe!#1@\classstyle{#1}}}
332 \DeclareRobustCommand*\file[1]{\filestyle{#1}%
333   \index{file!#1@\filestyle{#1}}}
334 \DeclareRobustCommand*\amb[1]{\ambstyle{#1}%
335   \index{ambiente!#1@\ambstyle{#1}}}
336 %
337 \DeclareRobustCommand*\opzstyle[1]{\textsl{\textl{#1}}}
338 \DeclareRobustCommand*\contastyle[1]{\texttm{#1}}
339 \DeclareRobustCommand*\stilestyle[1]{\texttm{#1}}
340 \DeclareRobustCommand*\numeristyle[1]{\texttm{#1}}
341 \DeclareRobustCommand*\umisurastyle[1]{\texttm{#1}}
342 \DeclareRobustCommand*\chiavestyle[1]{\texttm{#1}}
343 \DeclareRobustCommand*\descrittorestyle[1]{\texttm{#1}}
344 \DeclareRobustCommand*\posizionestyle[1]{\texttm{#1}}
345 \DeclareRobustCommand*\allineamentostyle[1]{\texttm{#1}}
346
347 \DeclareRobustCommand*\opz[1]{\opzstyle{#1}%
348   \index{opzione!#1@\opzstyle{#1}}}
349 \DeclareRobustCommand*\conta[1]{\contastyle{#1}%
350   \index{contatore!#1@\contastyle{#1}}}
351 \DeclareRobustCommand*\stile[1]{\stilestyle{#1}%
352   \index{stile della pagina!#1@\stilestyle{#1}}}
353 \DeclareRobustCommand*\numeri[1]{\numeristyle{#1}%
354   \index{numerazione!#1@\numeristyle{#1}}}
355 \DeclareRobustCommand*\umisura[1]{\umisurastyle{#1}%
356   \index{unit\`a di misura!#1@\umisurastyle{#1}}}
357 \DeclareRobustCommand*\chiave[1]{\chiavestyle{#1}%
358   \index{chiave!#1@\chiavestyle{#1}}}
359 \DeclareRobustCommand*\descrittore[1]{\descrittorestyle{#1}%
360   \index{descrittore di colonna!#1@\descrittorestyle{#1}}}
361 \DeclareRobustCommand*\posizione[1]{\posizionestyle{#1}%
362   \index{posizione degli oggetti flottanti!#1@\posizionestyle{#1}}}
363 \DeclareRobustCommand*\allineamento[1]{\allineamentostyle{#1}%
364   \index{codice di allineamento!#1@\allineamentostyle{#1}}}

```

Finalmente la macro \GuIT per gestire l'immagine del logo del GuIT.

```

365 \DeclareRobustCommand*\GuIT{\dimen@=\f@size\p\dimen2=20.88\p@
366 \edef\guitscale{\strip@pt\dimexpr\dimen@*\p@/\dimen2\relax}%
367 \raisebox{-0.93ex}{\z@}{\z@}{%
368   \includegraphics[scale=\guitscale]{LogoGuITlineare}}}%
369 \let\guit\GuIT

```

Questo è tutto.