

# La classe `guidatematica.dtx`

GuIT\*

Versione v.2.1.00; ultima revisione 2020-04-29.

## Sommario

Questa classe `guidatematica` 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. Differisce dalla classe `guidatematica` delle versioni precedenti per il fatto che il disegno della pagina è simmetrico e quindi si presta ugualmente bene per comporre (e stampare) *oneside* o *twoside*. Solo le note marginali eventualmente presenti mostrano la differente impostazione. Inoltre è costruita in modo da poter lavorare indifferentemente con `pdfLaTeX`, `XeLaTeX` e `LuaLaTeX`,

## 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`; alcune opzioni per il formato della “carta”, sono state ridefinite, ma l’utente non dovrebbe osservare nessuna variazione rispetto all’impostazione che si avrebbe senza queste ridefinizioni.
- Nelle versioni precedenti si poteva specificare anche l’opzione `ipertesto`, ma è stata eliminata per questo motivo: il suo scopo era quello di caricare il pacchetto `hyperref` per ultimo, perché si riteneva che fosse utile per l’utente non doversi preoccupare di questo pacchetto essendo sua scelta personale quella di usare o non usare i collegamenti ipertestuali. Ma, pur restando una raccomandazione importante quella di caricare `hyperref` il più tardi possibile, esistono dei pacchetti, come, per esempio, `glossaries`, che vanno caricati dopo. Perciò ora al momento di eseguire il comando `\begin{document}` questa classe imposta solo i colori dei link; l’utente è pregato di non modificarli perché le guide tematiche devono avere gli stessi aspetti generali.
- 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 al loro stile uniforme di composizione.

---

\*Web: [www.guitex.org](http://www.guitex.org)

- 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.
- 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, se occorre, 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 fosse 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 e LuaLaTeX 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 specificare 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 quasi tutte passate alla classe `memoir`. Viene definita l'opzione `ipertesto` per caricare il pacchetto `hypertext` che non viene caricato di default, perché va caricato per ultimo solo se viene specificata l'opzione suddetta.

L'autore potrebbe avere bisogno di caricare ulteriori pacchetti nel preambolo, quindi è meglio che `hyperref` venga caricato dopo questi ulteriori pacchetti. Per fare ciò basta che l'autore specifichi l'opzione `ipertesto` alla classe che provvede a caricare il pacchetto `hyperref` come ultima cosa al momento di iniziare la composizione della guida tematica.

Vengono reimpostate le opzioni `a4paper`, `a5paper` e `\b5paper` e vengono definite le opzioni `tablet` e `pad`, per poter in tutti questi casi creare il layout della pagina in modo simile e non personalizzabile. Per tutte queste opzioni bisogna

ricorrere ad opportuni comandi condizionali, per poter ritardare la loro esecuzione ad un momento successivo al caricamento della classe `memoir`.

```

1 %
2 \AtBeginDocument{\@ifpackageloaded{hyperref}{\hypersetup{colorlinks,linkcolor={blue},
3   citecolor={blue!80!black},urlcolor={blue}}}{}}
4
5
6 \newif\ifPAD\PADfalse
7 \DeclareOption{pad}{\PADtrue\tabletfalse\Aivfalse\Bvfalse\Avfalse}
8 \newif\iftablet\tabletfalse
9 \DeclareOption{tablet}{\tablettrue\Aivfalse\Bvfalse\Avfalse\PADfalse}
10 \newif\ifAiv \Aivfalse
11 \DeclareOption{a4paper}{\Aivtrue\tabletfalse\Bvfalse\Avfalse\PADfalse}
12 \newif\ifBv \Bvtrue
13 \DeclareOption{b5paper}{\Aivfalse\Bvtrue\tabletfalse\Avfalse\PADfalse}
14 \newif\ifAv \Avfalse
15 \DeclareOption{a5paper}{\Avtrue\Aivfalse\Bvfalse\tabletfalse\PADfalse}
16
17 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
18 \ProcessOptions*\relax
19
20 \LoadClassWithOptions{memoir}

```

Si carica subito il pacchetto `iftex` che permette di distinguere quale motore di composizione si sta usando fra `pdftex`, `xetex`, oppure `luatex`.

```

21 \RequirePackage{iftex}

```

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.

```

22 \ifPDFTeX
23   \RequirePackage[utf8]{inputenc}
24   \RequirePackage[T1]{fontenc}
25   \RequirePackage[italian]{babel}
26   \RequirePackage[tt={oldstyle=false,tabular,monowidth}]{cfr-lm}
27 \else
28   \RequirePackage{fontspec}
29   \defaultfontfeatures[\rmfamily,\sffamily]{Ligatures=TeX,Numbers=OldStyle}
30   \defaultfontfeatures[\ttfamily,\tvfamily]{Numbers=Lining}
31   \setmainfont{Latin Modern Roman}[SmallCapsFont={* Caps}]
32   \setsansfont{Latin Modern Sans}
33   \setmonofont{Latin Modern Mono}
34   \newfontfamily{\tvfamily}{Latin Modern Mono Prop}
35   \DeclareTextFontCommand{\textttm}{\ttfamily}
36   \DeclareTextFontCommand{\textttv}{\tvfamily}
37   \def\textl#1{\addfontfeature{Numbers=Lining}#1}
38   \RequirePackage{unicode-math}
39   \setmathfont{Latin Modern Math}
40   \RequirePackage{polyglossia}
41   \setmainlanguage[babelshorthands]{italian}
42   \ifx\tmspace\@undefined
43     \newlength{\tmspace}\tmspace=\z@

```

```

44 \fi
45 \fi

```

Si caricano poi i pacchetti ritenuti necessari; in questa versione della classe non viene caricato il pacchetto `guit.sty` perché è incompatibile con `XeLaTeX`; certo lo si sarebbe potuto caricare nel ramo `true` del test `\ifPDFTeX` e la definizione del logo del gruppo in forma grafica la si sarebbe potuta limitare all'uso con `XeLaTeX` o di `LuaLaTeX`; siccome alla data attuale il pacchetto `guit.sty` è sottoposto a revisione per l'uso con i font OpenType, è molto meglio usare temporaneamente le macro in ogni caso e modificare questa classe solo con l'aggiunta di un nuovo pacchetto quando `guit.sty` sarà diventato compatibile con tutti i programmi di composizione. Il logo tondo, il “timbro”, del GuIT viene invece allegato come codice, visto che si tratta di una variante nella quale le lettere della scritta attorno al bordo non appaiono mai capovolte.

```

46 \RequirePackage{graphicx}
47 \RequirePackage{metalogo}
48 \RequirePackage{pict2e}[2009/06/01]
49 \RequirePackage{microtype}
50 \RequirePackage{etoolbox}
51 \RequirePackage{xcolor}
52 \RequirePackage{natbib}
53 \RequirePackage{multicol}
54 \RequirePackage{imakeidx}
55 \RequirePackage{fancyvrb}
56 \RequirePackage{afterpage}
57 \RequirePackage{enumitem}
58 \RequirePackage[right]{eurosym}
59 \RequirePackage{fancyvrb}
60 \RequirePackage{listings}
61 \lstloadlanguages{[LaTeX]TeX}

```

Il disegno della gabbia del testo è ispirato al pacchetto `canoniclayout`. Però è stato osservato che se si compone in modo `oneside` la lettura a schermo è agevole, ma la stampa della guida risulta mal composta in fronte retro. Se invece si compone in modo `twoside` da una pagina all'altra il testo si sposterebbe alternativamente a destra e a sinistra a seconda della parità del numero di pagina, visto che il disegno della pagina prodotto da `canoniclayout` è asimmetrico.

Per ovviare a questi inconvenienti contemporaneamente necessita centrare orizzontalmente la gabbia e definire successivamente le testatine e i piedini in forma simmetrica così da rendere il layout delle pagine pari identiche a quelle dispari; la lettura a schermo e la stampa risultano pertanto identiche. Per testatine e piedini si veda più avanti. Per simmetria la gabbia viene centrata anche verticalmente.

Prima però bisogna definire il formato della carta a seconda del formato specificato con l'apposita opzione. Poi, visto che la classe `memoir` è già stata caricata, si possono usare i suoi comandi specifici per definire gabbia, margini e le altre dimensioni. Si sfruttano registri già definiti da `memoir` e si usano alcuni comandi specifici della classe.

```

62 \normalfont
63 \ifPAD\setstocksize{160mm}{120mm}\fi
64 \iftablet\setstocksize{120mm}{90mm}\fi
65 \ifBv\setstocksize{250mm}{176mm}\fi
66 \ifAiv\setstocksize{297mm}{210mm}\fi
67 \ifAv\setstocksize{210mm}{146mm}\fi

```

```

68 \settrims{0pt}{0pt}
69
70 \settypeblocksize{\paperwidth}{\dimexpr\paperwidth*\paperwidth/\paperheight}{*}
71 \setlrmargins{*}{*}{*}
72 \setulmargins{*}{*}{*}
73 \setheadfoot{\baselineskip}{2\onelineskip}
74 \setheaderspaces{*}{\onelineskip}{*}
75 \setmarginnotes{7pt}{\dimexpr\foremargin-3em}{5pt}
76 \checkandfixthelayout[nearest]

```

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.

```

77 \def\GetFileInfo#1{%
78   \def\filename{#1}%
79   \def\@tempb##1 ##2 ##3\relax##4\relax{%
80     \def\filedate{##1}%
81     \def\fileversion{##2}%
82     \def\fileinfo{##3}}%
83   \edef\@tempa{\csname ver@#1\endcsname}%
84   \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 particolarmente per una lettura confortevole a monitor.

```

85 \let\@copymark\@empty % Di default è vuoto
86 \newcommand*\setcopymark[1]{\gdef\@copymark{#1}}
87 \newcommand*\insertcopymark{%
88   \begin{picture}(0,0)\unitlength=1pt\relax
89   \if@twoside
90     \ifodd\value{page}
91       \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
92         {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
93     \else
94       \put(-\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
95         {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
96     \fi
97   \else
98     \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
99     {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
100   \fi
101   \end{picture}%
102 }

```

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 suppone che la maggior parte delle guide tematiche sia composta con l'opzione `oneside`. Questi comandi sono completamente parametrizzati alle dimensioni del foglio rifilato e valgono per qualunque formato.

```

103 \def\cblayoutsinistro{%
104 \dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
105 \dimen258=\dimexpr\evensidemargin+1in\relax
106 \dimen262=1mm\relax
107 \dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
108 \dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
109     -\headsep-\textheight\relax
110 \begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
111 \put(0,0){\unitlength=\p@
112 \put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
113 \color{red}%
114 \put(\strip@pt\dimen258,\strip@pt\dimen264){%
115     \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
116     \Line(0,0)(\strip@pt\paperwidth,\strip@pt\paperheight)%
117     \put(\strip@pt\dimexpr\paperwidth/2\relax,%
118         \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
119         {\circle{\strip@pt\paperwidth}}}%
120 }
121 \end{picture}}
122
123 \def\cblayoutdestro{%
124 \dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
125 \dimen258=\dimexpr\oddsidemargin+1in\relax
126 \dimen262=1mm\relax
127 \dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
128 \dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
129     -\headsep-\textheight\relax
130 \begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
131 \put(0,0){\unitlength=\p@
132 \put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
133 \color{red}%
134 \put(\strip@pt\dimen258,\strip@pt\dimen264){%
135     \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
136     \Line(0,\strip@pt\paperheight)(\strip@pt\paperwidth,0)%
137     \put(\strip@pt\dimexpr\paperwidth/2\relax,%
138         \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
139         {\circle{\strip@pt\paperwidth}}}%
140 }
141 \end{picture}}

```

Si noti che non sempre il rettangolo che rappresenta la gabbia è rasente agli ascendenti della prima riga della pagina e alla linea di base dell'ultima linea; questo dipende dal fatto che il layout della pagina creato da `memoir` prevede un piccolo aggiustamento dell'altezza della gabbia affinché contenga un numero intero di righe; si aggiunga che il programma di composizione talvolta inserisce un salto di pagina quando il "goal" di altezza della gabbia non è ancora raggiunto; questa cosa generalmente dipende dal contenuto della pagina stessa che potrebbe contenere oggetti il cui ingombro sulla pagina non corrisponde ad un numero intero di

righe in corpo normale. Se l'aggiustamento è irrisorio, la diagonale della gabbia del testo coincide con la diagonale della pagina.

Si specifica ora il comando `\contribguit` che realizza una pagina con le informazioni relative all'iscrizione nel gruppo. Il comando viene usato in automatico nel retro del frontespizio.

```

142 \providecommand*\Ars{%
143   \textsf{\lower -.48ex\hbox{\rotatebox{-20}{A}}\kern -.3em{rs}}\hskip0pt%
144   \kern -.05em\TeX\hskip0pt\kern -.17em\lower -.357ex\hbox{nica}}
145
146 \AtBeginDocument{\let\originaltitle\title
147 \renewcommand\title[1]{%
148 \def\@title{\let\\\ \normalfont\normalsize#1}\originaltitle{#1}}
149 \let\originalauthor\author
150 \renewcommand\author[2][\Large]{\def\@author{#2}\originalauthor{#1#2}}
151 \def\@copyright{}
152 \DeclareRobustCommand\Copyright[1]{\edef\@copyright{#1}}
153 \let\originalmaketitle\maketitle
154 \renewcommand\maketitle{\frontmatter*\originalmaketitle
155 \contribguit}}
156
157 \let\@licenza\voidbox
158 \newcommand\licenza[1]{\long\gdef\@licenza{#1}}
159
160 \newcommand{\contribguit}{\newpage
161 \thispagestyle{empty}
162 \noindent{\Large
163 Associati anche tu al \GuIT
164 \hfill
165 \setlength{\unitlength}{1mm}
166 \begin{picture}(50,2)
167   \setlength{\fboxsep}{0pt}
168   \put(1,-3){\colorbox{gray}{\framebox(50,6.5){}}}
169   \put(0,-2){\colorbox{white}{\framebox(50,6.5){%
170     \href{https://www.guitex.org/home/it/diventa-socio/associarsi-a-guit}{%
171       Fai click per associarti}}}}
172 \end{picture}%
173 }
174 \bigskip
175
176 L'associazione per la diffusione di \TeX\ in Italia, riconosciuta
177 ufficialmente in ambito internazionale, si sostiene \emph{unicamente}
178 con le quote sociali.
179
180 Se anche tu trovi che questa guida tematica gratuita ti sia stata utile,
181 il mezzo principale per ringraziare gli autori è diventare socio.
182 \medskip
183
184 Divenendo soci si ricevono gratuitamente:
185 \begin{itemize}
186 \item
187 l'abbonamento alla rivista \Ars;
188 \item
189 il DVD \TeX\ Collection;

```

```

190 \item
191 un eventuale oggetto legato alle attività del \GuIT.
192 \end{itemize}
193
194 L'adesione al \GuIT\ prevede un quota associativa compresa tra \EUR{12,00}
195 e \EUR{70,00} a seconda della tipologia di adesione prescelta e ha
196 validità per l'anno solare in corso.
197
198 \vspace{\stretch{1}}
199 \providecommand\authorspace{ }
200 {\parindent=z@\let\authorspace\ }
201 \ifx\@licenza\voidbox
202 \noindent\@title\
203 Copyright \textcopyright\ \ifcsvoid{\@Copyright}{\the\year}{\@Copyright}, %
204 \@author\[\baselineskip]
205 %
206 Questa documentazione è soggetta alla licenza LPPL (\LaTeX\ Project
207 Public Licence), versione 1.3 o successive; il testo della licenza è
208 sempre contenuto in qualunque distribuzione del sistema \TeX\ e nel
209 sito \url{http://www.latex-project.org/lppl.txt}.\[\baselineskip]
210 Questo documento è curato da \@author.
211 \else
212 \@licenza
213 \fi\par}
214 \newpage
215 }

```

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.

Nei piedini 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.

```

216 \nouppercaseheads
217 %\renewcommand{\footruleheight}{\normalrulethickness}
218 \renewcommand{\footruleskip}{0pt}
219 \makeheadrule{headings}{\textwidth}{0pt}
220 \makeheadrule{myheadings}{\textwidth}{0pt}
221 \makeevenfoot{plain}{\@insertcopymark}{\textsc{\thepage}}{}
222 \makeoddfoot{plain}{\@insertcopymark}{\textsc{\thepage}}{\@insertcopymark}
223 \makeevenfoot{headings}{\@insertcopymark}{\textsc{\thepage}}{}
224 \makeoddfoot{headings}{\@insertcopymark}{\textsc{\thepage}}{\@insertcopymark}
225 \makeevenfoot{myheadings}{\@insertcopymark}{\textsc{\thepage}}{}
226 \makeoddfoot{myheadings}{\@insertcopymark}{\textsc{\thepage}}{\@insertcopymark}

```

I mark delle pagine destre e sinistre sono composti centrati sia nelle pagine destre sia in quelle sinistre.

```

227 \makeevenhead{headings}{\textsc{\small\leftmark}}{}
228 \makeoddhead{headings}{\textsc{\small\rightmark}}{}
229 \makeevenhead{myheadings}{\textsc{\small\leftmark}}{}
230 \makeoddhead{myheadings}{\textsc{\small\rightmark}}{}

```



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

```
231 \makepsmarks{headings}{%
232 \createmark{chapter}{both}{shownumber}{\@chapapp\space}{.\qqquad}
233 \createmark{section}{right}{shownumber}{\mathsection$,}{\qqquad}
234 \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`.

```
235 \makepagestyle{headingslayout}
236 \makeevenhead{headingslayout}{\cblayoutsinistro}{\textsc{\small\leftmark}}{}
237 \makeevenfoot{headingslayout}{\@insertcopymark}{\textsc{\thepage}}{}
238 \makeoddhead{headingslayout}{\cblayoutdestro}{\textsc{\small\rightmark}}{}
239 \makeoddfoot{headingslayout}{}{\textsc{\thepage}}{\@insertcopymark}
```

Viene definito lo stile per comporre la pagina iniziale dei capitoli; si compone il titolo del capitolo in maiuscolo 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.

```
240 \makechapterstyle{guidatematica}{%
241   \renewcommand*\chapnumfont{%
242     \fontshape{it}\fontsize{40}{40}\selectfont}
243   \renewcommand*\printchaptername{}
244   \renewcommand*\chaptername{}
245   \renewcommand*\chaptitfont{%
246     \fontsize{18}{16}\scshape}% sterlineato
247   \renewcommand*\printchapternum{%
248     \noindent\rlap{\makebox[\textwidth][r]{%
249       \rlap{\makebox[\foremargin][l]{%
250         \chapnumfont \thechapter}}}\printchaptertitle}
251   \renewcommand*\afterchapternum{}
252 }
253 \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.

```
254 \renewcommand*\partnamefont{\normalfont\large\scshape}
255 \renewcommand*\partnumfont{\normalfont\large\scshape}
256 \renewcommand*\partttitlefont{\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.

```
257 \setbeforesecskip{-3.5ex plus-1ex minus-0.2ex}
258 \setbeforesubsecskip{-3ex plus-1ex minus-0.2ex}
259 \setbeforesubsubsecskip{-2.5ex plus-1ex minus-0.2ex}
260 \setbeforeparaskip{1\onelineskip plus1ex minus0.2ex}
261 \setbeforesubparaskip{1\onelineskip plus1ex minus0.2ex}
262 \setaftersecskip{1.5ex plus0.2ex}
263 \setaftersubsecskip{1.5ex plus0.2ex}
264 \setaftersubsubsecskip{1.5ex plus0.2ex}
265 \setafterparaskip{-1em}
266 \setaftersubparaskip{-1em}
267 \setsubparindent{\parindent}
```

```

268
269 \setsecheadstyle{\large\scshape\raggedright}
270 \setsubsecheadstyle{\large\scshape\raggedright}
271 \setsubsubsecheadstyle{\large\scshape\raggedright}
272 \setparaheadstyle{\small\scshape}
273 \setsubparaheadstyle{\small\scshape}
274
275 \aliaspagestyle{part}{empty}
276 \aliaspagestyle{chapter}{empty}
277 \copypagestyle{titlepage}{headings}
278 \renewcommand\cftpartpagefont{\scshape}
279 \renewcommand\cftpartfont{\large\scshape}
280 \renewcommand\cftchapterfont{\large\scshape}
281 \renewcommand\cftchapterpagefont{\scshape}
282 \renewcommand*\cftchapterfillnum[1]{%
283     {\cftchapterleader}\nobreak
284     \cftchapterformatpnum{#1}%
285     \cftchapterafterpnum\par\nobreak}
286 \setlength{\cftbeforechapterskip}{1.0em \@plus 2\p@}

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

287 \captiondelim{\quad}
288 \captionnamefont{\small\scshape}
289 \captiontitlefont{\small}
290 \captionstyle[\centering]{ }
291 \hangcaption
292 \captionwidth{\dimexpr\textwidth-2\parindent\relax}\changecaptionwidth

    Qui ora grazie al pacchetto enumitem si definiscono, o ridefiniscono, comun-
    que 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'ambien-
    te blockdescription viene definito senza le etichette sporgenti e senza rientran-
    za 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.

293 \renewcommand\descriptionlabel[1]{\hspace\labelsep\normalfont\scshape #1}
294 \renewcommand\blockdescriptionlabel[1]{\normalfont\scshape #1}
295 \providecommand\plaindescriptionlabel[1]{\hspace\labelsep\normalfont #1}
296
297 \renewlist{blockdescription}{description}{1}
298 \setlist[blockdescription]{before={\let\makelabel\blockdescriptionlabel},
299 leftmargin=\z@,labelsep*=0.5em,labelindent=\z@,labelwidth=\z@}
300
301 \newlist{plaindescription}{description}{1}
302 \setlist[plaindescription]{before={\let\makelabel\plaindescriptionlabel}}
303
304 \newlist{compactenumerate}{enumerate}{1}
305 \setlist[compactenumerate,1]{label=\arabic*.,
306     noitemsep, partopsep=\z@, topsep=.25\onelineskip}
307

```

```

308 \newlist{compactitemize}{itemize}{4}
309 \setlist[compactitemize]{label=•,
310   noitemsep,partopsep=\z@,topsep=.25\onelineskip}
311
312 \newlist{compactdescription}{description}{1}
313 \setlist[compactdescription]{%
314 style=sameline,noitemsep,partopsep=\z@,topsep=.25\onelineskip}

```

Un comando utile epr impostare con i font continuamente scalabili, come i Latin Modern semplici o estesi, un corpo qualsiasi; di default lo scartamento è impostato a 1,2 volte il corpo, ma può essere specificato come primo argomento opzionale.

```

315 \newcommand*\cambiacorpo[2][1.2]{\bgroup\dimen@=#2\p@{\dimen@=#1\dimen@
316 \edef\x{\noexpand\egroup\noexpand\fontsize{#2}{\strip@pt\dimen@}}\x\selectfont}
317 \let\setfontsize\cambiacorpo

```

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.

```

318 \AtBeginDocument{\setsecnumdepth{subsection}}
319   \settocdepth{subsection}\maxsecnumdepth{subsection}
320   \maxtocdepth{subsection}}

```

Viene ora definita la virgola intelligente; si veda una descrizione più dettagliata della sua utilità in matematica per esempio della documentazione di questa classe `gidedematica-doc.pdf`. Qui si commenta solo il codice.

Se il comando `\virgoladecimale` risulta già definito, questo è merito del modulo `babel-italian`, quindi non se ne fa niente perché la virgola decimale con tutto il suo armamentario di macro è già definita. Si fornisce invece una definizione più semplice e meno efficiente per quando si usa `xelatex` o `lualatex`. 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.

```

321 \unless\ifcsname virgoladecimale\endcsname
322 \unless\ifPDFTeX
323 \AtBeginDocument{\ifcsstring{xpg@main@language}{english}{\relax}{%
324   \mathcode'\,=\string"8000}}%
325   \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}%
326   \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}%
327 }
328 \else
329 \AtBeginDocument{\ifcsstring{languagesname}{english}{\relax}{%
330   \mathcode'\,=\string"8000}}
331 \fi
332 {\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 alfabetico 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 esterno 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.

```

333 \AtEndPreamble{%
334 \gdef\m@thcomma#1{%
335 \unless\ifcat\noexpand\let@token*%
336   \virgola
337 \else
338   \expandafter\expandafter\expandafter
339     \ifcsname\expandafter\@gobble\string#1\endcsname
340     \virgola
341   \else
342     \ifnum'#1<'0\relax
343       \virgola
344     \else
345       \ifnum'#1>'9\relax
346         \virgola
347       \else
348         \virgoladecimale
349     \fi
350   \fi
351 \fi
352 \fi#1}%
353 }\fi

```

Per essere completi conviene provvedere ai comandi per impostare e per disimpostare la virgola intelligente, rinviandoli alla fine del preambolo.

```

354 \AtEndPreamble{%
355 \providecommand\IntelligentComma{}

```

```

356 \providecommand\NoIntelligentComma{}
357 \renewcommand\IntelligentComma{\mathcode'\,\=\string"8000}
358 \renewcommand\NoIntelligentComma{\mathcode'\,\=\string"613B}
359 }

```

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 l'indice o gli indici analitici non dovrebbero esserci problemi, nel senso che il pacchetto `imakeidx` ridefinisce a sua volta quelli analitici 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.

```

360 \let\imki@idxprologue\empty
361 \def\imki@columns{2}
362 \renewenvironment{theindex}
363 {%
364   \clearforchapter
365   \csname phantomsection\endcsname
366   \chapter{\indexname}%
367   \indexmark%
368   \parindent\z@
369   \parskip\z@ \@plus .3\p@\relax
370   \let\item\@idxitem
371   \begin{multicols}{\imki@columns}[\imki@idxprologue]
372   \raggedright
373 }
374 {%
375   \end{multicols}\gdef\imki@idxprologue{}\clearpage
376 }

```

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.

```

377 \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.

```

378 \unless\ifPDFTeX
379   \ifcsstring{xpg@main@language}{italian}%
380   {\csgappto{init@extras@italian}{\nofrench@indent}}{ }%
381 \fi

```

L'ambiente per la pagina del titolo, `titlepage` non è definita con la classe `memoir`; qui lo 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.

```

382 \newenvironment{titlepage}{\clearpage
383 \pagestyle{titlepage}
384   \makeoddhead{titlepage}{\smash{\Large\scshape\@author}}{\}
385   \makeoddfoot{titlepage}{\smash{\large\scshape\@date}}{\}
386 }{\clearpage}
387
388 \providecommand\subtitle[1]{\gdef\@subtitle{#1}}
389 \let\@subtitle\empty
390
391 \renewcommand\maketitle{
392   \begin{titlepage}
393     \vspace*{\stretch{.7}}
394     {\centering\Large\scshape\@title\par}
395     \ifdefvoid{\@subtitle}{\vspace{2\onelineskip}
396     {\centering\scshape\Large\@subtitle\par}}
397     \vspace{\stretch{.7}}
398     {\centering\resizebox{0.23\textwidth}{!}{\logoguittondo}\par}
399     \vspace*{\stretch{.7}}
400     \end{titlepage}
401 }

```

Invece sono importanti questi tre ambienti: `esercizio`, `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.

```

402 \newenvironment{medaglione}[1][\linewidth]{%
403   \begin{lrbox}{0}%
404   \begin{minipage}{\dimexpr#1-2\fbboxsep-2\fbboxrule}
405 }{%
406   \end{minipage}\end{lrbox}\fbbox{\usebox{0}}\relax
407 }
408
409 \newenvironment{sintassi}{\flushleft\medaglione}%
410 {\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` completo 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` (alias `\Marg` per non confondersi con altre classi) per indicare fra graffe un argomento specifico {123}.

```

411 \newcommand*\hz{\nobreak\hskip\z@}
412 \renewcommand*\meta[1]{\textnormal{$\angle$\textit{\hz#1}$\rangle$}}
413 \renewcommand*\marg[1]{%
414   {\textnormal{\texttt{\char123}\meta{#1}\texttt{\char125}}}
415 \renewcommand*\oarg[1]{\textnormal{\texttt{[]}\meta{#1}\texttt{[]}}}
416 \newcommand*\Arg[1]{\textnormal{\texttt{\{#1\}}}}
417 \let\Marg\Arg
418 \newcommand*\Oarg[1]{\textnormal{\texttt{[#1]}}}
419 \def\GT@splitargs#1,#2!\def\@tempA{#1}\def\@tempB{#2}}
420 \newcommand\garg[1]{\textnormal{\GT@splitargs#1!\texttt{({}\meta{\@tempA}%
421   \texttt{,})\meta{\@tempB}\texttt{)}}}}
422 \newcommand*\comando[1]{\textnormal{\texttt{\string#1}}}
423 \renewcommand*\{cs}[1]{%
424   {\textnormal{\texttt{\char92#1}\index{#1@\texttt{\char92#1}|textsc}}}
425 \let\csindex\cs
426 \newcommand*\Sambiente[2]{\comando{\begin}\marg{#1}\oarg{#2}\,\dots
427   \comando{\end}\marg{#1}}
428 \newcommand*\Dambiente[3]{%
429   \comando{\begin}\marg{#1}\oarg{#2}\oarg{#3}\,\dots
430   \comando{\end}\marg{#1}}
431 \newcommand*\Bambiente[1]{\comando{\begin}\Marg{#1}}
432 \newcommand*\Eambiente[1]{\comando{\end}\Marg{#1}}

```

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. I comandi che seguono permettono di inserire le voci in un unico indice analitico; se ne occorre più di uno essi possono o debbono venire ridefiniti in modo da inviare la voce all'indice desiderato.

```

433 %
434 \DeclareRobustCommand*\ambstyle[1]{\normalfont\textsf{\slshape#1}}
435 \DeclareRobustCommand*\classstyle[1]{\normalfont\texttt{\itshape#1}}
436 \DeclareRobustCommand*\filestyle[1]{\normalfont\texttt{\textl{#1}}}
437 \DeclareRobustCommand*\packstyle[1]{\normalfont
438   \texttt{\ifbool{PDFTeX}{\textl}{\itshape}{#1}}}
439 \DeclareRobustCommand*\progstyle[1]{\normalfont\textsf{#1}}
440 \DeclareRobustCommand*\prog[1]{\progstyle{#1}%
441   \index{programma!#1@\progstyle{#1}|textsc}}
442 \DeclareRobustCommand*\pack[1]{\packstyle{#1}%
443   \index{pacchetto!#1@\packstyle{#1}|textsc}}
444 \DeclareRobustCommand*\class[1]{\classstyle{#1}%
445   \index{classe!#1@\classstyle{#1}|textsc}}
446 \DeclareRobustCommand*\file[1]{\filestyle{#1}%
447   \index{file!#1@\filestyle{#1}|textsc}}
448 \DeclareRobustCommand*\amb[1]{\ambstyle{#1}%
449   \index{ambiente!#1@\ambstyle{#1}|textsc}}
450 %
451 \DeclareRobustCommand*\opzstyle[1]{\normalfont\textsl{\textl{#1}}}
452 \DeclareRobustCommand*\contastyle[1]{\normalfont\texttt{#1}}
453 \DeclareRobustCommand*\stilestyle[1]{\normalfont\texttt{#1}}
454 \DeclareRobustCommand*\numeristyle[1]{\normalfont\texttt{#1}}
455 \DeclareRobustCommand*\umisurastyle[1]{\normalfont\texttt{#1}}
456 \DeclareRobustCommand*\chiavestyle[1]{\normalfont\texttt{#1}}
457 \DeclareRobustCommand*\descrittorestyle[1]{\normalfont\texttt{#1}}
458 \DeclareRobustCommand*\posizionestyle[1]{\normalfont\texttt{#1}}
459 \DeclareRobustCommand*\allineamentostyle[1]{\normalfont\texttt{#1}}
460 %
461 \DeclareRobustCommand*\opz[1]{\opzstyle{#1}%
462   \index{opzione!#1@\opzstyle{#1}|textsc}}
463 \DeclareRobustCommand*\conta[1]{\contastyle{#1}%
464   \index{contatore!#1@\contastyle{#1}|textsc}}
465 \DeclareRobustCommand*\stile[1]{\stilestyle{#1}%
466   \index{stile della pagina!#1@\stilestyle{#1}}}
467 \DeclareRobustCommand*\numeri[1]{\numeristyle{#1}%
468   \index{numerazione!#1@\numeristyle{#1}|textsc}}
469 \DeclareRobustCommand*\umisura[1]{\umisurastyle{#1}%
470   \index{unit\'a di misura!#1@\umisurastyle{#1}|textsc}}
471 \DeclareRobustCommand*\chiave[1]{\chiavestyle{#1}%
472   \index{chiave!#1@\chiavestyle{#1}|textsc}}
473 \DeclareRobustCommand*\descrittore[1]{\descrittorestyle{#1}%
474   \index{descrittore di colonna!#1@\descrittorestyle{#1}|textsc}}
475 \DeclareRobustCommand*\posizione[1]{\posizionestyle{#1}%
476   \index{posizione degli oggetti flottanti!#1@\posizionestyle{#1}|textsc}}
477 \DeclareRobustCommand*\allineamento[1]{\allineamentostyle{#1}%
478   \index{codice di allineamento!#1@\allineamentostyle{#1}|textsc}}

```

Finalmente la macro \GuIT per gestire l'immagine del logo del GuIT, sia quello lineare, sia quello tondo.

```

479 %

```



```

480 \definecolor{verdeguit}{rgb}{0, 0.40, 0}
481 \def\GuIT{\mbox{\color{verdeguit}%
482 \ifPDFTeX\usefont{T1}{lmr}{m}{sc}%
483   g\raisebox{-0.715ex}{\kern-0.26em u}\kern-0.13em
484   \textcolor{black}{I}\kern-0.14em t}%
485 \else\usefont{TU}{lmr}{m}{sc}%
486   g\raisebox{-0.60ex}{\kern-0.285em u}\kern-0.14em
487   \textcolor{black}{I}\kern-0.14em t}%
488 \fi}}
489
490 \usepackage{tikz}
491 \usetikzlibrary{decorations.text}
492
493 \def\logoguittondo{\begin{tikzpicture}[x=1.65em,y=1.65em]\small
494 \draw (0,-0.15) node [circle] {\Huge\GuIT};
495 \path[decorate,decoration={text along path, text={Gruppo Utilizzatori}}]
496   (-1.5,0) .. controls (-1.5,2) and (1.5,2) .. (1.5,0);
497 \path[decorate,decoration={text along path, text={\kern.25em}{\star$}
498 {\star$} Italiani di {\TeX} {\star$} {\star$}}]
499   (-1.8,0) .. controls (-1.8,-2.4) and (1.8,-2.4) .. (1.8,0);
500 \end{tikzpicture}
501 }

```

Ultimo ma non meno importante, la definizione dell'ambiente `pdfxmetadata`. Serve per immettere nel testo i metadati, nel caso che si voglia creare un file archiviabile secondo le norme ISO. L'utente di una guida tematica che volesse creare un guida conforme a queste norme ISO deve caricare esplicitamente caricare il pacchetto `pdfx` con l'opzione `[a-1b]` Prima di caricare `pdfx` l'utente deve inserire nel preambolo questo ambiente `pdfx`, all'interno del quale sono inseriti i metadati necessari. Qui non si scende nei dettagli, ma l'utente deve riferirsi alla guida tematica *FileArchiviabili.pdf*, perché il discorso relativo ai file archiviabili è lungo e complesso. Tuttavia pare corretto mettere già a disposizione in questa classe l'ambiente `pdfxmetadata` così che l'utente non debba "inventarselo".

```

502 \newenvironment{pdfxmetadata}{%
503 \VerbatimOut{\jobname.xmpdata}}{\endVerbatimOut}

```

Questo è tutto.