Claudio Beccari

Introduzione alla creazione di file di classe

IntroCreaClassi.tex v.1.9 del 2015/02/301

Associati anche tu al GUIT

Fai click per associarti

L'associazione per la diffusione di TEX in Italia, riconosciuta ufficialmente in ambito internazionale, si sostiene *unicamente* con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile, il mezzo principale per ringraziare gli autori è diventare socio.

Divenendo soci si ricevono gratuitamente:

- l'abbonamento alla rivista ArsTEXnica;
- il DVD TEX Collection;
- un eventuale oggetto legato alle attività del GUIT.

L'adesione al GUIT prevede un quota associativa compresa tra 12,00 € e 70,00 € a seconda della tipologia di adesione prescelta ed ha validità per l'anno solare in corso.

Introduzione

La creazione di nuove classi è un problema che si presenta più spesso di quanto non si creda, ma che allo stesso tempo scoraggia la maggior parte degli utenti di LaTeX 2_ε i quali, se hanno mai aperto un file di classe, vi hanno trovato dei codici misteriosi, indecifrabili, scoraggianti... e si sono scoraggiati.

In effetti, alcune classi contengono codici scritti in linguaggi misteriosi e non sempre se ne riesce a capire il significato; ora, poi, che sta entrando nell'uso anche il riferimento ai nuovi potentissimi comandi predisposti per il prossimo venturo LaTeX3, ma forniti senza o con poca documentazione, anche i vecchi guru del mestiere cominciano ad avere dei problemi.

Io scrivo classi da almeno venti anni e mi muovo con una certa agilità in questo campo, ma con i nuovi comandi di LaTeX3 mi sento veramente impacciato, per non dire completamente incapace. Tuttavia, se non altro per retrocompatibilità, i vecchi comandi sono sempre lì disponibili e a quelli mi riferirò in questa introduzione.

Quale formazione pregressa bisogna avere acquisito per scrivere un file di classe? Poca o molta, a seconda di quale classe si voglia creare. Per "poca" intendo una sufficiente conoscenza del linguaggio di mark-up di LaTeX 2_ε, non proprio a livello base, ma niente di più di quanto, per esempio, non sia scritto nella guida introduttiva di Lorenzo Pantieri e Tommaso Gordini, *L'arte di scrivere con LaTeX*, (Pantieri e Gordini, 2012). Per "molta" intendo una ottima conoscenza del linguaggio TeX primitivo e dei comandi interni di servizio del nucleo di LaTeX.

È chiaro che con una formazione *media* si può arrivare a scrivere file di classe semplici; con *poca* formazione si possono scrivere file di classe quasi banali, ma pur sempre utili; con *molta* formazione si possono scrivere file di classe partendo da zero, senza appoggiarsi a niente già scritto da altri.

Ma in tutti e tre i casi bisogna sapersi documentare. Ci sono tre modi per

documentarsi e nel seguito mi riferirò esplicitamente alla documentazione già presente nel proprio calcolatore se si dispone di una versione aggiornata della distribuzione TEX Live completa, disponibile per tutti e tre i sistemi operativi maggiori, Windows (da XP in poi), Mac OS X, Linux (per il quale però potrebbero esserci dei problemi di aggiornamento se si tratta di una versione Linux sottoposta alle condizioni del Consorzio Debian[1]). Probabilmente la documentazione distribuita con MiKTEX è altrettanto aggiornata e altrettanto completa, ma generalmente si trova in alberi di cartelle sul disco fisso che hanno nomi diversi da quelli di TEX Live. Nel seguito, con `TEXMF` indico la radice della distribuzione; in un Mac sta per `/usr/local/texlive/`⟨*anno*⟩`/`, dove ⟨*anno*⟩ indica l'anno in cui la versione specifica di TEX Live è stata resa disponibile; sulle altre macchine il percorso è simile, ma specifico di ognuna.

Documentazione di TEX Live L'albero dove cercare la documentazione è `TEXMF/texmf-dist/doc/latex/`. Ogni pacchetto è elencato nella vastissima lista di cartelle; quasi sempre la documentazione è in formato PDF ed è piuttosto curata; per pacchetti che si riferiscono a lingue particolari il file di documentazione potrebbe essere scritto nella lingua a cui si riferisce il pacchetto, ma normalmente la documentazione è in inglese. Un inglese sempre abbastanza facile per cui è comprensibile anche per coloro che hanno una conoscenza molto basilare di questa lingua; al più potrebbero essere di difficile comprensione, all'inizio, le parole tecniche relative all'arte della tipografia, ma dopo poco che si prende l'abitudine di leggere questa documentazione le si impara a conoscere senza problemi.

Il lato ancora migliore di questa enorme quantità di documentazione è che, senza andare a cercare nell'albero di cartelle indicato sopra, ogni file di documentazione è leggibile in un modo semplicissimo: si apre il terminale o prompt dei comandi e si dà il comando `texdoc` seguito da uno spazio e dal nome del pacchetto; al premere del tasto di "invio", si apre il visualizzatore dei file PDF e il documento si legge subito sullo schermo; volendo lo si può anche stampare.

Non è vietato andare a cercare in rete, naturalmente, ma spesso

[1] In questo caso suggerisco di riferirsi al breve testo `texlive-ubuntu.pdf` (Gregorio, 2010b), che insegna come far convivere su una macchina Linux/Debian sia la versione TEX Live/Debian, sia una versione aggiornatissima, *non Debian*, di TEX Live.

i forum che sono liberamente accessibili danno risposte a quesiti specifici; non sempre è facile trovare risposte valide e aggiornate. Il forum del GUIT è una eccezione a questa regola: dalla pagina `http://www.guitex.org/home/` ci si può rivolgere alla varie targhette, in particolare a "Documentazione".

Fra la documentazione presente nella distribuzione TEX Live c'è in particolare il documento `clsguide.pdf` (LATEX 3 Project Team, 2006), da leggere con il solito comando `texdoc clsguide`, intitolato *LATEX 2ε for class and package writers*, che dovrebbe essere studiato molto a fondo: non dico "imparato a memoria", ma quasi.

Documentazione cartacea La documentazione cartacea abbonda nei negozi (anche quelli di *e-commerce*), per cui consultarla è particolarmente facile; alcuni libri sono disponibili gratuitamente anche in rete, ma spesso si tratta di libri il cui copyright è scaduto, quindi sono vecchiotti.

Sul sito GUIT basta cliccare la targhetta "Documentazione" per accedere a una pagina da cui scaricare vari testi: il primo da studiare è *L'arte di scrivere con LATEX* (Pantieri e Gordini, 2012), già citato, e che dovrebbe essere conosciuto in ogni suo dettaglio. Si può scaricare l'*Introduzione all'arte della composizione tipografica con LATEX* (soprannominato *Guida GUIT* (Beccari, 2011a)), per lettori con maggiore esperienza. Ci sono varie monografie di Enrico Gregorio che vale la pena studiare attentamente; in particolare, c'è un link alla pagina Web di Enrico Gregorio dalla quale si può scaricare la seconda edizione del suo *Appunti di programmazione in LATEX e TEX* (Gregorio, 2009) che ritengo indispensabile per chi si accinge a scrivere file di classe anche solo mediamente complessi. Ci sono tutti gli articoli di ArsTEXnica, ma quelli più recenti di un anno sono a disposizione solo dei membri regolari del GUIT; tuttavia ci sono molti articoli didattici che vale la pena leggere. Tutti questi testi si possono scaricare gratuitamente dalla rete; alcuni possono venire acquistati già stampati e brossurati. Visto l'uso che se ne dovrebbe fare le versioni stampate potrebbero essere più maneggevoli, anche per poterle annotare adeguatamente secondo i propri gusti e le proprie necessità. Anche scaricando i file dalla rete e poi stampandoli non è che si risparmi granché, anzi forse si spende di più. Però alcuni dei testi suddetti sono dotati, nella loro forma elettronica, di adeguati

collegamenti ipertestuali, grazie ai quali non è difficile consultare quei documenti "navigandovi" con grande agilità.
In libreria ci sono testi prevalentemente in inglese; in italiano esiste forse ancora la mia vecchia Guida *LaTeX. Guida a un sistema di tipografia elettronica*, ma non lo consiglio perché è abbondantemente obsoleto; c'è anche il testo *TeX per l'impaziente* (Abrahams *et al.*, 1991), traduzione in italiano del corrispondente originale in inglese; sempre in italiano c'è il testo *Il TeX* (Gilardi, 1993); ma tutti questi sono libri "vecchi". È pur vero che quelli che riguardano il linguaggio TeX non si possono considerare veramente vecchi, perché quel linguaggio e il relativo interprete sono stati congelati agli inizi degli anni '90 da Donald Knuth in persona, che si limita a correggere i rarissimi errori che gli vengono segnalati in cambio di un assegno da un dollaro firmato di suo pugno. Tra le guide a LaTeX io ho trovato sempre molto utile quella di Helmut Kopka e Patrick Daly, *Guide to LaTeX 2ε* (Kopka e Daly, 2004); la quarta edizione è del 2004 e quindi è piuttosto recente.
Ma i libri più autorevoli fanno parte di una collana pubblicata dalla Addison Wesley, coordinata da Frank Mittelbach, che è anche il "caposquadra" del *LaTeX3 Project Team*, al lavoro per produrre l'attesa versione 3 di LaTeX. Questa collana, fra gli altri, contiene i testi *The LaTeX Companion* (Mittelbach *et al.*, 2004), *The LaTeX Graphics Companion* (Goossens, 1997), *The LaTeX Web Companion* (Goossens, 1999), elencati in ordine decrescente di importanza per chi vuole prepararsi a scrivere file di classe complessi. Merita segnalare che la Addison Wesley ha sempre sostenuto il sistema TeX e fin dall'inizio ha pubblicato le successive edizioni del *TeXbook* e il manuale di LaTeX ufficiale *LaTeX: A Document Preparation System* (Lamport, 1994).
Certo, se uno si compra dalla Addison Wesley tutti questi testi spende una bella cifra; il lettore interessato sappia che come membro regolare del GuIT può avere, con un piccolo sovrapprezzo sulla quota associativa, anche la "membership" al TeX Users Group (TUG) internazionale, grazie al quale può ottenere sconti significativi sui libri di queste collane.
Va da sé che molti dei documenti elencati contengono molte indicazioni ed esempi, per cui l'aiuto fornito da quei testi è piuttosto valido.

Documentazione su file altrui È molto importante imparare a leggere i file di classe scritti da altri; magari anche il file `TEXMF/tex/latex/base/latex.ltx` che contiene il nucleo di LaTeX; nella stessa cartella ci sono i file `.cls`, quelli che definiscono le classi; ci sono i file `.clo`, quelli che definiscono le varie opzioni, specialmente le dimensioni del corpo dei caratteri e quelle per la posizione delle equazioni o dei loro numeri a destra o a sinistra.

Lo ammetto: all'inizio la lettura è tutt'altro che facile, ma è l'unico modo con cui potevo documentarmi quando circa 30 anni fa ho cominciato ad usare LaTeX. Ho conservato quest'abitudine, per cui trovo spesso più utile e diretto andare a cercare nei file di stile o di classe che non consultare le guide. Non seguite il mio esempio di preferire i file altrui alle guide; consultate prima le guide.

Quando apro la documentazione di un pacchetto che non conosco (e sono centinaia), spesso vado a leggere i corrispondenti file `.sty` e magari mi prendo appunti: è molto utile. Programmare in linguaggio TeX o LaTeX non è come programmare con un linguaggio "macina numeri"; il risultato che si vuole ottenere spesso è difficile da definire perché ha delle valenze estetiche. Questo significa che non esiste *un solo* modo per raggiungere risultati accettabili; esistono forse solo modi più o meno efficienti. Talvolta bisogna avere l'intuizione buona e magari salta fuori una macro che non risolve un solo problema, ma è sufficientemente versatile per risolverne diversi.

Dunque in questo campo è difficile dare consigli; si può certo dire "io farei così", ma senza pretendere che si *debba* fare così. Poi, con analisi critiche e proposte costruttive due persone di parere opposto possono anche discutere e convenire che si potrebbe fare in un terzo modo ritenuto da entrambi i contendenti migliore del proprio. Non capita spesso; ma il mondo della programmazione TeX/LaTeX è abbastanza "fuzzy" (nel senso matematico dei *fuzzy sets*) per far prevalere il buon senso, invece della teoria.

Porto un esempio: nessuno potrebbe considerare che spaziare due entità vuoi in modo testuale, vuoi in modo matematico usando una serie di spazi interparola:

```
parola␣precedente␣\␣\␣\␣\␣parola␣seguente
```

sia più efficiente che scrivere qualcosa come:

```
parola precedente\hspace{2.5em}parola seguente
```

eppure ci sono moltissime persone felici di fare come nel primo modo e non soffrono di sensi di colpa per un procedimento così *naïf*.
Ecco, scrivendo macro per costruire un nuova classe, bisogna stare attenti a evitare cose di questo genere, ma sono convinto che metà delle cose che si scrivono, qualcun altro le avrebbe realizzate meglio. Cerchiamo quindi di fare del nostro meglio e non sentiamoci in colpa se qualcun altro ci facesse osservare che sarebbe stato meglio fare in un altro modo. Dice un vecchio proverbio "cambian col tempo i saggi lor consiglio"; se il modo suggerito con spirito costruttivo ci convince, perché non adottarlo?

Ringraziamenti

Qui intendo ringraziare Tommaso Gordini; da letterato qual egli è, ha ripulito il mio povero italiano; inoltre, con l'esperienza che si è fatto cooperando nella scrittura del testo Pantieri e Gordini (2012), mi ha dato preziosi consigli nella presentazione del materiale contenuto in questa guida tematica.

Ringrazio anche Orlando Iovino che si è prodigato con estrema attenzione a segnalarmi refusi ed errori, alcuni dei quali veramente subdoli.

Ringrazio infine Ivan Valbusa che ha passato al setaccio questo testo ed ha segnalato ancora un buon numero di refusi; questi apparentemente sono dotati di vita propria e generazione spontanea, visto che compaiono o ricompaiono anche dopo che il testo è stato rivisto da diverse persone e ogni volta è stato debitamente corretto.

Indice

Introduzione III

Ringraziamenti IX

Indice X

1 Documentare classi e pacchetti 1

2 Creare classi o pacchetti di estensione 4
- 2.1 Creare classi semplici 4
  - 2.1.1 Una classe semplicissima 4
  - 2.1.2 Un'altra classe semplicissima 11
- 2.2 Creazione di una classe mediante personalizzazioni 13
  - 2.2.1 Una classe per libri con lo stile di Bringhurst 13
  - 2.2.2 Una classe per tesi con stile americano 14
  - 2.2.3 Nuova classe usando pacchetti di configurazione . . . 15

3 Produrre una classe partendo da zero 19
- 3.1 La via breve 19
- 3.2 La struttura della classe `book` 23
  - 3.2.1 Identificazione e opzioni 38
  - 3.2.2 Spazi tecnici e parametri per gli oggetti mobili . . . 39
  - 3.2.3 Lo stile delle pagine 39
  - 3.2.4 I titoli e la pagina del titolo 40
  - 3.2.5 Gestione dei comandi di sezionamento 40
  - 3.2.6 Le liste 42
  - 3.2.7 Tabelle e matrici, figure e tavole 43
  - 3.2.8 I vecchi comandi per i font 44

3.2.9 Indici ed elenchi di oggetti mobili 44
3.2.10 La bibliografia e l'indice analitico 45
3.2.11 Definizioni finali e di default 46
3.3 Verifica del funzionamento di una nuova classe 47

4 Classe per comporre lettere personalizzate 49
4.1 Commento alla classe 51
4.2 Il file `lettera.dtx` 57

5 La classe per comporre questa guida tematica 71
5.1 Il file documentato 72
5.2 Il file di classe estratto dal file documentato 96
5.3 Il file di documentazione debitamente composto 107
5.4 Conclusione 107

Bibliografia 117

Documentare classi e pacchetti 1

Per prima cosa bisognerebbe abituarsi ad usare il pacchetto *docstrip* e la classe standard *ltxdoc*; esiste poi un tutorial (Pakin, 2004) che spiega come confezionare in un solo file sia il codice della classe o del pacchetto sia i suoi commenti esplicativi; la documentazione si legge con il solito programma: `texdoc dtxtut`.

Ecco, questo è il punto e sembra strano che io voglia parlarne prima ancora di aver cominciato a descrivere come si crea la più semplice delle classi. Il punto è che ogni codice di programmazione va adeguatamente commentato, perché bastano pochi giorni e non si capisce più che cosa si è fatto; si perde moltissimo tempo a decifrare quello che noi stessi abbiamo scritto.

Quindi è meglio cominciare con il piede giusto. Per far questo bisogna comporre un file con estensione `.dtx`, acronimo di "documented TEX". Questo file deve avere una struttura particolare i cui dettagli sono spiegati nel tutorial indicato sopra. Il concetto generale è che brani di testo esplicativo e brani di codice sono interlacciati in modo da ottenere la documentazione compilando il file con pdfLaTeX. Il tutorial spiega anche come creare un secondo file con estensione `.ins` per estrarre dal file `.dtx` il file di classe o il file del pacchetto "pulito", cioè senza commenti. In certe situazioni questo modo di procedere non produce risultati negativi, anzi, tiene ben separate le due funzioni del file commentato; ma c'è un problema; il file `.ins`, anche se racchiuso in un archivio compresso, non può venire trasmesso per posta elettronica ad altri, perché ci sono alcuni provider di posta elettronica che vietano il transito di qualunque file con l'estensione `.ins`. Il problema si risolve inserendo il contenuto del file `.ins` dentro allo stesso file `.dtx`, marcato in modo opportuno, in modo che una sola esecuzione del programma pdfLaTeX produca sia il codice pulito sia il file di documentazione.

Più avanti si mostrerà un esempio; per ora ci si accontenti di quanto segue per sapere come strutturare il file `.dtx`.

1. Il file comincia con il codice che permette di estrarre il codice ripulito dai commenti; questa prima parte va racchiusa fra due delimitatori scritti in righe a sé stanti: `%<*system>` subito prima del codice e `%</system>` subito dopo la fine del codice.
2. Segue poi una sezione delimitata da due delimitatori dello stesso tipo `%<*driver>` e `%</driver>` contenente un breve brano di codice con la sintassi del linguaggio LaTeX, che invoca la classe *ltxdoc*, contiene un preambolo, e poi fra `\begin{document}` e `\end{document}` esegue un comando che rilegge lo stesso file `.dtx` (il file `.dtx` legge sé stesso).
3. Segue una terza parte che contiene una descrizione della classe o del pacchetto, lo scopo, le specifiche, le funzionalità e le parti che necessitano di descrizioni particolari. Questa parte non contiene codice vero e proprio, e tutte le sue righe cominciano con il carattere di commento %.
4. La quarta parte, infine, è delimitata dai marcatori `%<*class>` oppure `%<*package>` o contenenti nomi più specifici, ed è terminata dai marcatori di chiusura `%</class>` oppure `%</package>` o quelli corrispondenti ai nomi usati per i marcatori di apertura. Questa parte contiene intercalati blocchi di righe con il testo commentato e blocchi di codice racchiusi fra altri marcatori dalla sintassi un po' particolare:

   ```
   %␣␣␣␣\begin{macrocode}
   ...
   %␣␣␣␣\end{macrocode}
   ```

   in cui i quattro spazi evidenziati sono necessari. Tra questi due delimitatori le righe di codice non vanno commentate. I blocchi di codice possono venire spezzati in blocchi più piccoli, specialmente per le macro più lunghe, in modo da poter intercalare i commenti più appropriati per descrivere il codice adeguatamente.

Naturalmente, come il tutorial descrive bene, esistono molti altri comandi che permettono di fare altre cose, compresa quella di poter descrivere in un solo file `.dtx` diverse classi, diversi file di estensione, diversi file di opzioni, diversi file di definizioni, eccetera. Il grande file `.dtx` contiene

quindi la documentazione e il codice di una collezione coordinata di file, quelli che formano un pacchetto propriamente detto.

L'abilità di maneggiare i file `.dtx` cresce con l'uso e la pratica, ovviamente, ma fin dal primo momento, senza strafare con cose complicatissime, si possono ottenere file documentati di ottima qualità.

Pertanto io consiglierei di iniziare sempre con lo scrivere il file `.dtx`, dettagliandone bene la terza parte, dove bisogna essere molto chiari, specialmente con sé stessi, nel descrivere le specifiche di quello che si vuol fare, come lo si fa, quali funzionalità si possono ottenere, eccetera. Il resto viene dopo; prima bisogna avere chiaro quello che si vuol fare.

Creare classi o pacchetti di estensione 2

2.1 Creare classi semplici

2.1.1 Una classe semplicissima

Supponiamo di voler creare una semplice classe per comporre libri in italiano, usando sempre la codifica di entrata *utf8*, usando sempre i font Latin Modern con codifica *T1* insieme alle estensioni fornite dal pacchetto *textcomp*. Possiamo sempre copiare di volta in volta il preambolo di un altro documento simile, ma siamo sicuri di essere molto più tranquilli se usiamo una classe ad hoc.

Allora predisporremo un file `.dtx` con il contenuto seguente:

```
% \iffalse
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdflatex
%<*system>
\begingroup
\input docstrip.tex
\keepsilent
\preamble
  ----------------------------------------
  The libro class for typesetting books in Italian
  Copyright (C) 2011 Mario Rossi
  All rights reserved

  License information appended

\endpreamble
\postamble
Copyright 2011 Mario Rossi
```

```

Distributable under the LaTeX Project Public License,
version 1.3c or higher (your choice). The latest version of
this license is at: http://www.latex-project.org/lppl.txt

This work is "author-maintained"

This work consists of this file libro.dtx,
and the derived files libro.cls and libro.pdf.

\endpostamble
\askforoverwritefalse
%
\generate{\file{libro.cls}{\from{libro.dtx}{class}}}
%
\def\tmpa{plain}
\ifx\tmpa\fmtname\endgroup\expandafter\bye\fi
\endgroup
%</system>
% \fi
%
% \iffalse
%<*driver>
\ProvidesFile{libro.dtx}
%</driver>
%<class>\NeedsTeXFormat{LaTeX2e}[2009/01/01]
%<class>\ProvidesClass{libro}%
%<*class>
   [2011/11/25 v.0.4 Classe per comporre libri in italiano]
%</class>
%<*driver>
\documentclass[a4paper]{ltxdoc}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern,textcomp}
\usepackage[italian]{babel}
\hfuzz 10pt
%
\begin{document}\errorcontextlines=9
\GetFileInfo{libro.dtx}
\title{La classe \texttt{libro}\thanks{Version number \fileversion;
```

```
 ultima revisione \filedate.}}
\author{Mario Rossi\thanks{e-mail: \texttt{mario dot rossi
    at mymail dot net}}}
 \maketitle
 \DocInput{libro.dtx}
\end{document}
%</driver>
% \fi
%
%
% \CheckSum{10}
% \begin{abstract}
% Questa classe serve per comporre libri in italiano
% appoggiandosi alla classe \texttt{book} ma includendo
% i pacchetti necessari.
% \end{abstract}
%
% \section{Generalità}
%   La classe \texttt{libro.cls} serve per comporre libri in italiano
%   con le seguenti specifiche:
%        \begin{itemize}
%        \item La classe di supporto è la classe standard del
%        system \TeX\ \texttt{book.cls}.
%        \item Tutte le opzioni inserite nell'istruzione
%   |\documentclass| vengono trasferite integralmente
%   alla classe \texttt{book}.
%        \item La codifica di entrata è fissata con la specifica
%   \texttt{utf8} e non è possibile con questa classe specificare
%   una codifica differente.
%        \item La codifica dei font di uscita è \texttt{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.
%        \item Il pacchetto \texttt{babel} viene caricato con la sola
%   opzione
%        \texttt{italian}; nei rari casi in cui sia necessario usare
%   anche altre lingue, esse vanno specificate come opzione alla
%   classe, perché non si può chiamare \texttt{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.
```

```
101 %        \end{itemize}
102 % \StopEventually{}
103 %
104 % \section{Codice commentato}
105 % Le due righe contenti il formato e l'identificazione della classe
106 % sono già state inserite prima; ora possiamo passare alla gestione
107 % delle opzioni che vengono tutte passate alla classe \texttt{book}
108 % \iffalse
109 %<*class>
110 % \fi
111 %    \begin{macrocode}
112 \DeclareOption*{\PassOptionToClass{\CurrentOption}{book}}
113 \ProcessOptions*\relax
114 \LoadClass{book}
115 %    \end{macrocode}
116 % Poi carichiamo i pacchetti relativi alle codifiche di ingresso
117 % e di uscita:
118 %    \begin{macrocode}
119 \usepackage[utf8]{inputenc}
120 \usepackage[T1]{fontenc}
121 %    \end{macrocode}
122 % Scegliamo per l'uscita un font eccellente come Latin Modern, ma
123 % lo accompagniamo con il pacchetto texttt{textcomp} così da
124 % completarlo con molti altri segni alfabetici e analfabetici.
125 %    \begin{macrocode}
126 \usepackage{lmodern,textcomp}
127 %    \end{macrocode}
128 % Infine chiamiamo \texttt{babel} per scegliere la lingua del
129 % documento.
130 %    \begin{macrocode}
131 \usepackage[italian]{babel}
132 %    \end{macrocode}
133 % \iffalse
134 %</class>
135 % \fi
136 % Questo è tutto.
137 % \Finale
138 %
139 % \endinput
```

Nel codice appena mostrato si riconosce che la prima parte del file documentato va dalla riga 1 alla riga 38; la seconda parte dalla riga 40 alla

riga 67, la terza parte, puramente descrittiva, dalla riga 70 alla riga 107 e, infine, la quarta parte va dalla riga 108 alla riga 139. I vari comandi `\iffalse` e `\fi` delimitano dei brani di codice commentato che pdfLaTeX deve saltare durante la sua seconda lettura. Servono anche per non far scrivere nella documentazione le parti che i vari marcatori indicano come parti da estrarre. Per esempio:

```
% \iffalse
%<*class>
% \fi
```

serve per evitare che il marcatore `<*class>` venga stampato nella documentazione, anche se continua ad essere efficace per dire alla parte di programma per l'estrazione che le righe della classe cominciano da quel marcatore.

Salvando questo file con codifica *utf8* come specificato nell'istruzione della codifica di entrata della seconda parte, e compilandolo con pdfLaTeX si ottengono contemporaneamente i file `libro.cls` e `libro.pdf`; il primo file appare così:

```
%%
%% This is file ‘libro.cls’,
%% generated with the docstrip utility.
%%
%% The original source files were:
%%
%% libro.dtx  (with options: ‘class’)
%% ----------------------------------------
%%   The libro class for typesetting books in Italian
%%   Copyright (C) 2011 Mario Rossi
%%   All rights reserved
%%
%%   License information appended
%%
\NeedsTeXFormat{LaTeX2e}[2009/01/01]
\ProvidesClass{libro}%
   [2011/11/25 v.0.4 Classe per comporre libri in italiano]
\DeclareOption*{\PassOptionToClass{\CurrentOption}{book}}
\ProcessOptions*\relax
\LoadClass{book}
```

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern,textcomp}
\usepackage[italian]{babel}
%% Copyright 2011 Mario Rossi
%%
%% Distributable under the LaTeX Project Public License,
%% version 1.3c or higher (your choice). The latest version of
%% this license is at: http://www.latex-project.org/lppl.txt
%%
%% This work is "author-maintained"
%%
%% This work consists of this file libro.dtx,
%% and the derived files libro.cls and libro.pdf.
%%
%%
%% End of file ‘libro.cls’.
```

Il secondo file è mostrato nella pagina 10.

I comandi `\StopEventually` e `\Finale` nella quarta parte racchiudono il codice commentato. Il comando `\Finale` controlla anche la “check sum”, che deve risultare uguale a quella specificata nel comando `\CheckSum`; essa rappresenta il conteggio di tutti i segni `\` incontrati nel codice; non è un conteggio infallibile, ma è una buona assicurazione che se il file contiene il numero giusto di backslash è quasi certamente integro attraverso i passaggi per la posta elettronica, gli algoritmi di compressione e di decompressione, ed altri accidenti a cui sono sottoposti i file.

I due file suddetti vanno spostati in due cartelle diverse nel proprio albero personale; dove sia radicato questo albero dipende dal sistema operativo usato, ma leggendo la documentazione di TEX Live (Berry, 2011) (`texdoc texlive`) c’è un paragrafo apposta che spiega le differenze nei vari sistemi operativi.

Comunque nel proprio albero personale (che va creato se non lo si è mai fatto prima) bisogna creare la cartella `.../doc/latex/libro/`[1] e lì spostare il file `libro.pdf`; poi bisogna creare la cartella `.../tex/latex/libro/` e lì bisogna spostare il file `libro.cls`; se si vuole conservare il file

[1] I tre puntini indicano la radice dell’albero personale; i separatori delle cartelle sono indicati con lo slash; nei sistemi Windows bisogna usare il backslash.

La classe `libro`*

Mario Rossi†

30 ottobre 2012

Sommario

Questa classe serve per comporre libri in italiano appoggiandosi alla classe `book` ma includendo i pacchetti necessari.

1 Generalità

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

- La classe di supporto è la classe standard del system TEX `book.cls`.
- Tutte le opzioni inserite nell'istruzione `\documentclass` vengono trasferite integralmente alla classe `book`.
- La codifica di entrata è fissata con la specifica `utf8` e non è possibile con questa classe specificare una codifica differente.
- La codifica dei font di uscita 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.
- 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.

2 Codice commentato

Le due righe contenti 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 `book`

```
\DeclareOption*{\PassOptionToClass{\CurrentOption}{book}}
\ProcessOptions*\relax
\LoadClass{book}
```

Poi carichiamo i pacchetti relativi alle codifiche di ingresso e di uscita:

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

*Version number v.0.4; ultima revisione 2011/11/25.

†e-mail: `mario dot rossi at mymail dot net`

Scegliamo per l'uscita un font eccellente come Latin Modern, ma lo accompagniamo con il pacchetto textttextcomp così da completarlo con molti altri segni alfabetici e analfabetici.

```
\usepackage{lmodern,textcomp}
```

Infine chiamiamo `babel` per scegliere la lingua del documento.

```
\usepackage[italian]{babel}
```

Questo è tutto.

`libro.dtx`, come è opportuno fare, lo si mette nella cartella `.../source/latex/libro/`. Con queste operazioni la documentazione è accessibile; il file documentato è accessibile per possibili modifiche, e il file di classe è immediatamente usabile[2].

2.1.2 Un'altra classe semplicissima

Questa classe non è così semplice come la precedente, e non scriverò il file `.dtx` per contenere entro limiti ragionevoli questa guida tematica. Si è certamente notato quante pagine sono risultate necessarie per descrivere una classe che in fondo ha solo sette righe di codice. Il tutto sembra quasi sprecato (e in certo senso lo è, per una classe così semplice) ma avendo a che fare con classi più complesse è l'unico modo per mantenere ordinato e comprensibile il lavoro nel tempo. Quindi, anche se nel seguito non mostrerò altri file `.dtx`, ci si ricordi *sempre* di impostare il lavoro attraverso questo file documentato.

Ecco l'esempio. Il prof. Tizio sta scrivendo un trattato di tricotetratomia (*taglio del capello in quattro* ma, se si tratta di un trattato, l'Accademia usa solo termini di origine greca, altrimenti tutti potrebbero capire di che cosa si parla). Egli comincia a impostare il trattato con la classe *`book`*, ma via via che procede nella redazione di questo fondamentale documento, deve definire comandi e ambienti nuovi rispetto a quelli predefiniti in quella classe; si accorge che deve anche cambiare i margini e la gabbia di composizione del testo. Così facendo, il preambolo del documento diventa molto lungo e non riesce più a gestire comodamente i file sorgente. Decide allora di raccogliere i suoi comandi e ambienti nuovi in un file di estensione `.sty`; attraverso un file `.dtx` provvede a documentarli per bene; sta attento affinché il file documentato inserisca in testa al file di estensione `tttextension.sty` le specificazioni:

```
\NeedsTeXFormat{LaTeX2e}[2009/01/01]
\ProvidesPackage{tttextension}%
    [2011/11/25 v.1.0 Estensione per la classe tttclass]
```

Per la classe `tttclass.cls` si assicurerà che il file documentato inserisca in testa al file `.cls` le righe:

[2] TEX Live non richiede operazioni speciali; altre distribuzioni potrebbero richiedere l'aggiornamento del database dei nomi dei file prima di poterli usare.

```
\NeedsTeXFormat{LaTeX2e}[2009/01/01]
\ProvidesClass{tttclass}%
 [2011/11/25 v.1.0 Classe per il trattato di TricoTetraTomia]
```

nel quale dichiarerà che tutte le opzioni sono da trasferire alla classe book e caricherà la classe book. Nello stesso tempo decide di caricare il pacchetto *geometry* per ridefinire la gabbia di composizione del testo e poi caricherà il suo pacchetto di estensione:

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{book}}
\ProcessOptions*\relax
\LoadClass{book}
\RequirePackage[total={140mm,250mm},top=30mm,%
                                          left=20mm]{geometry}
\RequirePackage{tttextension}
```

In questo modo avrà un testo in cui la gabbia di composizione è di 140 mm per 250 mm, il margine superiore di 30 mm e il margine interno di 20 mm; le testatine e il piedini non fanno parte della gabbia di composizione e le distanze di questi elementi dalla gabbia restano quelle definite nella classe *book*.

Ovviamente queste osservazioni andranno nel file, chiamato per esempio `ttt.dtx`, il quale può contenere sia il codice per la classe, sia quello per il pacchetto di estensione, con marcatori diversi; il codice per l'estrazione dei rispettivi file (la prima parte del file `.dtx`) conterrà quindi due righe del tipo:

```
\generate{\file{tttclass.cls}{\from{ttt.dtx}{class}}}
\generate{\file{tttextension.sty}{\from{ttt.dtx}{package}}}
```

cosicché lanciando pdfLaTeX sul file `ttt.dtx` si produrranno un unico file di documentazione e due file, `tttclass.cls` e `tttextension.sty`, avendo specificato rispettivamente i nomi dei due marcatori `class` e `package` ai comandi `\generate`.

L'idea di tenere le macro personali in un file di estensione separato è utile perché, indipendentemente da quella in lavorazione, lo si può caricare anche quando si usano altre classi.

2.2 Creazione di una classe mediante personalizzazioni

Una classe può essere creata anche a partire da altre classi; una molto versatile e fortemente personalizzabile in ogni suo dettaglio è la classe *memoir* (Wilson e Madsen, 2011); merita leggere i capitoli 20 e 21 della sua documentazione (leggibile come al solito con `texdoc memoir`).

2.2.1 Una classe per libri con lo stile di Bringhurst

Nel capitolo 20 di quella documentazione si descrive come realizzare una personalizzazione in modo da comporre un libro con lo stile del testo di Robert Bringhurst *The Elements of typographic style* (Bringhurst, 2004), mentre nel capitolo 21 si descrive come predisporre una classe per comporre una tesi.

Nel primo caso si potrebbe costruire un classe *Bringhurstbook.cls* cominciando con le solite poche righe:

```
\NeedsTeXFormat{LaTeX2e}[2009/01/01]
\ProvidesClass{Bringhurstbook}%
   [2011/11/25 v.1.0 Classe per libri alla Bringhurst]
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
\ProcessOptions*\relax
\LoadClass{memoir}
```

per poi aggiungerci tutte le personalizzazioni copiandole dal capitolo 20 della documentazione di *memoir*. Quel capitolo spiega anche le motivazioni, le specifiche, il modo di ottenerle e anche il codice usato: tutte cose da aggiungere al file documentato `Bringhurstbook.dtx`. Ma il bello è che i comandi copiati dal capitolo 20 possono servire da modello per apportare ulteriori modifiche: basta infatti modificare ulteriormente gli argomenti dei comandi di personalizzazione.

Si noti che la classe *memoir* emula già da sola una moltitudine di pacchetti e altri ne carica apposta quindi, una volta fatta la personalizzazione suddetta, di solito non è necessario aggiungere alcunché. È bene lasciare la scelta delle codifiche, delle lingue e dei font nel preambolo del documento che si comporrà con questa classe; in questo modo si è liberi di specificare queste cose nei modi propri dei programmi di composizione pdfLaTeX oppure XeLaTeX che richiedono forme particolari diverse.

2.2.2 Una classe per tesi con stile americano

Nel manuale di *memoir* al capitolo 21 si parla di configurare la classe per comporre una tesi secondo le specifiche di una ipotetica università ASU, *Archibald Smythe University*, le cui specifiche sono molto severe, come succede spesso negli Stati Uniti; in Europa le specifiche sono spesso più elastiche per cui in Europa, secondo Wilson, si possono ottenere disegni editoriali migliori; ecco quindi che servirsi della classe *memoir* con le sue infinite possibilità di personalizzazione diventa un'ottima strada per costruire una classe a sé stante.

Ci vorranno sempre le righe iniziali nel file di classe `ASUthesis.cls`:

```
\NeedsTeXFormat{LaTeX2e}[2009/01/01]
\ProvidesClass{ASUthesis}%
   [2011/11/25 v.1.0 Classe per tesi presso
             la Archibald Smythe University]
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
\ProcessOptions*\relax
\LoadClass{memoir}
```

Poi bisogna copiare dal manuale di *memoir* sia le spiegazioni, sia i vari blocchi di codice per inserire il tutto dentro il file documentato `ASUthesis.dtx`; infine bisogna modificare gli argomenti dei comandi di configurazione per adattarli al formato di tesi che si desidera; bisognerà anche, per una tesi in italiano, sostituire parole come "Supervisor" con "Relatore"; sarà anche il caso di cambiare il nome `ASUthesis` in qualcosa di più consono alla classe così ottenuta.

Ci si ricordi che la spaziatura doppia per le bozze, se usata, va eliminata quando si compone la bella copia, quindi quando si esegue la composizione definitiva.

Tuttavia il grosso è fatto; l'impalcatura c'è ed è possibile modificarla fino a quando il risultato è esteticamente soddisfacente, ma tenendo in mente che l'eleganza del disegno grafico non deve distrarre il lettore né, al contrario, metterlo a disagio con linee troppo lunghe in relazione al corpo o con cambi di font troppo numerosi; sarà un filino eccessivo, ma è meglio ricordare che secondo Robert Bringhurst non si dovrebbero mai usare font della serie nera, e fra le forme usabili ci sarebbero solo il tondo, il corsivo e il maiuscoletto; ogni altra forma sarebbe sconsigliata.

Ovviamente è esagerato: come si potrebbe scrivere in questo modo una tesi di informatica senza distinguere con un font decisamente contrastante (come il typewriter type) le parole che appartengono ad un linguaggio di programmazione da quelle che appartengono alla lingua corrente?

2.2.3 Nuova classe usando pacchetti di configurazione

Un altro metodo, invece di usare una classe altamente configurabile come *memoir*, è quello di usare una classe standard e modificare le impostazioni grafiche mediante pacchetti di configurazione.

In questo modo, dopo le prime cinque righe per identificare la classe, per specificare il nome del file di formato che contiene le definizioni del nucleo del linguaggio di mark-up, per dichiarare che tutte le opzioni devono venire passate integralmente alla classe standard prescelta, bisogna caricare tutti i pacchetti che servono per la configurazione e usare i loro comandi per configurare il disegno e i font della nuova classe che si vuole definire. I pacchetti più importanti (riportati anche nella bibliografia) sono:

geometry per selezionare le dimensioni della gabbia di composizione, i margini, la larghezza o l'altezza di diversi elementi, come le note marginali, le testatine, i piedini, i rientri dei capoversi, gli spazi prima e dopo gli elementi in display, mobili o fissi che siano, eccetera.

tocloft per configurare il modo di comporre l'indice generale e gli elenchi di figure e tabelle, oltre che degli altri elementi flottanti creati con il pacchetto *float* (Lingnau, 2001); si possono configurare anche gli indici analitici e i glossari, almeno per quel che riguarda il lato LaTeX, non certo quello che riguarda i file di stile che deve usare makeindex per alfabetizzare e strutturare questi indici.

titlesec per configurare ogni genere di titolo, da quello principale del documento ai titoli dei capitoli, dei paragrafi, e di ogni altra sezione di rango inferiore, scegliendo famiglie, corpi, serie e forme dei font da usare e il modo di comporre di quei titoli o titolini.

caption per configurare il modo di comporre le didascalie sia sotto l'aspetto dei font usati sia per il tipo di composizione; in particolare il pacchetto riesce a configurare le didascalie da mettere sopra le tabelle diversamente dalle didascalie da mettere sotto le figure: in buona

sostanza lo spazio che separa la didascalia dall'oggetto che essa deve identificare.

subcaption oppure *subfig* per poter avere più figure o tabelle dentro lo stesso ambiente mobile, che quindi potrà avere una didascalia generale, ma ogni figurina o tabellina può avere la sua piccola didascalia identificativa alla quale ci si può riferire con i soliti comandi `\label` e `\ref`.

fancyhdr sia per configurare nuove testatine e piedini, sia per riconfigurare quelli della classe principale: in sostanza, per configurare o riconfigurare gli stili delle pagine. La cosa notevole di questo pacchetto è che riesce a configurare con pochi semplici comandi le testatine e i piedini delle pagine sia destre sia sinistre, curando che in ogni testatina o piedino si possano mettere fino a tre informazioni distinte; ogni pagina quindi può contenere sei informazioni distinte e due pagine affacciate fino a dodici informazioni distinte e diverse da quelle della pagina a fronte. Permette anche di configurare questi elementi con testatine e piedini che sporgono nel margine esterno essendo più larghi della giustezza della gabbia di composizione, un po' come nel manuale di LaTeX predisposto da Leslie Lamport. Comodo e facile da usare.

enumitem serve per configurare tutti gli aspetti di composizione delle liste create con gli ambienti *enumerate*, *itemize*, *description*; definisce anche nuovi ambienti compatti senza spazi per separare le varie voci.

varioref serve per accontentare tutti coloro che vogliono usare espressioni come "la figura 3.3 nella pagina seguente"; bisogna invocare questo pacchetto con l'opzione specifica della lingua principale, altrimenti la parte testuale dell'indicazione risulta in un'altra lingua, e perciò "stonata".

hyperref se si vuole che la classe abbia i riferimenti ipertestuali da un punto all'altro del documento, tipicamente per comporre documenti da leggere a video; siccome *hyperref* deve essere l'*ultimo* pacchetto caricato è meglio configurarne le prestazioni nel preambolo del documento piuttosto che dentro la classe, a meno che non si desideri bloccare apposta la configurazione ipertestuale della classe.

Ognuno di questi pacchetti è ben documentato e la documentazione in inglese si può facilmente leggere con il solito comando `texdoc` ⟨*nome del*

pacchetto⟩ come indicato nella bibliografia. Esiste una documentazione più concisa in italiano nel documento di Enrico Gregorio *LaTeX, breve guida ai pacchetti di uso più comune* (Gregorio, 2010a) scaricabile dal sito http://www.guitex.org/home/ selezionando la targhetta "Documentazione" e poi nel menù "Guide per iniziare" selezionando le "Guide di Enrico Gregorio".

È assolutamente evidente che qui un esempio sarebbe un'inutile ripetizione degli esempi che si trovano nell'ultimo riferimento; la gestione delle configurazioni non è difficile, basta leggere la documentazione, ma è lunga per la quantità di informazioni che bisogna specificare.

Tuttavia conviene preparare un primo abbozzo di classe senza caricare nessun pacchetto configurabile. Contemporaneamente si prepara un documento `TestClasse.tex` che usi quella bozza di classe e che contenga ogni genere di strutture: pagina del titolo, parti, capitoli, paragrafi, sottoparagrafi, eccetera, figure e tabelle con le loro didascalie, teoremi, definizioni e simili enunciati, figure e tabelle semplici o formate da diversi elementi, stili di pagina diversi, liste di ogni tipo, collegamenti ipertestuali. Ovviamente il testo sarà composto dentro una gabbia le cui dimensioni possono venire collaudate via via.

Il file `TestClasse.tex` deve compilarsi senza nessun problema con la classe di base. Poi si inserisce il primo pacchetto di configurazione, se ne impostano i parametri e si compila `TestClasse.tex`; successivamente si riconfigura e si controlla, finché le impostazioni non sono soddisfacenti. Poi si aggiunge un altro pacchetto e con questo si eseguono le nuove configurazioni che esso gestisce, ripetendo finché basta; si continua così fin quando la nuova classe non è più una bozza ma è diventata la classe definitiva. Descritto come sopra, sembra un lavoro terribile; in realtà, a seconda dell'abilità di chi scrive la classe e della quantità di configurazioni da eseguire, richiede da una a tre ore, spese soprattuto a leggere la documentazione dei vari pacchetti. Il tempo relativamente lungo per costruire una classe abbastanza complessa come descritto qui è ricompensato dal risultato generalmente molto buono, per non dire eccellente. L'eccellenza, in realtà, si raggiunge quando si capisce che i font scelti, la gabbia di composizione, la distribuzione dei vari elementi sulla pagina, eccetera, danno un senso di agio al solo guardare il libro aperto. Anzi, si cerchi di stampare un paio di pagine affacciate in formato vero in modo da poter valutare l'agio della lettura; in alternativa, si imposti il visualizzatore PDF

in modo che mostri due pagine affacciate e se ne configuri l'ingrandimento, schermo permettendo, che mostri le pagine in scala 1 : 1. Solo guardando due pagine affacciate si può valutare l'effetto del disegno grafico unito alla particolare scelta dei font. Quando si raggiunge il meglio l'autore se ne accorge subito, sempre che abbia ben chiaro in mente quello che riguarda le basi dell'arte tipografica e non scambi per agio di lettura l'apprezzamento delle forme e dei colori.

Produrre una classe partendo da zero

3

Non è impossibile produrre una classe partendo da zero, ma oggigiorno è un po' masochistico. Con il vecchio LaTeX 209 ne ho dovute preparare molte perché allora non si disponeva di pacchetti di configurazione. In pratica si prendeva una classe standard, le si cambiava nome e vi si introducevano tutte le modifiche e le aggiunte che si volevano. Sembra facile, ma non è vero; ogni modifica implica la scrittura o la modifica di brani di programmazione scritti da altri, poco o nulla documentati, e generalmente scritti in linguaggio TeX primitivo, o con comandi di Plain TeX, o con comandi del nucleo di LaTeX.

3.1 La via breve

Oggi farlo è certamente più facile, se non altro perché si possono usare i pacchetti di personalizzazione, con i quali si fa la maggior parte del lavoro. Poi bisogna aggiungere un certo numero di nuovi ambienti o comandi un po' particolari che non si possono direttamente scrivere con i comandi di definizione (`\newcommand`) o di dichiarazione (`\Declare...`) ad alto livello forniti dal nucleo di LaTeX.

A questo punto sarebbe meglio avere una certa dimestichezza con i comandi del nucleo di LaTeX (non tutti, certo, ma una buona dimestichezza, sì, ci vuole), bisogna saper distinguere le definizioni e i comandi con argomenti delimitati da quelli con argomenti non delimitati. Bisogna sapere destreggiarsi con i comandi `\@ifundefined` e `\@ifdefinable` e le loro sottili differenze. Meglio ancora sarebbe studiarsi a fondo la documentazione del pacchetto *etoolbox* (Lehman, 2011b), la scatola degli attrezzi di pdfTeX esteso, sapendo che le potenzialità dell'attuale pdfTeX e di XeTeX sono

molto maggiori della versione knuthiana di TeX. Caricando nella futura classe il pacchetto *etoolbox* si possono sfruttare tutti i suoi potentissimi strumenti per maneggiare e modificare correttamente macro preesistenti; per eseguire test inimmaginabili con i comandi primitivi di TEX; per dilazionare l'esecuzione di comandi anche di definizione o di dichiarazione all'inizio dell'esecuzione dell'istruzione `\begin{document}`, a metà o alla fine (i risultati possono essere assai diversi); e tante altre cose, compresa la possibilità di eseguire calcoli direttamente in memoria, eliminando i difetti dei comandi primitivi, e senza bisogno di ricorrere al pacchetto *calc*.

Ci si ricordi che i registri di ogni genere non sono più limitati a 256 come nella versione knuthiana, ma possono essere un numero sterminato, difficilissimo da raggiungere; per venire allocati correttamente per nome è necessaria la presenza del pacchetto *etex*, che è già caricato da *etoolbox*.

Un'altra cosa a cui bisogna stare attenti è che il carattere `@` dentro i file di classe e dentro i file di estensione può avere lo status di lettera; quando viene eseguito il comando `\begin{document}` il suo status viene ripristinato a quello di un carattere analfabetico; dentro i file di classe i nomi dei comandi, che devono essere formati solo da lettere maiuscole e minuscole, possono contenere anche la 'lettera' `@`. Non c'è quindi bisogno di usare comandi come `\makeatletter` per cambiare status a quel carattere, né di usare `\makeatother` dopo il brano di codice che lo usa nei nomi dei comandi.

In questa pagina ho messo come sfondo del testo il disegno in rosso dello schema grafico della gabbia di composizione. Nel preambolo del file sorgente di questo file ho inserito le definizioni seguenti:

```
% Layout disegnato e definizione di uno stile di pagina
\newcommand*\cblayoutdestro{%
\dimen256=
   \dimexpr\headheight+\topmargin+1in-\paperheight\relax
\dimen258=\dimexpr\oddsidemargin+1in\relax
\dimen260=\dimexpr \paperheight-\topmargin-
      \headheight-1in-\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)
\put(0,0){\unitlength=\p@
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen260)%
```

```
     {\makebox(0,0)[bl]{%
        \framebox(\strip@pt\textwidth,\strip@pt\textheight){}
        }}
\Line(0,\strip@pt\paperheight)(\strip@pt\paperwidth,0)
\put(\strip@pt\dimexpr\paperwidth/2\relax,%
     \strip@pt\dimexpr\dimen260+\textheight/2\relax)%
        {\circle{\strip@pt\paperwidth}}
}
\end{picture}}
%
\newcommand*\ps@headingslayout{\ps@headings
    \toks@=\expandafter{\@oddhead}%
    \edef\@oddhead{\noexpand\cblayoutdestro\the\toks@}%
}
```

Indipendentemente dall'uso specifico che se ne fa, questo è il genere di comandi che si possono o può essere necessario inserire in un file di classe. Si nota la presenza del carattere `@` nei nomi dei comandi; si nota l'asterisco dopo il comando `\newcommand`, che classifica la macro così definita come una macro *non* avente la caratteristica di "`long`", quindi, se dovesse accettare in ingresso degli argomenti, questi non potrebbero contenere un qualsiasi comando che rappresenti un fine capoverso, che sia una riga vuota o il comando `\par` o qualunque cosa equivalente a queste.

Nella macro `\cblayoutdestro` si comincia col caricare dei registri "lunghezza" numerati esplicitamente con numeri superiori a 255, grazie alla versione *estesa* dei programmi di composizione pdfLaTeX e XeLaTeX; questi programmi sono estesi fin dal 2005, quindi qualunque distribuzione recente del sistema TEX li usa di default. Si usano i numeri pari, perché si conviene di usare questa parità per i registri "scratch", destinati a conservare informazioni che interessano solo localmente e che non interessa se vengono poi alterate da altre macro.

Le espressioni che vengono determinate con `\dimexpr` e terminate con `\relax` sono assolutamente esplicite e non richiedono spiegazioni. Servono per calcolare lo spostamento che la figura di dimensioni nulle definita poco dopo, e che verrà posta all'inizio della testatina di destra (l'unica usata in questo documento composto solo in bianca – solo fronte) ha rispetto al vertice in alto a sinistra del foglio di carta.

Subito dopo si definisce un disegno di dimensioni nulle; l'unità di misura non è rilevante, che siano punti, centimetri o pollici, il valore zero è sempre lo stesso. Ma dentro la figura si definisce come unità di misura il punto tipografico. Prima si definisce che si vuole colorare il disegno in rosso poi nell'origine di questo disegno vene collocato il disegno vero e proprio e per prima cosa si usa un comando `\put` per metterlo al suo posto con le coordinate determinate sopra per lo spostamento, in modo che il disegno della gabbia di composizione sia esattamente al suo posto.

Poi si disegna la diagonale del foglio con uno dei comandi aggiunti dal pacchetto di estensione *pict2e*, che estende le funzionalità dell'ambiente *picture* del nucleo di LaTeX. Infine si mette in posizione il cerchio di diametro pari alla base del foglio di carta. Sulla base di questo disegno, il pacchetto *canoniclayout* (Beccari, 2011b), caricato nella classe di questo stesso documento, ha creato il layout della pagina.

La successiva definizione permette di dichiarare un layout della pagina uguale al layout `headings`, ma alla testatina dispari è stato aggiunto il precedente comando `\cblayoutdestro`, che non occupa spazio (disegno di dimensioni nulle) ma che viene inserito per primo dalla routine di output di LaTeX per cui il testo della pagina vi si sovrappone.

Si noti come si è fatto per aggiungere il comando alla precedente definizione di `\@oddhead`: si è prima caricata la sua espansione nel registro 'token' numero zero, rappresentato dal comando `\toks@`, poi si è ridefinito lo stesso comando espanso mediante il comando `\edef` (non disponibile in LaTeX, ma compreso nel linguaggio primitivo TeX) stando attenti a non espandere `\cblayoutdestro`, la cui espansione sarebbe del tutto inutile, mentre l'espansione di `\toks@` rimette nello stesso ordine dentro la macro le parti che conteneva anche in precedenza e che erano state conservate nel registro 'token'. Al posto di questo apparente giro "di conigli e cappelli a cilindro", si sarebbe potuto usare il comando definito da *etoolbox* `\cspreto` in questo modo:

```
\cspreto{@oddhead}{\cblayoutdestro}
```

e questo esempio mostra al di là di ogni dubbio quanto possano essere utili le macro definite nel pacchetto *etoolbox*.

3.2 La struttura della classe *book*

Per meglio comprendere di che cosa le classi standard (che si possono usare come modelli per costruire classi partendo da zero) sono tenute ad occuparsi, ora esaminiamo per sezioni la struttura della classe *book*, senza scendere nei dettagli riga per riga; con questa descrizione, la lettura del codice vero e proprio nel file `TEXMF/texmf-dist/tex/latex/base/book.cls` dovrebbe risultare agevolata; il file con le righe numerate viene riportato qui di seguito ma si suggerisce di aprire il file suddetto e di tenerne aperta la finestra accanto a quella dove si sta leggendo questa documentazione; in questo modo è più facile seguire la descrizione e leggere il codice. Possibilmente si imposti il programma per visualizzare il file `book.cls` in modo che mostri a sinistra di ogni riga il suo numero progressivo; in questo modo le indicazioni del tipo "la seconda sezione si svolge dalla riga... alla riga..." diventa comprensibile e permette di localizzare meglio nel file sorgente il blocco che viene descritto.

Si noti che la classe *book* è concepita anche per funzionare con documenti precedenti al 1994, anno di nascita di LaTeX 2_ε, e quindi contiene moltissime istanze del test `\if@compatibility`; nel leggere il file si possono saltare sistematicamente i blocchi di codice corrispondenti all'esito "true" del test e leggere solo quello che appare dopo il comando `\else` fino alla fine segnata dal comando `\fi`. Se si usa questa classe come modello per creare una nuova classe, dopo averlo copiato in un altro file con nome diverso, la prima modifica da fare è quella di eliminare tutti questi test e il ramo relativo alla compatibilità col passato; si elimina circa la metà dell'intero file.

Nel nuovo file si cambi nella riga 2 il nome del file, si cancelli tutto quanto dalla riga 3 alla riga 53, si modifichi di nuovo il nome del file (senza estensione) nella riga 55, e si cambi data, versione e descrizione nelle righe 56 e 57. Non si dimentichino mai queste operazioni ogni qual volta si crei un nuovo file partendo da un file preesistente.

```
1 %%
2 %% This is file ‘book.cls’,
3 %% generated with the docstrip utility.
4 %%
5 %% The original source files were:
6 %%
7 %% classes.dtx  (with options: ‘book’)
8 %%
```

```
%% This is a generated file.
%%
%% The source is maintained by the LaTeX Project team and bug
%% reports for it can be opened at http://latex-project.org/bugs.html
%% (but please observe conditions on bug reports sent to that address!)
%%
%%
%% Copyright 1993-2016
%% The LaTeX3 Project and any individual authors listed elsewhere
%% in this file.
%%
%% This file was generated from file(s) of the LaTeX base system.
%% ---------------------------------------------------------------
%%
%% It may be distributed and/or modified under the
%% conditions of the LaTeX Project Public License, either version 1.3c
%% of this license or (at your option) any later version.
%% The latest version of this license is in
%%    http://www.latex-project.org/lppl.txt
%% and version 1.3c or later is part of all distributions of LaTeX
%% version 2005/12/01 or later.
%%
%% This file has the LPPL maintenance status "maintained".
%%
%% This file may only be distributed together with a copy of the LaTeX
%% base system. You may however distribute the LaTeX base system without
%% such generated files.
%%
%% The list of all files belonging to the LaTeX base distribution is
%% given in the file `manifest.txt'. See also `legal.txt' for additional
%% information.
%%
%% The list of derived (unpacked) files belonging to the distribution
%% and covered by LPPL is defined by the unpacking scripts (with
%% extension .ins) which are part of the distribution.
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesClass{book}
              [2014/09/29 v1.4h
 Standard LaTeX document class]
\newcommand\@ptsize{}
\newif\if@restonecol
\newif\if@titlepage
\@titlepagetrue
\newif\if@openright
\newif\if@mainmatter \@mainmattertrue
\if@compatibility\else
\DeclareOption{a4paper}
   {\setlength\paperheight {297mm}%
    \setlength\paperwidth  {210mm}}
\DeclareOption{a5paper}
   {\setlength\paperheight {210mm}%
    \setlength\paperwidth  {148mm}}
```

```
\DeclareOption{b5paper}
   {\setlength\paperheight {250mm}%
    \setlength\paperwidth  {176mm}}
\DeclareOption{letterpaper}
   {\setlength\paperheight {11in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{legalpaper}
   {\setlength\paperheight {14in}%
    \setlength\paperwidth  {8.5in}}
\DeclareOption{executivepaper}
   {\setlength\paperheight {10.5in}%
    \setlength\paperwidth  {7.25in}}
\DeclareOption{landscape}
   {\setlength\@tempdima   {\paperheight}%
    \setlength\paperheight {\paperwidth}%
    \setlength\paperwidth  {\@tempdima}}
\fi
\if@compatibility
  \renewcommand\@ptsize{0}
\else
\DeclareOption{10pt}{\renewcommand\@ptsize{0}}
\fi
\DeclareOption{11pt}{\renewcommand\@ptsize{1}}
\DeclareOption{12pt}{\renewcommand\@ptsize{2}}
\if@compatibility\else
\DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\fi
\DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\if@compatibility\else
\DeclareOption{final}{\setlength\overfullrule{0pt}}
\fi
\DeclareOption{titlepage}{\@titlepagetrue}
\if@compatibility\else
\DeclareOption{notitlepage}{\@titlepagefalse}
\fi
\if@compatibility
\@openrighttrue
\else
\DeclareOption{openright}{\@openrighttrue}
\DeclareOption{openany}{\@openrightfalse}
\fi
\if@compatibility\else
\DeclareOption{onecolumn}{\@twocolumnfalse}
\fi
\DeclareOption{twocolumn}{\@twocolumntrue}
\DeclareOption{leqno}{\input{leqno.clo}}
\DeclareOption{fleqn}{\input{fleqn.clo}}
\DeclareOption{openbib}{%
  \AtEndOfPackage{%
   \renewcommand\@openbib@code{%
      \advance\leftmargin\bibindent
```

```
      \itemindent -\bibindent
      \listparindent \itemindent
      \parsep \z@
      }%
   \renewcommand\newblock{\par}}%
}
\ExecuteOptions{letterpaper,10pt,twoside,onecolumn,final,openright}
\ProcessOptions
\input{bk1\@ptsize.clo}
\setlength\lineskip{1\p@}
\setlength\normallineskip{1\p@}
\renewcommand\baselinestretch{}
\setlength\parskip{0\p@ \@plus \p@}
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
\setcounter{topnumber}{2}
\renewcommand\topfraction{.7}
\setcounter{bottomnumber}{1}
\renewcommand\bottomfraction{.3}
\setcounter{totalnumber}{3}
\renewcommand\textfraction{.2}
\renewcommand\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\renewcommand\dbltopfraction{.7}
\renewcommand\dblfloatpagefraction{.5}
\if@twoside
  \def\ps@headings{%
      \let\@oddfoot\@empty\let\@evenfoot\@empty
      \def\@evenhead{\thepage\hfil\slshape\leftmark}%
      \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
      \let\@mkboth\markboth
    \def\chaptermark##1{%
      \markboth {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \if@mainmatter
            \@chapapp\ \thechapter. \ %
          \fi
        \fi
        ##1}}{}}%
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\z@
          \thesection. \ %
        \fi
        ##1}}}}
\else
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
    \def\chaptermark##1{%
```

```
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \if@mainmatter
            \@chapapp\ \thechapter. \ %
          \fi
        \fi
        ##1}}}}
\fi
\def\ps@myheadings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\@gobbletwo
    \let\chaptermark\@gobble
    \let\sectionmark\@gobble
    }
  \if@titlepage
  \newcommand\maketitle{\begin{titlepage}%
  \let\footnotesize\small
  \let\footnoterule\relax
  \let \footnote \thanks
  \null\vfil
  \vskip 60\p@
  \begin{center}%
    {\LARGE \@title \par}%
    \vskip 3em%
    {\large
     \lineskip .75em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
      \vskip 1.5em%
    {\large \@date \par}%       % Set date in \large size.
  \end{center}\par
  \@thanks
  \vfil\null
  \end{titlepage}%
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\else
\newcommand\maketitle{\par
  \begingroup
```

```
    \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
    \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
    \long\def\@makefntext##1{\parindent 1em\noindent
            \hb@xt@1.8em{%
                \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
    \if@twocolumn
      \ifnum \col@number=\@ne
        \@maketitle
      \else
        \twocolumn[\@maketitle]%
      \fi
    \else
      \newpage
      \global\@topnum\z@   % Prevents figures from going at top of page.
      \@maketitle
    \fi
    \thispagestyle{plain}\@thanks
  \endgroup
  \setcounter{footnote}{0}%
  \global\let\thanks\relax
  \global\let\maketitle\relax
  \global\let\@maketitle\relax
  \global\let\@thanks\@empty
  \global\let\@author\@empty
  \global\let\@date\@empty
  \global\let\@title\@empty
  \global\let\title\relax
  \global\let\author\relax
  \global\let\date\relax
  \global\let\and\relax
}
\def\@maketitle{%
  \newpage
  \null
  \vskip 2em%
  \begin{center}%
  \let \footnote \thanks
    {\LARGE \@title \par}%
    \vskip 1.5em%
    {\large
      \lineskip .5em%
      \begin{tabular}[t]{c}%
        \@author
      \end{tabular}\par}%
    \vskip 1em%
    {\large \@date}%
  \end{center}%
  \par
  \vskip 1.5em}
\fi
\newcommand*\chaptermark[1]{}
\setcounter{secnumdepth}{2}
```

```
\newcounter {part}
\newcounter {chapter}
\newcounter {section}[chapter]
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\newcounter {subparagraph}[paragraph]
\renewcommand \thepart {\@Roman\c@part}
\renewcommand \thechapter {\@arabic\c@chapter}
\renewcommand \thesection {\thechapter.\@arabic\c@section}
\renewcommand\thesubsection   {\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}
\renewcommand\theparagraph    {\thesubsubsection.\@arabic\c@paragraph}
\renewcommand\thesubparagraph {\theparagraph.\@arabic\c@subparagraph}
\newcommand\@chapapp{\chaptername}
\newcommand\frontmatter{%
    \cleardoublepage
  \@mainmatterfalse
  \pagenumbering{roman}}
\newcommand\mainmatter{%
    \cleardoublepage
  \@mainmattertrue
  \pagenumbering{arabic}}
\newcommand\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}
\newcommand\part{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \thispagestyle{plain}%
  \if@twocolumn
    \onecolumn
    \@tempswatrue
  \else
    \@tempswafalse
  \fi
  \null\vfil
  \secdef\@part\@spart}

\def\@part[#1]#2{%
    \ifnum \c@secnumdepth >-2\relax
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \else
      \addcontentsline{toc}{part}{#1}%
```

```
    \fi
    \markboth{}{}%
    {\centering
     \interlinepenalty \@M
     \normalfont
     \ifnum \c@secnumdepth >-2\relax
       \huge\bfseries \partname\nobreakspace\thepart
       \par
       \vskip 20\p@
     \fi
     \Huge \bfseries #2\par}%
    \@endpart}
\def\@spart#1{%
    {\centering
     \interlinepenalty \@M
     \normalfont
     \Huge \bfseries #1\par}%
    \@endpart}
\def\@endpart{\vfil\newpage
              \if@twoside
               \if@openright
                \null
                \thispagestyle{empty}%
                \newpage
               \fi
              \fi
              \if@tempswa
                \twocolumn
              \fi}
\newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
                    \thispagestyle{plain}%
                    \global\@topnum\z@
                    \@afterindentfalse
                    \secdef\@chapter\@schapter}
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
                       \if@mainmatter
                         \refstepcounter{chapter}%
                         \typeout{\@chapapp\space\thechapter.}%
                         \addcontentsline{toc}{chapter}%
                                   {\protect\numberline{\thechapter}#1}%
                       \else
                         \addcontentsline{toc}{chapter}{#1}%
                       \fi
                    \else
                      \addcontentsline{toc}{chapter}{#1}%
                    \fi
                    \chaptermark{#1}%
                    \addtocontents{lof}{\protect\addvspace{10\p@}}%
                    \addtocontents{lot}{\protect\addvspace{10\p@}}%
                    \if@twocolumn
                      \@topnewpage[\@makechapterhead{#2}]%
                    \else
```

```
                      \@makechapterhead{#2}%
                      \@afterheading
                    \fi}
\def\@makechapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \huge\bfseries \@chapapp\space \thechapter
        \par\nobreak
        \vskip 20\p@
      \fi
    \fi
    \interlinepenalty\@M
    \Huge \bfseries #1\par\nobreak
    \vskip 40\p@
  }}
\def\@schapter#1{\if@twocolumn
                   \@topnewpage[\@makeschapterhead{#1}]%
                 \else
                   \@makeschapterhead{#1}%
                   \@afterheading
                 \fi}
\def\@makeschapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright
    \normalfont
    \interlinepenalty\@M
    \Huge \bfseries  #1\par\nobreak
    \vskip 40\p@
  }}
\newcommand\section{\@startsection {section}{1}{\z@}%
                                   {-3.5ex \@plus -1ex \@minus -.2ex}%
                                   {2.3ex \@plus.2ex}%
                                   {\normalfont\Large\bfseries}}
\newcommand\subsection{\@startsection{subsection}{2}{\z@}%
                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\large\bfseries}}
\newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\normalsize\bfseries}}
\newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
                                    {3.25ex \@plus1ex \@minus.2ex}%
                                    {-1em}%
                                    {\normalfont\normalsize\bfseries}}
\newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                       {3.25ex \@plus1ex \@minus .2ex}%
                                       {-1em}%
                                      {\normalfont\normalsize\bfseries}}
\if@twocolumn
```

```
  \setlength\leftmargini  {2em}
\else
  \setlength\leftmargini  {2.5em}
\fi
\leftmargin  \leftmargini
\setlength\leftmarginii  {2.2em}
\setlength\leftmarginiii {1.87em}
\setlength\leftmarginiv  {1.7em}
\if@twocolumn
  \setlength\leftmarginv  {.5em}
  \setlength\leftmarginvi {.5em}
\else
  \setlength\leftmarginv  {1em}
  \setlength\leftmarginvi {1em}
\fi
\setlength  \labelsep  {.5em}
\setlength  \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
\newcommand\labelitemi{\textbullet}
\newcommand\labelitemii{\normalfont\bfseries \textendash}
\newcommand\labelitemiii{\textasteriskcentered}
\newcommand\labelitemiv{\textperiodcentered}
\newenvironment{description}
               {\list{}{\labelwidth\z@ \itemindent-\leftmargin
                        \let\makelabel\descriptionlabel}}
               {\endlist}
\newcommand*\descriptionlabel[1]{\hspace\labelsep
                                 \normalfont\bfseries #1}
\newenvironment{verse}
               {\let\\\@centercr
                \list{}{\itemsep      \z@
                        \itemindent   -1.5em%
                        \listparindent\itemindent
                        \rightmargin  \leftmargin
                        \advance\leftmargin 1.5em}%
                \item\relax}
               {\endlist}
\newenvironment{quotation}
```

```
                {\list{}{\listparindent 1.5em%
                         \itemindent    \listparindent
                         \rightmargin   \leftmargin
                         \parsep        \z@ \@plus\p@}%
                 \item\relax}
                {\endlist}
\newenvironment{quote}
                {\list{}{\rightmargin\leftmargin}%
                 \item\relax}
                {\endlist}
\if@compatibility
\newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\z@
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
    }
\else
\newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\@ne
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
     \if@twoside\else
        \setcounter{page}\@ne
     \fi
    }
\fi
\newcommand\appendix{\par
  \setcounter{chapter}{0}%
  \setcounter{section}{0}%
  \gdef\@chapapp{\appendixname}%
  \gdef\thechapter{\@Alph\c@chapter}}
\setlength\arraycolsep{5\p@}
\setlength\tabcolsep{6\p@}
\setlength\arrayrulewidth{.4\p@}
\setlength\doublerulesep{2\p@}
\setlength\tabbingsep{\labelsep}
```

```
\skip\@mpfootins = \skip\footins
\setlength\fboxsep{3\p@}
\setlength\fboxrule{.4\p@}
\@addtoreset {equation}{chapter}
\renewcommand\theequation
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
\newcounter{figure}[chapter]
\renewcommand \thefigure
     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreakspace\thefigure}
\newenvironment{figure}
               {\@float{figure}}
               {\end@float}
\newenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}
\newcounter{table}[chapter]
\renewcommand \thetable
     {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{\tablename\nobreakspace\thetable}
\newenvironment{table}
               {\@float{table}}
               {\end@float}
\newenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{10\p@}
\setlength\belowcaptionskip{0\p@}
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  \sbox\@tempboxa{#1: #2}%
  \ifdim \wd\@tempboxa >\hsize
    #1: #2\par
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
```

```
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
\DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
\newcommand\@pnumwidth{1.55em}
\newcommand\@tocrmarg{2.55em}
\newcommand\@dotsep{4.5}
\setcounter{tocdepth}{2}
\newcommand\tableofcontents{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\contentsname
        \@mkboth{%
           \MakeUppercase\contentsname}{\MakeUppercase\contentsname}}%
    \@starttoc{toc}%
    \if@restonecol\twocolumn\fi
    }
\newcommand*\l@part[2]{%
  \ifnum \c@tocdepth >-2\relax
    \addpenalty{-\@highpenalty}%
    \addvspace{2.25em \@plus\p@}%
    \setlength\@tempdima{3em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      {\leavevmode
       \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
       \nobreak
         \global\@nobreaktrue
         \everypar{\global\@nobreakfalse\everypar{}}%
    \endgroup
  \fi}
\newcommand*\l@chapter[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \vskip 1.0em \@plus\p@
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leavevmode \bfseries
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
      \penalty\@highpenalty
    \endgroup
  \fi}
\newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
\newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
```

```
\newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
\newcommand*\l@subparagraph{\@dottedtocline{5}{12em}{6em}}
\newcommand\listoffigures{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\listfigurename}%
      \@mkboth{\MakeUppercase\listfigurename}%
              {\MakeUppercase\listfigurename}%
    \@starttoc{lof}%
    \if@restonecol\twocolumn\fi
    }
\newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
\newcommand\listoftables{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\listtablename}%
      \@mkboth{%
          \MakeUppercase\listtablename}%
         {\MakeUppercase\listtablename}%
    \@starttoc{lot}%
    \if@restonecol\twocolumn\fi
    }
\let\l@table\l@figure
\newdimen\bibindent
\setlength\bibindent{1.5em}
\newenvironment{thebibliography}[1]
     {\chapter*{\bibname}%
      \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
      \list{\@biblabel{\@arabic\c@enumiv}}%
           {\settowidth\labelwidth{\@biblabel{#1}}%
            \leftmargin\labelwidth
            \advance\leftmargin\labelsep
            \@openbib@code
            \usecounter{enumiv}%
            \let\p@enumiv\@empty
            \renewcommand\theenumiv{\@arabic\c@enumiv}}%
      \sloppy
      \clubpenalty4000
      \@clubpenalty \clubpenalty
      \widowpenalty4000%
      \sfcode`\.\@m}
     {\def\@noitemerr
       {\@latex@warning{Empty `thebibliography' environment}}%
      \endlist}
\newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
\let\@openbib@code\@empty
```

```
\newenvironment{theindex}
               {\if@twocolumn
                  \@restonecolfalse
                \else
                  \@restonecoltrue
                \fi
                \twocolumn[\@makeschapterhead{\indexname}]%
                \@mkboth{\MakeUppercase\indexname}%
                        {\MakeUppercase\indexname}%
                \thispagestyle{plain}\parindent\z@
                \parskip\z@ \@plus .3\p@\relax
                \columnseprule \z@
                \columnsep 35\p@
                \let\item\@idxitem}
               {\if@restonecol\onecolumn\else\clearpage\fi}
\newcommand\@idxitem{\par\hangindent 40\p@}
\newcommand\subitem{\@idxitem \hspace*{20\p@}}
\newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
\newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
\renewcommand\footnoterule{%
  \kern-3\p@
  \hrule\@width.4\columnwidth
  \kern2.6\p@}
\@addtoreset{footnote}{chapter}
\newcommand\@makefntext[1]{%
    \parindent 1em%
    \noindent
    \hb@xt@1.8em{\hss\@makefnmark}#1}
\newcommand\contentsname{Contents}
\newcommand\listfigurename{List of Figures}
\newcommand\listtablename{List of Tables}
\newcommand\bibname{Bibliography}
\newcommand\indexname{Index}
\newcommand\figurename{Figure}
\newcommand\tablename{Table}
\newcommand\partname{Part}
\newcommand\chaptername{Chapter}
\newcommand\appendixname{Appendix}
\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
\setlength\columnsep{10\p@}
\setlength\columnseprule{0\p@}
\pagestyle{headings}
\pagenumbering{arabic}
\if@twoside
\else
  \raggedbottom
\fi
\if@twocolumn
  \twocolumn
```

```
737   \sloppy
738   \flushbottom
739 \else
740   \onecolumn
741 \fi
742 \endinput
743 %%
744 %% End of file ‘book.cls’.
```

3.2.1 Identificazione e opzioni

Questa sezione si riferisce alle righe da 54 a 131.

Le righe relative al file di formato e al tipo e nome di file sono le solite; se si vuole creare una nuova classe, bisogna sempre cambiare la frase `Standard LaTeX document class` in qualcosa di attinente alla nuova classe e non usare mai quella frase per file che non siano prodotti dal LaTeX3 Project Team.

I comandi `\DeclareOption` sono abbastanza numerosi e corrispondono alle varie opzioni che si possono specificare per la classe *`book`*; si noterà che per le opzioni *10pt*, *11pt* e *12pt* si richiede di caricare rispettivamente i file `bk10.clo`, `bk11.clo` e `bk12.clo`; queste operazioni di caricamento vengono dilazionate a dopo l'esecuzione di `\ProcessOptions`. Invece per le opzioni *leqno* e *fleqn* (numeri delle equazioni a sinistra e equazioni allineate a sinistra) viene ordinata la lettura di due altri file, rispettivamente `leqno.clo` e `fleqn.clo`.

Prima di `\ProcessOptions` c'è il comando `\ExecuteOptions` per specificare la lista di opzioni di default, nel caso che non ne fosse stata specificata nessuna nell'argomento facoltativo. Visto che per il formato della carta viene predefinito il formato *letterpaper*, nella nuova classe si inserirà di default *a4paper*, se non altro perché qui in Europa è difficile trovare carta di formato quadrotto[1].

I file `bk1*.clo` definiscono le varie misure della gabbia di composizione e degli spazi verticali da inserire per gli oggetti in display, e gli spazi verticali e orizzontali da usarsi nelle liste a diversi livelli di annidamento; definiscono inoltre qualunque lunghezza che in qualche misura debba essere collegata al corpo normale del font in uso; in realtà tutti questi calcoli sono eseguiti

[1] Il formato letter è di 8,5 pollici di base (215,9 mm) per 11 pollici di altezza (279,4 mm). Quando in Italia esisteva il formato quadrotto, esso corrispondeva a 220 mm di base per 280 mm di altezza, quasi uguale al formato letter.

assumendo l'uso dei font della famiglia Computer Modern, e *prima* che l'utente abbia caricato o specificato l'uso di altri font; si potrebbe fare di meglio mediante il comando `\AtBeginDocument`, che consente di ritardare al principio dell'esecuzione dell'istruzione `\begin{document}` questi calcoli dipendenti dal font in uso.

3.2.2 Spazi tecnici e parametri per gli oggetti mobili

Questa sezione si riferisce alle righe da 132 a 148.

Seguono poi le definizioni degli spazi tecnici come, per esempio, l'interlineatura aggiuntiva, da inserire se due righe consecutive rischiano di interferire tra loro.

I parametri per la gestione delle code degli oggetti mobili riguardano i loro numeri massimi per le posizioni in testa e in calce alla pagina, le frazioni di pagina massime per l'uso degli oggetti mobili in testa o in calce, o la frazione minima della pagina da lasciare per il testo.

È molto importante sapere gestire questi parametri; nella guida (Beccari, 2011a) in un'appendice c'è una spiegazione dettagliata dell'uso di questi parametri; specificarli in modo corretto permette di gestire bene le code degli oggetti mobili.

3.2.3 Lo stile delle pagine

Questa sezione si riferisce alle righe da 149 a 190.

Vengono poi definiti gli stili di pagina `headings` e `myheadings`; gli stili `empty` e `plain` sono definiti nel nucleo di LaTeX.

Si nota che vengono definiti stili diversi a seconda che si componga solo fronte oppure fronte-retro. Magari nella nuova classe da definire si potrebbe evitare di produrre le definizioni per i due casi, se si vuole che la nuova classe venga composta in un solo modo; se fosse così, converrebbe allora ridefinire le opzioni di ingresso *oneside* e *twoside* in modo che non eseguano nessuna differente impostazione ma emettano solo un messaggio di avviso che, comunque sia, il testo verrà composto solo in un modo.

Si nota ancora che le macro `\chaptermark` e `\sectionmark` contengono entrambi il comando `\MakeUppercase`; questa impostazione è uno dei motivi per i quali esistono tanti pacchetti o classi configurabili per riconfigurare lo stile delle pagine in modo che questa trasformazione dei

contenuti delle testatine in sole lettere maiuscole sia evitata. Piuttosto, è meglio metterci un comando di interfaccia che vale `\relax` di default, ma può venire ridefinito dall'utente della classe con un comando apposito della classe. Questo in sostanza è quello che fanno le classi configurabili come, per esempio, *memoir* o *scrbook*.

3.2.4 I titoli e la pagina del titolo

Questa sezione si riferisce alle righe da 191 a 276.

Per la classe *book* il titolo del documento viene composto in una pagina a sé stante all'interno dell'ambiente *titlepage* attivato direttamente dal comando `\maketitle`; se però si sceglie l'opzione *notitlepage* allora `\maketitle` riceve una diversa definizione.

Per una nuova classe varrebbe forse la pena di far sì che il comando `\maketitle` eseguisse una composizione di pagine un poco più elegante di quella definita di default; perciò questa è una parte della classe che lascia ampio margine di configurazione o di modifica o di riscrittura ex novo.

Si noti ancora che è quanto mai opportuno, se si vuole comporre una pagina di titolo a sé stante, essere sicuri che sia specificata l'opzione *titlepage* di default o espressamente, perché se è invece in vigore l'opzione *notitlepage* la pagina composta, oltre al titolo, conterrebbe anche l'inizio del testo e quindi dovrebbe essere numerata: infatti, le impostazioni predefinite della classe richiedono che per la pagina che contiene il titolo non isolato venga composta con lo stile `plain`, quindi con il numero di pagina centrato nel piè di pagina.

3.2.5 Gestione dei comandi di sezionamento

Questa sezione si riferisce alle righe da 277 a 433.

Vengono poi impostati il contatore `secnumdepth` per regolare fino a che livello vengono numerati in automatico i titoli e titolini dei comandi di sezionamento; vengono definiti i loro contatori e il modo di scriverne il contenuto.

Vengono definiti i tre principali comandi di sezionamento della classe *book* che non scrivono niente ma in effetti suddividono il documento nelle sue tre parti principali: le pagine iniziali, il corpo del testo e le pagine finali; si tratta dei comandi `\frontmatter`, `\mainmatter` e `\backmatter`. Le

impostazioni predefinite fanno riferimento al modo di scrivere i numeri di pagina (numeri romani nelle pagine iniziali, cifre arabe nelle pagine del corpo del testo e in quelle finali), alla stato della variabile booleana `@mainmatter` che è vera solo nel corpo del testo e falsa nelle due parti estreme; questa variabile booleana controlla anche se i comandi di sezionamento delle parti estreme vengono numerati e messi nell'indice. Per essere più chiari: se si usano i comandi di sezionamento asteriscati, i relativi titoli non vengono numerati e non vengono messi nell'indice indipendentemente dal fatto che `@mainmatter` sia vera o falsa; se però `@mainmatter` è falsa, i comandi *non* asteriscati, che normalmente producono sezioni numerate, non le numerano, ma il loro titolo va ugualmente a finire nell'indice.

Cominciando da `\part` e scendendo via via fino a `\subparagraph` vengono poi definiti i modi di scrivere titoli e titolini, in pagine a sé stanti per le parti, in testa alla pagina per i capitoli, nel corpo del testo ma in evidenza con adeguate spaziature prima e dopo da `\section` a `\paragraph`, fino a `\subparagraph` il cui titolo è in linea con il testo a cui si riferisce.

Si noti che le spaziature prima e dopo i titoli, i corpi dei font da usare, nonché i loro stili sono specificati all'interno dei vari comandi; questo implica l'uso di pacchetti di configurazione per modificare il comportamento dei comandi di sezionamento. Quando si produce una nuova classe questo è un terreno fertile per introdurre comandi di interfaccia, predefiniti magari con gli stessi valori che essi avrebbero nella classe *book*, ma modificabili con comandi accessibili all'utente.

Si prenda in considerazione un'altra possibilità; come è noto, i comandi di sezionamento possono avere un argomento facoltativo e un argomento obbligatorio, per esempio:

`\section[`⟨*Titolo breve*⟩`]{`⟨*Titolo lungo*⟩`}`

Se l'argomento facoltativo non viene specificato, esso viene reso uguale all'argomento obbligatorio; poi questo argomento facoltativo viene scritto nel file `.toc` per la costruzione dell'indice generale, e l'argomento obbligatorio viene usato per comporre per davvero il titolo nel documento e possibilmente anche nella testatina. Quindi il comando facoltativo è un argomento mobile, ed eventuali macro che vi fossero incluse andrebbero protette (ma questo lo deve fare l'utente quando usa questi comandi). Non è detto che lo stesso testo vada bene sia per l'indice sia per la testatina; per ovviare con la classe *book* standard bisogna agire così:

`\section[`⟨*Titolo breve*⟩`]{`⟨*Titolo lungo*⟩`}`
`\markright{`⟨*Titolo per la testatina*⟩`}`

Allora tra le modifiche della nuova classe si potrebbe definire un comando `\section` che accetta due argomenti facoltativi:

`\section[`⟨*Titolo breve*⟩`][`⟨*Titolo per la testatina*⟩`]{`⟨*Titolo lungo*⟩`}`

dove il secondo argomento facoltativo può venire usato solo se si è usato anche il primo. Per fare ciò non basta ricorrere ai pacchetti di personalizzazione, ma bisogna scrivere direttamente comandi di definizione ricorrendo a comandi di basso livello, e curando di definire comunque comandi robusti; in questo aiuta molto il pacchetto *etoolbox* che dispone di due comandi `\robustify` e `\protecting` per rendere robusti comandi che potrebbero essere fragili; quel pacchetto permette anche di definire direttamente comandi protetti, cioè robusti, e di verificare se determinati comandi lo siano. Si può anche sbirciare nel codice della classe *memoir* dove i comandi di sezionamento con due argomenti facoltativi sono già definiti.

3.2.6 Le liste

Questa sezione si riferisce alle righe da 434 a 533.

Proseguendo l'analisi della classe *book* si arriva alle impostazioni per le liste e i loro contatori; si definiscono i vari rientri dipendenti dal livello di annidamento di una lista dentro un'altra; per i contatori si definiscono i modi di stampare il loro valore, sia in generale, sia nella lista, sia nei richiami di elementi di una lista. Per esempio, il contatore LaTeX `enumii`[2] viene scritto con il comando `\theenumii`, ma nella lista viene scritto come definito dal comando `\labelenumii`, mentre in un riferimento incrociato viene scritto premettendogli un prefisso definito con `\p@enumii`; quindi, proseguendo

[2] Il comando LaTeX `\newcounter` definisce un nome di contatore, ma l'algoritmo di allocazione attribuisce quel nome ad una control sequence ottenuta premettendo al nome la stringa `c@`: il nome del contatore `enumii` viene usato per identificare il contatore `\c@enumii`. LaTeX fornisce i meccanismi per agire sui contatori attraverso il loro nomi, ma per usare i contatori LaTeX bisogna sempre agire attraverso comandi che usano il nome per accedere al contatore; tipico è il comando `\value` con il quale si accede al contenuto del contatore; con il linguaggio primitivo di TeX queste cose sono molto più semplici. Ci si ricordi anche che le assegnazioni ai contatori LaTeX sono sempre assegnazioni globali, che prescindono dai limiti dei gruppi.

l'esempio, se il contatore `enumi` contiene il valore 3 e il contatore `enumii` contiene il valore 2, allora `\theenumii` produce b; nella lista compare l'etichetta prodotta da `\labelenumii` (b); in una citazione del secondo elemento della lista annidata nel terzo elemento della prima lista compare 3(b). Come si vede il meccanismo è piuttosto complesso, ma è funzionale.

Queste definizioni riguardano solo l'ambiente *enumerate* definito nel nucleo di LaTeX, come anche l'ambiente *itemize*; le altre liste *description*, *verse*, *quotation* e *quote*[3] sono definite subito dopo. Si noti che la lista *description* consente di definire facilmente un diverso modo di scrivere l'etichetta mediante `\descriptionlabel`; non ci si stupisca troppo per la presenza iniziale di uno spostamento orizzontale pari a `\labelsep`; questo fatto apparentemente incomprensibile dipende dal modo particolare con cui l'ambiente *list* compone l'etichetta. Come è tradizione in questo genere di comandi LaTeX, le caratteristiche stilistiche di scrittura dell'etichetta (font, serie, forma, corpo) sono definite nei comandi stessi, ma sarebbe più interessante se ci fosse un comando di interfaccia e un comando per l'utente per modificare il modo di scrivere l'etichetta. Tuttavia in questo caso ridefinire l'intero comando `\descriptionlabel` non è così complicato, ma la soluzione precedente sarebbe decisamente preferibile; quella soluzione, in effetti, è quella praticata dal pacchetto *enumitem*, dalla classe *memoir* e dalle classi della collezione Koma-Script.

In questa sezione si ridefinisce anche l'ambiente *titlepage* che logicamente sarebbe stato più agevole da trattare se fosse stato adiacente alla definizione di `\maketitle`.

Anche il comando `\appendix` è definito qui. Sarebbe stato più facile da gestire se fosse stato collocato più vicino alla definizione di `\chapter`.

3.2.7 Tabelle e matrici, figure e tavole

Questa sezione si riferisce alle righe da 534 a 584.

[3] Questi ultimi ambienti non sono delle vere liste, ma sono definiti in termini del comando/ambiente `\list` che è utilissimo per gestire ogni minimo dettaglio compositivo delle liste, anche "finte", ma che gli utenti non usano quasi mai; la documentazione dell'ambiente *list* riportata in (Beccari, 2011a) descrive ogni parametro di configurazione di *list*.

La classe prosegue definendo i parametri stilistici per la composizione di tabulazioni, tabelle e matrici, i cui ambienti sono definiti nel nucleo di LaTeX.

Un po' disordinatamente qui vengono impostati i parametri stilistici per le scatole riquadrate `\fbox` e `\framebox`; viene definito il contatore delle equazioni, che viene azzerato ad ogni nuovo capitolo; necessariamente viene quindi ridefinito anche il modo di scrivere questo contatore. Lo stesso viene fatto per i contatori `figure` e `table` dei rispettivi ambienti, che vengono definiti subito dopo.

Logicamente viene qui definito il comando `\caption` e il modo di comporlo; è il posto giusto, ma manca una definizione di `\caption*` per una didascalia non numerata. E manca la possibilità di gestire la posizione della didascalia, sopra o sotto la figura o tabella. Anche questo è un terreno fertile per modifiche e aggiunte.

3.2.8 I vecchi comandi per i font

Questa sezione si riferisce alle righe da 585 a 593.

Le nove definizioni dei comandi del vecchio LaTeX 209 potrebbero venire completamente cancellate da una nuova classe che non avrebbe problemi di compatibilità con i vecchi file scritti prima del 1994. Ci sono alcuni utenti che desiderano ancora usare questi comandi, ma non è opportuno usarli; sia nel testo sia in matematica si comportano diversamente da come lo fanno i corrispondenti comandi LaTeX 2_ε.

Si potrebbero lasciare modificandoli in modo che diventino sostanzialmente equivalenti alle corrispondenti dichiarazioni togliendo il comando `\normalfont` che compare nelle definizioni. Ci sono ragioni a favore e contro una simile modifica, quindi sta al creatore della nuova classe decidere come fare. Io preferirei eliminare quelle definizioni lasciando che durante la compilazione sia LaTeX stesso a dire con un messaggio di errore che quei comandi non sono definiti.

3.2.9 Indici ed elenchi di oggetti mobili

Questa sezione si riferisce alle righe da 594 a 671.

Vengono poi definiti i parametri dimensionali e stilistici dell'indice generale e degli elenchi delle figure e delle tabelle. In questa parte viene anche

impostato il contatore `tocdepth` che regola quali comandi di sezionamento compaiono nell'indice.

Qui viene definito il comando `\tableofcontents`, ma vengono definiti anche i comandi `\l@part`, `\l@chapter`, eccetera, fino a `\l@subparagraph`, per descrivere il modo di comporre ogni voce dell'indice, differenziata nello stile a seconda del comando di sezionamento che l'ha prodotta. Corrispondentemente, vengono definiti i comandi `\listoffigures` e `\listoftables` e il modo di comporre le rispettive voci con i comandi `\l@figure` e `\l@table`. Attenzione: `\l@table` viene reso equivalente mediante `\let` al comando `\l@figure`; per cui, se quest'ultimo venisse ridefinito in seguito, il primo manterrebbe la definizione precedente a meno che non si stabilisca una nuova equivalenza.

Bisogna ancora fare attenzione che i comandi per gli elenchi delle figure e delle tabelle fanno uso del comando `\chapter*`, quindi i loro titoli non sono numerati ma non vanno nemmeno a finire nell'indice. Non solo, ma viene usato di nuovo il comando `\MakeUppercase` per le testatine. Visto che questi elenchi insieme all'indice generale devono essere composti solo nella parte iniziale o solo nella parte finale del documento, sarebbe molto desiderabile che si usasse il comando `\chapter` ma inserendo questi comandi nella *front matter* oppure nella *back matter*, cosicché non verrebbero numerati ma andrebbero direttamente nell'indice generale. Piuttosto si definiscono dei nuovi comandi del tipo `\listoffigures*` in cui viene eliminata la possibilità di numerare e di inserire la voce nell'indice. Sono cose da pensare per la nuova classe e sarebbe utile se si mantenesse una certa omogeneità con altre classi, come *memoir*, dove i comandi asteriscati esistono già.

3.2.10 La bibliografia e l'indice analitico

Questa sezione si riferisce alle righe da 672 a 713.

Vengono definiti gli ambienti *thebibliography* e *theindex*; anche qui, come nell'indice generale viene usato il comando `\chapter*` e per le testatine viene di nuovo usato il comando `\MakeUppercase`; sono gli stessi problemi che si hanno con gli elenchi delle tabelle e delle figure e vanno affrontati e risolti nello stesso modo.

Si badi bene a non confondere lo stile di composizione della bibliografia e dell'indice analitico con lo stile della bibliografia e dell'indice analitico;

lo stile di composizione dipende dalla classe e stabilisce la giustezza delle righe, i font, gli spazi, eccetera; la classe gestisce l'aspetto tipografico. Lo stile della bibliografia dipende da quali informazioni vi vengono inserite, se le voci sono numerate o etichettate in qualche modo o sono prive di etichetta, eccetera. Gli ambienti *thebibliography* e *theindex* sono riempiti con le informazioni che ci mettono dentro i programmi *esterni* BibTeX e makeindex e queste informazioni dipendono rispettivamente dai file di stile `.bst` o `.ist`; se si usa il pacchetto *biblatex* e magari il motore di elaborazione della bibliografia biber, il modo di procedere è diverso e si affida ad un interfaccia a parole chiave, ma il concetto sottostante è sostanzialmente lo stesso: lo stile di composizione dipende da che cosa il programma di elaborazione della bibliografia ha scritto dentro l'ambiente *thebibliography*. Per la bibliografia è l'utente stesso che, con il comando `\bibliographystyle` o con le opzioni per *biblatex*, sceglie quale stile bibliografico usare, mentre con makeindex questo stile può venire scelto solo dando a mano nel terminale o prompt dei comandi il comando letterale con tutte le opzioni necessarie, compreso il nome del file `.ist`, a meno che...

A meno che nella nuova classe non si richieda di default l'uso del pacchetto *imakeidx* (Beccari e Gregorio, 2010) che, con classi basate sul modello di *book*, consente all'utente di specificare il titolo dell'indice analitico, il file di stile con cui comporlo, il motore di elaborazione per ordinare l'indice analitico alfabeticamente e strutturalmente; infine *imakeidx* consente di comporre uno o più indici analitici in modo sincrono con l'esecuzione di pdfLaTeX o di XeLaTeX. Anche per un solo indice del tutto ordinario, con il titolo di default e lo stile `.ist` di default, questo pacchetto è comodissimo perché consente di avere sempre l'indice analitico coerente con il documento nel quale compare.

3.2.11 Definizioni finali e di default

Questa sezione si riferisce alle righe da 714 a 752.

La classe *book* termina con alcune impostazioni stilistiche per le note, per assoggettare il contatore delle note a quello dei capitoli, cosicché le note al piede ricominciano da 1 ad ogni nuovo capitolo.

Vengono eseguite le definizioni di default in inglese (la lingua di default se non si carica *babel* o *polyglossia*); tutti questi "nomi" vengono automaticamente cambiati ogni volta che si cambia lingua ricorrendo ai comandi

definiti da *babel* e *polyglossia*; ma questi, come i font del documento, è bene che siano scelti dall'utente nel suo preambolo.

La classe finisce senza impostare la dichiarazione `\frontmatter`. Forse è giusto che sia così, ma ritengo che la cosa sia da specificare bene nella documentazione. Io preferirei impostare `\frontmatter` di default, obbligando quindi l'utente ad accorgersi che è obbligato a usare anche `\mainmatter` e `\backmatter`, mentre se deve essere lui a pensare di cominciare con `\frontmatter` non prenderà mai l'abitudine e non comporrà mai in modo corretto; poi se vuole può anche ridefinirsi il comando `\frontmatter` come preferisce; per esempio, ci sono utenti che desiderano avere una numerazione in cifre arabe delle pagine senza differenze fra pagine iniziali, centrali o finali e senza ricominciare da 1 alla parte centrale. Per essere sicuri che le impostazioni siano eseguite all'inizio del documento, e non siano prese in considerazione le impostazioni che via via l'utente ha definito nel preambolo, si può usare il comando `\AtBeginDocument`, il cui argomento (costruito cumulativamente ad ogni impiego di questo comando) contenga cose che vengono eseguite solo durante l'esecuzione dell'istruzione `\begin{document}`.

A questo proposito, i comandi `\AtBeginDocument` e `\AtEndDocument` sono definiti nel nucleo di LaTeX; il pacchetto *etoolbox* ne aggiunge diversi altri che permettono di aggiungere l'esecuzione di comandi all'inizio, a metà o alla fine di `\begin{document}` e all'inizio, a metà o alla fine di `\end{document}`. Anzi, la lettura di Lehman (2011b) permette di capire meglio le operazioni delegate all'apertura e alla chiusura di quell'ambiente particolare che si chiama *document*.

3.3 Verifica del funzionamento di una nuova classe

Questa parte, in un certo senso, è più difficile che non scrivere una nuova classe a partire da un modello.

Bisogna cioè costruire un file di prova `.tex` che faccia uso della classe da verificare e che usi ogni comando, ogni impostazione, ogni ambiente, ogni lista, ogni oggetto flottante vecchio e nuovo, tutti i comandi di sezionamento che la classe contenga.

Risulta più facile se questo file viene costruito via via che si crea la classe in modo da inserire subito quei blocchi del file sorgente che permettano di eseguire la verifica di quel che è già stato implementato. Durante le

verifiche non ci si dimentichi di leggere con attenzione il file `.log`; spesso vi si nascondono insidie che non appaiono dal risultato della composizione del file di prova.

Per non dover scervellarsi a trovare testi fasulli per riempire di testo diverse pagine del file di prova, esiste anche il pacchetto ***lipsum*** per generare del testo in *latinorum*; un testo che non significa niente composto da parole messe insieme con pezzi di parole latine. Se un testo così assurdo desse fastidio, si può usare il pacchetto ***kantlipsum***, il cui scopo è lo stesso di ***lipsum***, ma il cui contenuto contiene dei brani in linguaggio pseudo-kantiano e in inglese.

Classe per comporre lettere personalizzate

4

Nel forum GUIT compaiono spesso domande relative alla personalizzazione di classi per comporre lettere. Spesso la personalizzazione implica un lavoro delicato perché una lettera è comunque un oggetto molto personale, sia per le parte grafica dell'intestazione sia per lo stile della prosa usata.

In questo capitolo si mostra come costruire una classe per scrivere lettere; una classe minimale, ma pur sempre capace di essere molto configurabile ed estensibile; ed è quindi adatta a molti usi; tuttavia il suo uso principale è proprio quello di mostrare come fare.

Attenzione; il codice non è banale; i pacchetti usati sono pochi, ma secondo chi scrive, sono molto utili propio per aiutare a creare un file di classe. Si notino i pacchetti usati: *etoolbox*, sempre consigliabile per scrivere codice; *graphicx*, usato non solo per servirsi del suo comando `\includegraphics`, ma anche di altri comandi meno noti; *picture*, che estende le funzionalità dell'ambiente nativo di LaTeX, *picture*, per fare semplici disegni, ma qui usato per collocare oggetti in posizioni le cui coordinate sono parametrizzate alle dimensioni della carta e della gabbia di stampa; *geometry*, per definire la geometria delle pagine della lettera.

In ogni caso spesso verrà usato linguaggio nativo di TeX; non necessariamente perché migliore o più efficiente di quello di LaTeX, ma perché certe funzionalità non sono direttamente disponibili in LaTeX.

Il risultato si può vedere nella pagina 50.

È importante sottolineare l'importanza di leggere il file `.dtx` riportato a partire dalla pagina 57; qui ci si limita a commentare alcune piccole cose che sembrano un po' meno comuni.

Giovanna Bianchi
Via Garibaldi, 35
61029 URBINO (PU)

E p. c. Vincenzo Verdi
Via Gradenigo, 215
11100 GENOVA

Rivoli, 18 marzo 2015

Oggetto: La guida tematica sugli allineamenti verticali

Cara prof. Bianchi,
le scrivo per chiederle la cortesia di inviarmi per posta elettronica il programma della sua nuova guida sugli allineamenti verticali, indicando nel contempo una previsione sui tempi di realizzazione.

Le ricordo che salvo rarissime eccezioni, tutte le guide tematiche non superano le 100 pagine.

I frequentatori del Forum del GUIT sono ansiosi di vedere al luce della nuova guida che compre un'area raramente trattata nelle guide generaliste, ma che riveste una notevole importanza tipografica.

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental

Mario Rossi
Via Giacomo Matteotti, 40
10098 RIVOLI (TO)

Telefono: (+39) 011 9520000
E-mail: mario.rossi@polimi.it
Web: www.guitex.org

unity of apperception can not take account of the discipline of natural reason, by means of analytic unity. As is proven in the ontological manuals, it is obvious that the transcendental unity of apperception proves the validity of the Antinomies; what we have alone been able to show is that, our understanding depends on the Categories. It remains a mystery why the Ideal stands in need of reason. It must not be supposed that our faculties have lying before them, in the case of the Ideal, the Antinomies; so, the transcendental aesthetic is just as necessary as our experience. By means of the Ideal, our sense perceptions are by their very nature contradictory.

Con i miei saluti più cordiali

Mario Rossi
Curatore delle Guide Tematiche
del GUIT

Allegati: Guida 2015-15.pdf

4.1 Commento alla classe

Come al solito le righe dalla 1 alla 41 sono dedicate all'estrazione automatica del codice dal file `.dtx`.

Le righe dalla 43 alla 111 contengono la parte che serve per stampare la documentazione; questa parte è insolitamente lunga, perché ci sono le definizioni di molti comandi utili per scrivere codice documentato; sono comandi che l'autore usa solitamente per scrivere codice all'interno del testo descrittivo; l'autore trova molto comodi questi comandi, ma naturalmente si tratta di gusti personali. Non tutte quelle macro vengono usate nella parte descrittiva, ma l'insieme di quei comandi viene tramandato tale e quale da un documento all'altro; non si è ritenuto opportuno di farne un pacchetto personale, perché questo non sarebbe disponibile quando si inviano i file `.dtx` all'archivio CTAN.

Dalla riga 116 fino alla riga 350 c'è la parte relativa allo scopo e all'uso della classe, con la descrizione dei comandi disponibili per inserire le informazioni necessarie alla composizione della lettera e per personalizzare alcune stringhe. Vi sono esplicitati alcuni consigli; nonostante l'estrema importanza di questa parte per l'utente finale, questa parte non contiene informazioni particolari.

Dalla riga 354 alla fine del file c'è la descrizione del codice e del perché certi comandi sono stati fatti proprio in quel modo. Se l'utente vuole modificare qualche parte, copia il codice sorgente della parte da modificare, lo inserisce in un file con estensione `.sty` o con estensione `.tex` e poi lo può modificare come desidera, sempre che sappia come farlo e che sappia quello che sta facendo; con queste parole non si mettono in dubbio le competenze di chi vuole apportare delle modifiche, ma si sottolinea il fatto che le modifiche non consistono nel semplice uso di comandi già predisposti.

Poi l'utente immetterà questi file personali mediante il file di configurazione, nell'ipotesi che la maggior parte di tali modifiche servano solo per modificare lettere specifiche. Una piccola attenzione: se questi file vengono letti mediante il file di configurazione, allora quelli con estensione `.sty` vanno letti ricorrendo a `\RequirePackage`; invece quelli con estensione .tex vanno letti mediante `\input`. Se le modifiche alla classe dovessero essere permanenti, sarebbe meglio che l'utente copiasse l'intera classe, e la salvasse con un nome diverso, dopo di che può apportare tutte le modifiche che vuole a questa sua versione con il nome cambiato.

Qui non verrano commentate tutte le righe del codice. Il file `lettera.cls`, in totale, tolti i commenti iniziali e quelli finali, ammonta a solo 140 righe.

Si noterà per altro che viene usata la classe *article* come classe di base, su cui costruire la classe *lettera*. Con il pacchetto *geometry* si stabiliscono i margini parametrizzati alla dimensioni della carta, non mediante misure assolute. In questo modo la lettera può essere composta su carta di qualunque formato fra quelli accettabili dalla classe *article*. La classe è stata collaudata con le opzioni *a4paper*, *b5paper*, *a5paper*, *letterpaper*, *legalpaer*, e *executivepaper* constatando che l'aspetto della lettera ovviamente cambia, ma la composizione avviene in modo armonico con tutti e sei i formati. Con lo specifico testo della lettera mostrata nella pagina 50, si nota che con *legalpaper* la firma cade da sola nella seconda pagina; questo non è un caso insolito: lo si cura semplicemente cambiando il testo, oppure inserendo un comando `\newpage` prima dei saluti, o con qualche altro espediente; certo mettere la firma da sola nell'ultima pagina è poco elegante e non è accettabile e va evitata nel modo più assoluto, ma non lo si può ottenere in modo automatico. Ecco un altro motivo per il quale una classe per scrivere lettere deve sempre venire usata con cautela e con molta attenzione.

Il pacchetto *babel* è caricato solo se si usa come programma di composizione pdflatex; se si usasse per esempio xelatex non sarebbe preimpostato nulla in merito alle lingue e l'utente dovrebbe provvedere autonomamente a caricare *polyglossia* e a specificare la lingua da usare. Invece, con pdflatex, avendo precaricato *babel* con l'opzione *italian*, questo pacchetto non può più essere caricato nuovamente dall'utente, ma se vuole usare una lingua diversa dall'italiano, dovrebbe inserirne l'opzione corrispondente fra le opzioni della classe *lettera*.

Dalla riga 437 alla riga 477 sono definiti i comandi per impostare le stringhe con i nomi, gli indirizzi e ogni altra informazione che possa servire per comporre una lettera; vengono anche definiti i comandi per ridefinire le parole fisse.

Subito dopo, se esiste, viene letto il file di configurazione in modo che tutti questi comandi possono venire ridefiniti mediante quanto contenuto nel file di configurazione.

Successivamente, dalla riga 499, alla riga 546 vengono definiti i due stili di pagina, quello per la prima o unica pagina della lettera e quello per le pagine successive. Si noti l'uso del comando `\smash`: serve per "schiacciare" le scatole che contengono i vari oggetti in modo che le loro

altezze e profondità appaiano nulle; lo stesso si ottiene con i due parametri facoltativi del comando `\raisebox`; entrambi sono comandi LaTeX che però vengono usati raramente dall'utente comune.

Per lo stile della prima pagina si noti come è stato creato il "piedino" con le informazioni riguardanti il mittente. si tratta di tue tabelle, di una sola colonna ciascuna, contenute dentro scatole di dimensioni apparenti nulle, al fine di non doversi impegnare con la ripartizione dello spazio fra la tabella di sinistra e quella di destra; infatti, mettendo le tabelle dentro scatole di larghezza nulla, la loro larghezza effettiva viene nascosta; è chiaro che così facendo se le scatole sono troppo larghe, si possono sovrapporre in parte. Non sarebbe una cosa tanto strana; anche con il pacchetto *fancyhdr* sia le testatine sia i piedini sono divisi in tre parti, ciascuna contenuta in scatole di larghezza nulla; se i loro contenuti sono troppo larghi, si sovrappongono.

Nella testatina della prima pagina compare il comando `\foldingmarks`; serve per comporre i segni di piegatura in tre; questo comando agisce solo se la variabile booleana `Aquattro` è vera, perciò i segni di piegatura vengono inseriti solo se si compone su carta ISO A4. Il comando in sé viene definito dal codice compreso fra le righe 557 e 563.

Come si vede esso è un codice di pochissime righe, dove compaiono un po' di "sporchi trucchi", i famosi "dirty tricks" già usati anche da Knuth. Vi compare un ambiente *picture* (nativo di LaTeX) di dimensioni nulle, ma con l'origine spostata in una posizione la cui ordinata è espressa mediante le dimensioni effettive del margine superiore e dell'altezza della testatina; quindi l'origine viene spostata in alto fino al bordo della carta, sulla verticale della gabbia di stampa. I successivi comandi `\put` pongono il loro contenuto in posizioni relative all'origine della figura, spostate in basso rispettivamente di in terzo e di due terzi dell'altezza della pagina, ma spostati indietro al bordo sinistro del foglio. Ecco quindi che le due lineette di piegatura vengono poste sul bordo sinistro dividendolo in tre parti uguali.

Si finisce poi con l'ambiente *lettera*, definito con il codice riportato dalla riga 577 fino alla 597. Non c'è niente di particolare che non sia già descritto nel codice di spiegazione. Si sottolinea il fatto che, sempre grazie al pacchetto *picture* caricato dalla classe, le varie dimensioni per collocare la finestra dell'indirizzo sono in parte assolute in parte relative alla dimensione della pagina e della gabbia di stampa. Questo dipende dal

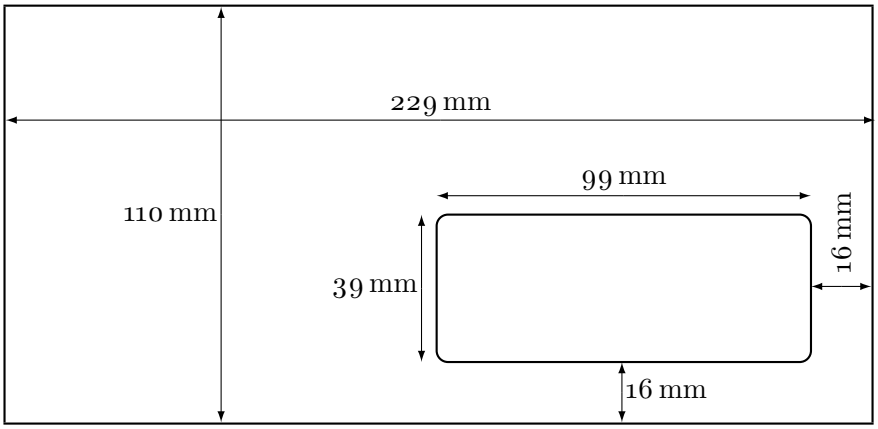

Figura 4.1 Le dimensioni di una busta commerciale

fatto che l'indirizzo del destinatario richiede uno spazio assoluto, visto che è scritto con i font di default e non si può modificare il corpo di questi font in proporzione alle dimensioni della carta. Nello stesso tempo la posizione delle finestra dovrebbe essere in corrispondenza della finestra della busta, la quale ha dimensioni adatte ma non proporzionali alle dimensioni della carta che deve essere introdotta nella busta; in Italia le buste con finestra sono disponibili solo per la carta ISO A4; una busta con finestra regolarmente in commercio ha le dimensioni della finestra di 99 mm per 39 mm, e i suoi bordi sono a 16 mm sia dal bordo inferiore sia dal bordo destro; si veda la figura 4.1.

La posizione della finestra dell'indirizzo sulla prima pagina della lettera è collocata con una certa tolleranza rispetto alla finestra della busta, perché la busta, per ovvi motivi, è più grande del foglio ISO A4 piegato in tre, che sarebbe di 210 mm per 99 mm. In commercio esistono anche buste per fogli ISO A4 leggermente più grandi di quelle indicate nella figura 4.1; a maggior ragione una certa tolleranza è d'obbligo. Questa tolleranza si realizza essenzialmente con delle dimensioni minori sulla prima pagina della lettera, rispetto alla finestra sulla busta.

La classe termina con le due righe 599 e 600, dove per la stile delle pagine si specifica come predefinito quello per le pagine successive alla prima: per la prima viene specificato il suo stile come valido per una sola

pagina, ma se ne ritarda la specificazione alla fine del preambolo, cioè alla fine dell'insieme di comandi eseguiti dall'istruzione `\begin{document}`. Si noti che LaTeX, da solo, provvede solo il comando `\AtBeginDocument`, mentre il pacchetto *etoolbox* fornisce fornisce altri comandi che ritardano l'esecuzione dei comandi contenuti nel loro argomento in punti diversi lungo la serie di istruzioni che vengono eseguite da `\begin{document}`

Finalmente il codice per generare un modello di file di configurazione è riportato nelle righe 669-677. Questo file di configurazione è particolarmente semplice, ma con le istruzioni indicate sia un questo testo sia nella documentazione della classe non dovrebbe essere difficile costruire file di configurazione più complessi, che contengano anche comandi `\RequirePackage` e/o `\input` per immettere ridefinizioni di alcune parti di codice; si raccomanda di ritardare queste ridefinizioni alla specificazione di inizio del documento, mediante il comando `\AtBeginDocument`.

Va da sé che il codice, qui presentato sotto forma di file `.dtx` riprodotto nelle pagine di questa guide, può essere selezionato e copiato pagina dopo pagina dal file PDF, e incollato in un file da salvare con il nome `lettera.dtx`; lo si può elaborare con pdflatex e si ottengono contemporaneamente la documentazione ben composta e i file `lettera.cls` e `lettera.cfg`. Se si pensa di volerli usare per scrivere le proprie lettere, è opportuno spostare il file PDF ottenuto nella cartella `$Home/texmf/doc/latex/lettera/` e gli altri due nella cartella `$HOME/texmf/tex/latex/lettera/`; `$HOME` rappresenta la radice del proprio albero personale per il sistema TeX; in Windows, potrebbe essere qualcosa come `C:\Users\`⟨*nome utente*⟩`\`. in Linux potrebbe essere semplicemente `~/`; in Mac OS X potrebbe essere `~/Library/`. Inoltre se l'albero non esiste o se le cartelle indicate non esistono bisogna crearle con gli strumenti propri di ogni sistema operativo; con MiKTeX è necessario anche creare o aggiornare il database dei nomi dei file relativo all'albero personale; con TeX Live non è necessario.

Un esempio d'uso della classe è riportato nella pagina 50 composto con il codice seguente.

```
% !TEX TS-program = pdflatex
% !TEX encoding = UTF-8 Unicode


\documentclass[12pt,a4paper]{lettera}
\usepackage[T1]{fontenc}
```

```
\usepackage{lmodern}
\usepackage[utf8]{inputenc}
\usepackage{guit}

\usepackage{kantlipsum}% Per generare testo di riempimento.


\mittente{Mario Rossi}
\indirizzomittente{Via Giacomo Matteotti, 40\\
10098 RIVOLI (TO)}
\telefono{(+39) 011 9520000}
\email{mario.rossi@polimi.it}
\web{www.guitex.org}
\destinatario{Giovanna Bianchi}
\indirizzodestinatario{Via Garibaldi, 35\newline
61029 URBINO (PU)}
\logo{../LogoGuITtondo}
\ragionesociale{\textsf{Gruppo utenti Italiani di \TeX\ e \LaTeX}}
\allegati{Guida 2015-15.pdf}
\Noggetto{\textbf{Oggetto}:}
\Nallegati{\textbf{Allegati}:}
\oggetto{La guida tematica sugli allineamenti verticali}
\data{\mbox{}\hfill Rivoli, 18 marzo 2015}
\perconoscenza{Vincenzo Verdi\\ Via Gradenigo, 215\\
11100 GENOVA}


\begin{document}
\begin{lettera}


\noindent Cara prof.~Bianchi,\\
le scrivo per chiederle la cortesia di inviarmi per posta elettronica
il programma della sua nuova guida sugli allineamenti verticali,
indicando nel contempo una previsione sui tempi di realizzazione.

Le ricordo che salvo rarissime eccezioni, tutte le guide tematiche
non superano le 100 pagine.

I frequentatori del Forum del \GuIT\ sono ansiosi di vedere al luce
della nuova guida che compre un'area raramente trattata nelle guide
```

```
generaliste, ma che riveste una notevole importanza tipografica.

\kant[1-2]% due capoversi di riempimento

Con i miei saluti più cordiali
\begin{flushright}
\begin{minipage}{0.5\textwidth}
\vspace*{2\baselineskip}
\centering
Mario Rossi\\
Curatore delle Guide Tematiche\\
del \GuIT
\end{minipage}
\end{flushright}

\end{lettera}
\end{document}
```

4.2 Il file lettera.dtx

```
%\iffalse
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdflatex
%<*system>
\begingroup
\input docstrip.tex
\keepsilent
\preamble
  ----------------------------------------
  The lettera class for typesetting a letter
  Copyright (C) 2015 A. Utore
  All rights reserved

  License information appended

\endpreamble
\postamble
Copyright 2015 A. Utore

        Distributable under the LaTeX Project Public License,
        version 1.3c or higher (your choice). The latest version of
        this license is at: http://www.latex-project.org/lppl.txt

        This Work has the status of 'maintained'

        The Current Maintainer is A. Utore

```

```
        This work consists of this file lettera.dtx, and the derived
        files lettera.cls, lettera.cfg, and lettera.pdf.

\endpostamble
\askforoverwritefalse
%
\generate{\file{lettera.cls}{\from{lettera.dtx}{class}}}
\generate{\file{lettera.cfg}{\from{lettera.dtx}{cfg}}}
%
\def\tmpa{plain}
\ifx\tmpa\fmtname\endgroup\expandafter\bye\fi
\endgroup
%</system>
% \fi
%
% \iffalse
%<*driver>
\ProvidesFile{lettera.dtx}
%</driver>
%<class>\NeedsTeXFormat{LaTeX2e}[2009/01/01]
%<class>\ProvidesClass{lettera}%
%<*class>
   [2015/03/18 v.1.0 Classe per la composizione di lettere di diverse pagine]
%</class>
%<*driver>
\documentclass[a4paper]{ltxdoc}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[italian]{babel}
\usepackage{multicol,pict2e}
\usepackage{guit}% carica anche xcolor
\hfuzz 10pt
%
    \makeatletter
    \newbox\mysintassi
    \newenvironment{sintassi}{\begin{flushleft}\fboxsep=6pt
       \begin{lrbox}{\mysintassi}%
       \begin{minipage}{\dimexpr\linewidth-2\fboxsep-2\fboxrule}\obeylines
    }{%
       \end{minipage}\end{lrbox}%
       \fbox{\box\mysintassi}\end{flushleft}%
    }
    \providecommand*\meta[1]{$\langle${\normalfont\textit{#1}}$\rangle$}
    \providecommand*\marg[1]{\texttt{\char123}\meta{#1}\texttt{\char125}}
    \providecommand*\oarg[1]{\texttt{[}\meta{#1}\texttt{]}}
    \def\GT@splitargs#1,#2!{\def\@tempA{#1}\def\@tempB{#2}}
    \newcommand\garg[1]{\GT@splitargs#1!%
       \texttt{(}\meta{\@tempA}\texttt{,}\meta{\@tempB}\texttt{)}}
    \newcommand*\comando[1]{\texttt{\string#1}}
    \renewcommand*{\cs}[1]{%
```

```
        \texttt{\char92#1}\index{#1@\texttt{\char92#1}}}
    \let\csindex\cs
    \newcommand*\Sambiente[2]{%
        \cs{begin}\marg{#1}\oarg{#2} \dots\ \cs{end}\marg{#1}}
    \newcommand*\Dambiente[3]{%
        \cs{begin}\marg{#1}\oarg{#2}\oarg{#3} \dots\ \cs{end}\marg{#1}}
    \newcommand*\Bambiente[1]{\cs{begin}\texttt{\char123#1\char125}}
    \newcommand*\Eambiente[1]{\cs{end}\texttt{\char123#1\char125}}
    \newcommand*\Arg[1]{\texttt{\char123\relax#1\char125}}
    \newcommand*\prog[1]{\textsf{#1}}
    \newcommand*\pack[1]{\textsf{\slshape#1}}
    \newcommand*\amb[1]{\textit{#1}}
    \newcommand*\class[1]{{\fontshape{ui}\selectfont#1}}
    \newcommand*\file[1]{\texttt{#1}}
    \newcommand*\opz[1]{\textsl{#1}}
\makeatother
%
\setactivedoublequote
%
\begin{document}\errorcontextlines=9
\GetFileInfo{lettera.dtx}
\date{Versione numero: \fileversion; ultima revisione: \filedate}
\title{La classe \texttt{\filename}}
\author{%
A. Utore\thanks{e-mail \texttt{augusto dot utore at server dot domain}}}
\maketitle
\columnseprule0.5pt
\begin{multicols}{2}
\tableofcontents
\end{multicols}
\DocInput{lettera.dtx}
\end{document}
%</driver>
% \fi
%
% \CheckSum{384}
%
% \begin{abstract}
% Questa classe mostra come creare una classe per scrivere lettere nelle
% quali la prima pagina sia dotata di una vistosa testata (accademica,
% aziendale, professionale, o altro) e contenga le coordinate del mittente
% in calce. Nelle pagina successive la testata sarà molto più sobria,
% mancheranno le coordinate del mittente, ma il piedino conterrà centrato il
% numero progressivo di pagina. Le altre informazioni come l'indirizzo del
% destinatario, l'oggetto, le formule di apertura e di chiusura compaiono
% normalmente. La lettera viene scritta in italiano; nulla vieta di
% scrivere il corpo della lettera in un'altra lingua, ma la specificazione
% della lingua va fatta mediante le opzioni della classe.
% \end{abstract}
%
%\section{Il formato della lettera}
```

```
% La lettera verrà stampata su un foglio di carta di formato qualsiasi, vale
% a dire uno di quelli accettabili dalla classe |article|, cioè
% |a4paper|, |a5paper|, |b5paper|, |letterpaper|, |legalpaper| e
% |executivepaper|; se non si specifica nessuna opzione per il formato
% della carta, sarà usato il formato preimpostato (durante l'installazione
% del sistema \TeX) probabilmente il formato ISO A4, ma senza specificare
% l'opzione del formato della carta in certi casi potrebbero manifestarsi
% degli errori di composizione.
%
% La prima e spesso unica pagina della lettera conterrà una testata molto
% visibile, con il logo e la denominazione sociale dell'ente a cui appartiene
% il mittente, sia esso un ente accademico, scientifico, professionale, aziendale
% o altro. Nel piedino di questa prima pagina apparirà il nome e gli altri
% dati del mittente, quali l'indirizzo, il telefono, il fax, l'indirizzo di
% posta elettronica, il sito web. Ognuno di questi dati può mancare (tranne
% il nome del mittente) e, nel caso, non viene stampato nulla al suo posto.
%
% La prima pagina conterrà anche i segni di piegatura in tre al fine di poter
% inserire la lettera piegata dentro alla corrispondente busta rettangolare;
% la cosa ha senso solo per il formato A4, almeno in Italia, probabilmente
% anche con il formato ``letter'' per gli USA; tuttavia con questa classe
% i segni di piegatura sono visibili solo quando si usa la carta~A4.
%
% I margini laterali e verticali sono abbastanza grandi per non far sembrare
% la lettera un dépliant pubblicitario denso di informazioni; essi sono
% personalizzabili usando le apposite istruzioni per il pacchetto
% |geometry| inserite nel file di configurazione specifico della particolare
% lettera. Questo file, infatti, ha il nome |\jobname.cfg| ed è presumibile
% che ogni lettera abbia un main file con un nome sufficientemente mnemonico
% da essere differente da qualunque altra lettera, almeno di quelle che
% risiedono nella stessa cartella. Le pagine successive alla prima contengono
% nella testatina solo il logo e il nome% dell'ente in corpo normale e nel
% piedino solo il numero della pagina.
%
%\section{Uso della classe}
% La classe si carica nel solito modo:
%\begin{sintassi}
%\cs{documentclass}\oarg{opzioni}\Arg{lettera}
%\end{sintassi}
% Bisogna poi specificare le codifiche di ingresso e dei font, nonché i font
% da usare, le informazioni relative al destinatario e ogni altra informazione
% relativa alla lettera, prima di comporre la lettera vera e propria.
%\begin{sintassi}
%\cs{usepackage}\oarg{codifica dei font}\Arg{fontenc}
%\cs{usepackage}\oarg{opzioni dei font}\Arg{font}
%\cs{usepackage}\oarg{codifica}\Arg{inputenc}
%\meta{informazioni relative al destinatario, all'oggetto,
% agli allegati, ai destinatari per conoscenza, eccetera.}
%\Bambiente{document}
%\end{sintassi}
%
```

```
% Si noti quanto segue.
%\begin{enumerate}
% \item
% Nel preambolo non è indicato l'uso del pacchetto |babel|; esso infatti
% è già caricato dalla classe con l'opzione \meta{italian}; non può quindi
% essere caricato nuovamente nel preambolo con opzioni diverse, nel caso che
% si voglia comporre la lettera in un'altra lingua. Quest'altra lingua, però
% può essere indicata fra le \meta{opzioni} della classe.
% \item
% La \meta{codifica} di ingresso deve essere specificata, ma si consiglia
% vivamente di usare la codifica |utf8| che non pone nessun problema con
% i caratteri accentati e con moltissimi caratteri analfabetici che potrebbero
% venire usati nel testo della lettera.
% \item
% Per l'italiano e per moltissime lingue europee l'uso degli accenti e di altri
% segni diacritici è essenziale; solo per l'inglese non sarebbe necessario
% disporre di font con i caratteri pre"accentati; tuttavia anche scrivendo in
% inglese può essere necessario scrivere nomi stranieri che fanno uso di segni
% diacritici, per cui la \meta{codifica dei font} |T1| è sempre consigliabile.
% Se non si specificano le \meta{opzioni dei font} nel caricare un pacchetto
% per l'uso di font non preimpostati, con lo stesso comando si possono caricare
% diversi pacchetti relativi ai \meta{font}, in particolare potrebbe essere
% conveniente, e quasi sempre lo è, specificare anche il pacchetto
% \pack{textcomp} per disporre di un altro centinaio di caratteri testuali
% analfabetici.
% \item
% I dati relativi al mittente è opportuno che siano contenuti nel file di
% configurazione |\jobname.cfg| insieme alle altre informazioni che lo
% riguardano e ad altre eventuali impostazioni per usare lingue diverse
% dall'italiano.
% Questo stesso file \file{.dtx} genera un file \file{lettera.cfg} che si può
% usare come modello per creare il proprio file di configurazione.
% Invece nel preambolo si devono mettere tutte le informazioni relative
% al destinatario e tutte le ulteriori informazioni necessarie per la
% completezza delle lettera.
%\end{enumerate}
%
% Per questo motivo la maggior parte della classe è dedicata alla definizione
% dei comandi per inserire le informazioni necessarie fisse e variabili della
% lettera che si vuole comporre.
%
% \section{Le informazioni per l'intestazione}
% I comandi per fornire tutti i dati alla classe per disporre di tutte
% le informazioni necessarie sono i seguenti.
% \begin{description}\let\originalitem\item
% \def\item[#1]{\originalitem[\normalfont\cs{#1}]}
%\item[mittente] accetta in argomento col nominativo del mittente.
%\item[indirizzomittente] accetta un argomento con l'indirizzo del mittente;
% se si svolge su più righe, bisogna inserire necessariamente i comandi
% \cs{\char92} alla fine di ogni riga.
%\item[telefono] accetta come argomento il numero di telefono del mittente
```

```
% completo del prefisso internazionale, per esempio \texttt{(+39) 011 987 65 43},
% la separazione die gruppi di cifre è utile ma non necessaria.
%\item[Ntelefono] accetta come argomento una parola diversa da ``Telefono:'' da
% premettere al numero di telefono; contiene anche la punteggiatura necessaria;
% è utile specialmente per cambiare font o per scrivere la parola in una lingua
% diversa dall'italiano; valgono le stesse considerazioni per i prossimi comandi
% che cominciano con una 'N' maiuscola.
%\item[fax] accetta come argomento il numero del fax del mittente.
%\item[Nfax] serve per definire un nome diverso da ``Fax:'' da premettere al
% numero del fax.
%\item[email] accetta come argomento l'indirizzo e-mail del mittente.
%\item[Nemail] serve per definire un nome diverso da ``E-mail:'' da premettere
% all'indirizzo e-mail.
%\item[web] accetta come argomento l'url del mittente o dell'ente  cui il
% mittente appartiene.
% \item[Nweb] serve per definire un nome diverso da ``Web:'' da premettere
% all'url del mittente.
% \item[destinatario] accetta come argomento il nominativo del destinatario.
%\item[indirizzodestinatario] accetta come argomento l'indirizzo completo
% del destinatario con le varie righe già divise con il comando \cs{\char92}.
%\item[allegati] accetta come argomento la lista degli allegati in una forma
% qualsiasi; un solo allegato, una lista  con i nomi separati da virgole
% seguite da uno spazio; con una costruzione completa come un intero ambiente
% tabular con i nomi messi uno per riga, eccetera.
%\item[Nallegati] serve per definire un nome diverso da ``Allegati:'' da
% premettere alla lista degli allegati.
%\item[oggetto] accetta come argomento la frase che descrive l'oggetto
% della lettera.
%\item[Noggetto] definisce una parola diversa da  ``Oggetto:'' da premettere
% alla frase che descrive l'oggetto.
%\item[perconoscenza] accetta come argomento la lista completa dei nominativi
% e degli indirizzi, separati convenientemente con i comandi \cs{\char92}
% dei destinatari a cui la lettera viene inviata per conoscenza.
%\item[Nperconoscenza] definisce una stringa diversa da ``E p.c.'' da
% premettere alla lista dei destinatari per conoscenza.
%\item[logo] accetta come argomento il nome del file che contiene il logo
% del mittente, sia esso una sua sigla o il logo dell'ente a cui
% il mittente appartiene.
%\item[ragionesociale] accetta come argomento il ``nome'' dell'ente a cui
% appartiene il mittente o una stringa a scelta del mittente.
%\item[data] pre"impostata alla data del giorno in cui si compone la lettera,
% serve per definire una data fissa, possibilmente preceduta dal luogo dove
% si trova il mittente al momento di scrivere la lettera; potrebbe quindi
% essere meglio usare questo comando, per esempio, con un argomento così
% fatto:\\[1.5ex]
%\cs{data}\Arg{Roma, 1 aprile 2015}
% \end{description}
% Ognuno di questi nomi o di queste liste può essere omesso; tutti gli
% argomenti che non rappresentino semplicemente delle parole fisse, hanno
% un valore predefinito ``vuoto'' e, se appunto il loro valore è vuoto,
% non viene scritto da nessuna parte, senza nemmeno la parola fissa che
```

```
% lo dovrebbe precedere. Solo la data è pre"impostata con la data fornita
% dal comando \cs{today} di \LaTeX.
%
% Ognuno di questi argomenti e di queste parole fisse può essere completata
% con comandi di formattazione; per esempio se si volesse ridefinire la
% stringa che precede l'oggetto, la si potrebbe ridefinire tranquillamente
% così:
%\begin{verbatim}
%\Noggetto{\textbf{Oggetto} --}
%\end{verbatim}
%
% \section{Lo stile delle pagine}
% Come già detto, l'intestazione della prima pagina della lettera richiede
% uno stile di pagina particolare che metta il logo e la ragione sociale
% dell'ente dello scrivente come intestazione della prima pagina.
% Le pagine successive conterranno solamente il logo con la ragione sociale
% scritti con un font normale. La classe deve quindi provvedere a due stili
% di pagina.
%
% Ma sorge spontanea una domanda: va bene per le lettere commerciali o ufficiali,
% me se uno desidera confezionarsi una carta intestata personale? La risposta
% è semplice, nel senso che il logo non è obbligatorio e la ragione sociale non
% è obbligatoria. Tuttavia la carta intestata personale potrebbe contenere un
% monogramma dello scrivente; oppure potrebbe contenere solo il nome e il
% cognome scritto con un font adeguato. Siccome tutto è configurabile, almeno
% entro certi limiti, si può fare quello che si vuole.
%
% L'utente di questa classe non deve preoccuparsi di come funzionino questi due
% stili di pagina a meno che non voglia personalizzare l'aspetto di queste pagine.
%
% Merita allora ricordare che  le pagine oltre la prima sono sostanzialmente
% una modesta variante dello stile standard |myheadings|, consistente,
% sostanzialmente nello spostamento del numero di pagina al centro del
% piedino; per impostazione predefinita, il logo (o le cifre dello scrivente)
% viene esposto in formato ridotto nella testatina insieme, se c'è, alla
% ragione sociale o al nome del mittente.
%
% Invece lo stile della prima pagina è molto più complesso; l'intestazione
% contiene il logo (o le cifre dello scrivente) sporgente nel margine sinistro.
% La testatina vera e proprio può avere un nome di ragione sociale scritto con
% carattere piuttosto grandi; il tutto però viene fatto apparire al programma
% di composizione come se avesse altezza e profondità nulle.
%
% Il piedino può essere composto di molte righe e sporge nel margine inferiore
% quanto basta, ma non è garantito che il tutto si mantenga dentro lo spazio
% fisico della carta.
%
% Una particolarità è che alcune parti del disegno della pagina sono assolute;
% per esempio i segni per le piegature in tre sono riferiti alle dimensioni
% fisiche della carta, non alla larghezza dei margini comunque definiti, o
% dalla classe o dalle impostazioni dell'utente. Questi segni di piegatura
```

```
% potrebbero essere inseriti per qualunque formato di carta specificato alla
% classe, ma si ritiene che per una classe da usare essenzialmente in Italia,
% sia ragionevole usarli solo per carta in formato ISO A4; non avrebbe senso
% piegare in tre fogli decisamente più piccoli; avrebbe senso con i formati
% USA, ma poi in Italia non si trovano buste di formati adatti.
%
% \section{Considerazioni finali}
%
% Va notato che fra le classi standard esiste la classe \class{letter} che
% ha le sue funzionalità e uno stile molto angloamericano. La classe
% \class{letteracpd} è italiana ed è molto configurabile; la documentazione
% si trova nel file \file{cpdbundle.pdf}. La classe \class{scrlettr2} è
% tedesca e fa parte del gruppo di classi KOMA Script; anche questa classe
% è configurabile e la sua documentazione si trova assieme a quella delle
% altre classi della stessa collezione in \file{scrguide.pdf} in tedesco e
% \file{scrguien.pdf} in inglese. Probabilmente esistono altre classi meno note
% e di uso più ristretto.
%
% \StopEventually{}
%
% \section{Descrizione del codice}
% \subsection{Preliminari}
% L'intestazione della classe è già stata fatta. Ora si tratta di procedere
% alla chiamata di \class{article}. Non ci sono opzioni specifiche per questa
% classe; tutte quelle che le vengono specificate vengono trasferite
% integralmente alla classe \class{article}. Tuttavia merita ricordare che,
% anche se il formato di carta ISO A4 è il formato predefinito per
% l'installazione del sistema \TeX, l'opzione \opz{a4paper} va specificata
% lo stesso, perché imposta la variabile booleana che permette di inserire
% i segni di piegatura in tre.
%
%\iffalse
%<*class>
%\fi
% Definiamo uno switch per verificare se si sta componendo su carta UNI A4:
%    \begin{macrocode}
\newif\ifAquattro \Aquattrofalse
\DeclareOption{a4paper}{\Aquattrotrue\paperwidth=210mm\paperheight=297mm}
%    \end{macrocode}
%    \begin{macrocode}
%\DeclareOption{*}{\PassOptionToClass{\CurrentOption}{article}}
\ProcessOptions*\relax
\LoadClassWithOptions{article}
%    \end{macrocode}
%
%\subsection{Test e impostazioni}
% Prima di leggere i dati e il file di configurazione è bene configurare bene
% la geometria della pagina e gli stili di pagina. Ci vuole la garanzia che
% lavorando con \prog{pdflatex} vengano trasmessi correttamente i dati al
% file PDF.
%    \begin{macrocode}
```

```
\RequirePackage{iftex}
\ifPDFTeX
\expandafter\@firstoftwo\else
\expandafter\@secondoftwo\fi
{%
  \pdfpagewidth=\paperwidth
  \pdfpageheight=\paperheight
  \RequirePackage[italian]{babel}
}{%
  \relax
}
%    \end{macrocode}
%
% Impostiamo i margini:
%    \begin{macrocode}
\RequirePackage{geometry}
\geometry{%
   top=0.1\paperheight,
   bottom=0.15\paperheight,
   left=0.2\paperwidth,
   right=0.1\paperwidth,
   heightrounded
   }
%    \end{macrocode}
%
%\subsection{Comandi per i dati}
% Si possono definire le varie stringhe e i vari comandi per acquisire i dati
% del mittente, del destinatario e le altre informazioni, impostando tutti
% i valori di default con il valore ``vuoto''. Gli indirizzi devono essere
% specificati per disteso inserendo i comandi |\\| alla fine di ogni riga.
%
% Per i destinatari a cui inviare la lettera per conoscenza bisogna specificare
% la lista completa, compreso il loro nome, non solo l'indirizzo, nell'argomento
% di un unico comando, inserendo tutti i comandi |\\| necessari, non solo alla
% fine di ogni riga come per il mittente e il destinatario principale; inoltre,
% se si vuole si più inserire a mano uno spazio verticale aggiuntivo fra il
% blocco di righe di ciascun destinatario per conoscenza; non sono accettabili
% righe vuote nell'argomento del comando |\perconoscenza|.
%
% Il comando |\logo| serve per acquisire il nome del file che contiene il
% disegno del logo; non occorre specificarne l'estensione se non ne esistono
% due versioni con lo stesso nome ma con estensione diversa. Si ricorda
% inoltre che le versioni moderne di \prog{pdflatex} accettano il
% file"-immagine nei formati PDF, PNG, JPG, MPS, EPS; altri formati devono
% essere trasformati preventivamente in uno di questi e si consiglia di
% preferire  i formati genuinamente vettoriali PDF, MPS, EPS; se non se ne
% dispone, non si commetta ``l'errore'' di incorporare un file bitmapped
% in un file con uno dei tre formati suddetti, perché il file ottenuto non
% sarebbe \emph{genuinamente} vettoriale, ma solo \emph{apparentemente}
% vettoriale. Piuttosto, se si dispone solo di un file bitmapped in un
% formato non consentito, lo si converta nel formato PNF.
```

```
%    \begin{macrocode}
\let\@mittente\empty
\newcommand*\mittente[1]{\gdef\@mittente{#1}}
\let\@indirizzomittente\empty
\newcommand*\indirizzomittente[1]{\gdef\@indirizzomittente{#1}}
\let\@telefono\empty
\newcommand*\telefono[1]{\gdef\@telefono{#1}}
\gdef\N@telefono{Telefono: }
\newcommand*\Ntelefono[1]{\gdef\N@telefono{#1\space}}
\let\@fax\empty
\newcommand*\fax[1]{\def\@fax{#1}}
\gdef\N@fax{Fax: }\newcommand*\Nfax[1]{\gdef\N@fax{#1\space}}
\let\@email\empty
\newcommand*\email[1]{\def\@email{#1}}
\gdef\N@email{E-mail: }\newcommand*\Nemail[1]{\gdef\N@email{#1\space}}
\let\@web\empty
\newcommand*\web[1]{\def\@web{#1}}
\gdef\N@web{Web: }\newcommand*\Nweb[1]{\gdef\N@web{#1\space}}
%
\let\@destinatario\empty
\newcommand*\destinatario[1]{\gdef\@destinatario{#1}}
\let\@indirizzodestinatario\empty
\newcommand*\indirizzodestinatario[1]{\gdef\@indirizzodestinatario{#1}}
\let\@allegati\empty
\newcommand*\allegati[1]{\gdef\@allegati{#1}}
\gdef\N@allegati{Allegati: }
\newcommand*\Nallegati[1]{\gdef\N@allegati{#1\space}}
\let\@oggetto\empty
\newcommand*\oggetto[1]{\gdef\@oggetto{#1}}
\gdef\N@oggetto{Oggetto: }
\newcommand*\Noggetto[1]{\gdef\N@oggetto{#1\space}}
\let\@perconoscenza\empty
\newcommand*\perconoscenza[1]{\gdef\@perconoscenza{#1}}
\gdef\N@perconoscenza{E p.~c. }
\newcommand*\Nperconoscenza[1]{\gdef\N@perconoscenza{#1\space}}
\gdef\@data{\today}
\newcommand*\data[1]{\gdef\@data{#1}}
%
\let\@logopic\empty
\newcommand*\logo[1]{\gdef\@logopic{#1}}
\let\@ente\empty
\newcommand*\ragionesociale[1]{\gdef\@ente{#1}}
%    \end{macrocode}
%
%\subsection{File di configurazione}
% Ora si può caricare il file |\jobname.cfg| con il quale l'utente può
% specificare sia le dimensioni della carta tramite le opzioni al comando
% |\geometry| sia le dimensioni dei margini (naturalmente deve sapere quello
% che sta facendo; si ricorda che sarebbe quanto mai opportuno che i dati del
% mittente vengano introdotti mediante il file di configurazione. Tutte le
% specifiche devono essere indicate mediante i comandi sopra definiti.
```

```
%    \begin{macrocode}
\IfFileExists{\jobname.cfg}%
{\input{\jobname.cfg}}%
{\ClassWarning{lettera}{Il file \jobname.cfg non e' presente nella
cartella del file principale}}
%    \end{macrocode}
%
%\subsection{stili delle pagine}
% Si può ora procedere alla definizione degli stili di pagina. Nelle pagine
% successive alla prima  si può fare riferimento a  anche ai mark ordinariamente
% caricati com |\markboth|, mark che però sono inizialmente impostati vuoti.
%    \begin{macrocode}
\markboth{}{}
\RequirePackage{etoolbox,graphicx}
%
\def\ps@altrepagine{%
  \def\@oddhead{\makebox[\textwidth][l]{%
  \ifcsvoid{@logopic}{\ifcsvoid{@ente}{\rightmark}{\@ente}}{%
  \raisebox{-0.3\height}[\z@][\z@]{%
  \includegraphics[height=2\headheight]{\@logopic}}\qquad \@ente}}}
  \let\@evehead\@oddhead
  \def\@oddfoot{\makebox[\textwidth]{\thepage}}
  \let\@evenfoot\@oddfoot
}
%    \end{macrocode}
%
% Per la prima pagina le cose sono più elaborate.
%    \begin{macrocode}
\newbox\intestazione
\def\ps@primapagina{%
\def\@oddhead{%
  \llap{\ifcsvoid{@logopic}{}{%
    \raisebox{-0.45\height}[\z@][\z@]{%
    \includegraphics[width=\dimexpr(\oddsidemargin+1in)/2]{\@logopic}}%
    \quad\ifAquattro\foldingmarks\fi}}%
    \setbox\intestazione\hbox{\ifcsvoid{@ente}{}{\smash{\Huge\@ente}}}
    \ifdim\wd\intestazione>\textwidth
      \setbox\intestazione
         \hbox{\resizebox{\textwidth}{!}{\box\intestazione}}%
    \fi
  \makebox[\textwidth][l]{\box\intestazione}%
}
\def\@evehead{}
\def\@oddfoot{\smash{\makebox[\textwidth]{\sffamily\footnotesize
    \makebox[\z@][l]{\let\\ = \cr
        \tabular[t]{@{}l@{}}
        \@mittente\\
        \@indirizzomittente
        \endtabular
    }\hfill
    \makebox[\z@][r]{\tabular[t]{@{}r@{}}
```

```
        \ifcsvoid{@telefono}{}{\N@telefono\@telefono\cr }
        \ifcsvoid{@fax}{}{\N@fax\@fax\cr }
        \ifcsvoid{@email}{}{\N@email\@email\cr }
        \ifcsvoid{@web}{}{\N@web\@web\cr }
        \endtabular
    }}
}}
\def\@evenfoot{}
}
%    \end{macrocode}
% Il comando |\foldingmarks| va definito per mettere i segni per la piegatura;
% si usa l'ambiente \amb{picture} esteso con il pacchetto \pack{picture} di Heiko
% Oberdiek, che consente di usare direttamente le dimensioni effettive e non
% solamente il moltiplicatore di |\unitlength|, come fanno tutti i comandi
% dell'ambiente nativo di \LaTeX. Probabilmente non abbiamo bisogno di usare
% \pack{pict2e}, ma lo carichiamo lo stesso perché deve essere caricato prima
% di \pack{picture}; se l'utente lo ricarica si spera che lo faccia senza
% opzioni, ma ci pensa |\usepackage| a non ricaricare un pacchetto già caricato.
%    \begin{macrocode}
\RequirePackage{pict2e,picture}
\newcommand*\foldingmarks{%
  \begin{picture}(0,0)(0,-\topmargin-1in-\headheight)
  \put(-\oddsidemargin-1in,-\paperheight/3){\line(1,0){10mm}}
  \put(-\oddsidemargin-1in,-\paperheight*2/3){\line(1,0){10mm}}
  \end{picture}%
}
%    \end{macrocode}
%
%\subsection{Ambiente \amb{lettera}}
% Ora bisogna definire l'ambiente che contiene il corpo della lettera; solo
% la posizione dell'indirizzo del destinatario deve essere posto in una
% posizione assoluta. Ecco perché dopo l'intestazione bisogna mettere la
% scatola con l'indirizzo del destinatario. Alcune dimensioni per mettere la
% scatola che contiene l'indirizzo del destinatario sono espresse in millimetri,
% invece che parametrizzate alle dimensioni della carta; la scatola dovrebbe
% essere collocata in posizioni parametrizzate, ma l'indirizzo del destinatario
% richiede comunque uno spazio indipendente dal formato della carta. bisogna
% quindi fare un misto, ma la cosa non dovrebbe essere particolarmente grave.
%    \begin{macrocode}
\newenvironment{lettera}{% Apertura
\noindent
\begin{picture}(0,0)(0,-\topmargin-1in-\headheight-\headsep-\topskip)
\put(\textwidth-80mm,-65mm){\makebox(99mm,0)[cc]{%
\parbox[b]{80mm}{\@destinatario\newline
\@indirizzodestinatario}}}
\end{picture}\par\vspace{65mm}
%
\ifcsvoid{@perconoscenza}{}{%
\ifcsvoid{N@perconoscenza}{}{\noindent\N@perconoscenza}
\parbox[t]{0.67\textwidth}{\@perconoscenza}\par\vspace*{\baselineskip}}
%
```

```
\noindent\@data\par
%
\ifcsvoid{@oggetto}{}{%
\ifcsvoid{N@oggetto}{}{\vspace*{\baselineskip}%
\noindent\N@oggetto\@oggetto\par\vspace*{\baselineskip}}}%
}{% Chiusura
\ifcsvoid{@allegati}{}{\ifcsvoid{N@allegati}{}{%
\vspace*{\baselineskip}\noindent\N@allegati\@allegati}\par}
}

\pagestyle{altrepagine}
\AtBeginDocument{\thispagestyle{primapagina}}
%    \end{macrocode}
% Il comando |\allegati| può anche essere usato dentro l'ambiente \amb{lettera};
% serve solo per impostare la lista degli allegati, poi ci pensa la classe
% a stampare la lista al posto giusto come ultima cosa della lettera.
%
% \subsection{Altri comandi}
% Perché questa classe non contiene altri comandi come |\signature|,
% |\opening|, |\closing|? Perché questi testi devono essere scritti di
% volta in volta in modo possibilmente meno burocratico di come si fa
% di solito e come viene fatto nelle classi menzionate nella sezione
% ``Considerazioni finali''.
%
% Sta quindi all'utente di gestire l'aspetto tipografico e il contenuto;
% per esempio non ci vorrebbe nulla per definire un comando |\signature|
% per comporre la firma con gli attributi del firmatario, cioè il mittente,
% in modo corretto.
%\begin{verbatim}
%\begin{flushright}
%  \begin{minipage}{0.5\textwidth}\centering
%  \vspace*{2\baselineskip}% spazio per la firma
%     Ambrogio Utore\\
%     Direttore Generale
%  \end{minipage}
%\end{flushright}
%\end{verbatim}
% Questo intero blocco di codice può costituire l'argomento di un altro
% comando da sistemare nel file di configurazione, chiamato per esempio
% \cs{firmaqualificata}, che può essere usato dall'utente per sistemare
% la firma con la qualifica del firmatario dove vuole lui.
% Anzi, se dispone di una firma scandita, resa parzialmente trasparente
% (con il solo sfondo trasparente), ne può mettere il relativo comando
% \cs{includegraphics} completo della larghezza adatta al tipo di firma,
% al posto della spaziatura di |2\basleineskip|, cosicché quando mette la
% \cs{firmaqualificata} vi appare anche la firma grafica.
%
% Cose analoghe valgono per altre informazioni come per esempio il fatto
% che la lettera sia una raccomandata; talvolta questevengono scritte, ma
% chi le scrive prima dell'''Oggetto'', chi dopo; chi allineate a sinistra
% chi a destra. Meglio fare tutto a mano piuttosto che definire comandi
```

```
% troppo complessi e fragili.
%\iffalse
%</class>
%\fi
% \section{Il file di configurazione}
% Il file di configurazione serve per diversi scopi, il primo dei quali
% consiste nel alleviare il possessore della classe per scrivere le sue
% stesse lettere ripetendo ogni vola le stesse informazioni che lo riguardano.
% Siccome bisogna predisporre un diverso file di configurazione per ogni
% lettera che si scrive, è possibile inserire nel file di configurazione
% ogni informazione che potrebbe essere collocata anche nel preambolo, come
% per esempio i dati del destinatario , gli allegati, l'oggetto della missiva,
% l'elenco dei destinatari per conoscenza.
%
% È possibile importare file nel file di configurazione, per cui certe
% informazioni come quelle dello scrivente possono risiedere in un file
% immesso con |\input| nel file di configurazione.
%
% Volendo, l'utente può anche modificare la configurazione e lo stile delle
% pagine; naturalmente chi fa queste cose sa come fare e sa dove mettere
% le mani; non è una operazione banale. Anche queste modifiche o diversi
% set di modifiche possono essere salvati in file distinti immessi nel file
% di configurazione mediante comandi |\input|.
%
% Qui si mostra un file di configurazione di base.
%\iffalse
%<*cfg>
%\fi
%    \begin{macrocode}
\mittente{Augusto Utore}
\indirizzomittente{Via Principale, 123\\
99099 TeXOPOLI, Italia}
\telefono{(+39)123\,9876543}
\fax{(+39)123\,9876542}
\email{a.utore@myserver.com}
\web{}% conferma che non c'e' sito web personale
\Noggetto{Oggetto:\space}
\endinput
%    \end{macrocode}
% Con un file di configurazione così essenziale è evidente che tutte le altre
% informazioni necessarie vanno inserite nel preambolo della lettera.
%\iffalse
%</cfg>
%\fi
% Questo è tutto.
%
% \Finale
% \endinput
```

La classe per comporre questa guida tematica 5

Per fare un esempio più esteso riporto qui il file `.dtx` completo per la descrizione della classe con cui ho composto questo testo[1]; seguono poi la classe stessa e il file di documentazione debitamente composto.

Grazie a questa classe, questo stesso documento comincia semplicemente con un (quasi) preambolo vuoto, così:

```
\documentclass[11pt,b5paper,twoside,openany,ipertesto]{guidatematica}
\ProvidesFile{IntroCreaClassi.tex}[2013/01/30 v.1.8
  Introduzione alla creazione di file di classe]

\DeclareRobustCommand*{\Ars}{% Definizione tratta da guit.sty
  \textsf{\lower -.48ex\hbox{\rotatebox{-20}{A}}\kern -.3em{rs}}%
  \kern -.05em\TeX\kern -.17em\lower -.357ex\hbox{nica}}

\usepackage{pdfpages}

\begin{document}

\frontmatter
\GetFileInfo{IntroCreaClassi.tex}
\author{Claudio Beccari}
\title{Introduzione alla creazione\\ di file di classe}
\date{\filename\ \fileversion\ del \filedate}
\maketitle

```

[1] Quanto si riporta qui è solo esemplificativo; il file `guidatematica.dtx` e i suoi derivati possono corrispondere ad una versione precedente a quella effettivamente usata per comporre questo testo. Per quel che riguarda la versione eventualmente aggiornata si esamini il kit distribuito ai potenziali autori di guide tematiche `guidatematica.zip`.

```
20 \contribguit
21
22 \chapter{Introduzione}
23 ...
24 \chapter{Ringraziamenti}
25 ...
26 \newpage
27 \tableofcontents
28 ...
```

e nella riga 14 si vede già usato uno dei nuovi comandi definiti nella classe *guidatematica*. Lo scarno preambolo contiene il comando `\ProvidesFile` con i suoi argomenti; la definizione del logo di ArsTEXnica; questo sarebbe contenuto nel pacchetto *guit*, ma come viene spiegato nel file `guidatamatica.pdf`, per motivi di compatibilità con XƎLATEX si è evitato di usare direttamente quel pacchetto. Infine vengono caricati il pacchetto *pdfpages* che è funzionale per questa specifica guida: il pacchetto *hyperref*, grazie all'opzione *iperteso*, va sempre caricato per ultimo e vi provvede la classe stessa rinviando la sua invocazione all'esecuzione del comando `\begin{document}`.

5.1 Il file documentato

Il file `guidatematica.dtx` ora, a differenza degli esempi precedenti, è piuttosto lungo, ma ho voluto mostrare il codice completo proprio per renderlo istruttivo anche per i lettori di questa guida. Esso è documentato, quindi non dovrebbero essere necessarie spiegazioni particolari che non siano già scritte nel file stesso.

```
1 % \iffalse
2 % !TEX encoding = UTF-8 Unicode
3 % !TEX TS-program = pdflatex
4 %<*system>
5 \begingroup
6 \input docstrip.tex
7 \keepsilent
8 \preamble
9   ----------------------------------------
10  The guidatematica class for typesetting books in the GuIT series
11  "Guide Tematiche"
12  Copyright (C) 2012, 2014 GuIT, Gruppo utilizzatori Italiani di TeX e LaTeX
```

```
  All rights reserved

  http://www.guitex.org

  License information appended

\endpreamble
\postamble
Copyright 2012, 2013 GuIT, Gruppo utilizzatori Italiani di TeX e LaTeX

        Distributable under the LaTeX Project Public License,
        version 1.3c or higher (your choice). The latest version of
        this license is at: http://www.latex-project.org/lppl.txt

        This Work has the status of ‘maintained’

        The Current Maintainer is the GuIT staff

        This work consists of this file guidatematica.dtx,
        and the derived files guidatematica.cls and guidatematica.pdf
        plus the associated documentation guidatematica-doc.tex
        and guidatematica-doc.pdf.

\endpostamble
\askforoverwritefalse
%
\generate{\file{guidatematica.cls}{\from{guidatematica.dtx}{class}}}
%
\def\tmpa{plain}
\ifx\tmpa\fmtname\endgroup\expandafter\bye\fi
\endgroup
%</system>
% \fi
%
% \iffalse
%<*driver>
\ProvidesFile{guidatematica.dtx}
%</driver>
%<class>\NeedsTeXFormat{LaTeX2e}[2009/01/01]
%<class>\ProvidesClass{guidatematica}%
%<*class>
   [2014/09/03 v.1.5 Classe simmetrica per comporre testi della collana
   Guide Tematiche del GuIT]
%</class>
%<*driver>
```

```
\documentclass[a4paper]{ltxdoc}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern,textcomp}
\usepackage[italian]{babel}
\hfuzz 10pt
%
\makeatletter
\providecommand*\meta[1]{$\langle$\textit{#1}$\rangle$}
\providecommand*\marg[1]{\texttt{\char123}\meta{#1}\texttt{\char125}}
\providecommand*\oarg[1]{\texttt{[}\meta{#1}\texttt{]}}
\def\GT@splitargs#1,#2!{\def\@tempA{#1}\def\@tempB{#2}}
\newcommand\garg[1]{\GT@splitargs#1!%
   \texttt{(}\meta{\@tempA}\texttt{,}\meta{\@tempB}\texttt{)}}
\newcommand*\comando[1]{\texttt{\string#1}}
\renewcommand*{\cs}[1]{%
   \texttt{\char92#1}\index{#1@\texttt{\char92#1}}}
\let\csindex\cs
\newcommand*\Sambiente[2]{%
   \cs{begin}\marg{#1}\oarg{#2} \dots\ \cs{end}\marg{#1}}
\newcommand*\Dambiente[3]{%
   \cs{begin}\marg{#1}\oarg{#2}\oarg{#3} \dots\ \cs{end}\marg{#1}}
\newcommand*\Bambiente[1]{\cs{begin}\texttt{\char123#1\char125}}
\newcommand*\Eambiente[1]{\cs{end}\texttt{\char123#1\char125}}
\newcommand*\Arg[1]{\texttt{\char123\relax#1\char125}}
%
\begin{document}\errorcontextlines=9
\GetFileInfo{guidatematica.dtx}
\title{La classe \texttt{\filename}\thanks{Version number \fileversion;
 ultima revisione \filedate.}}
\author{GuIT\thanks{Web: \texttt{www.guitex.org}}}
 \maketitle
 \DocInput{guidatematica.dtx}
\end{document}
%</driver>
% \fi
%
%
% \CheckSum{1076}
% \begin{abstract}
% Questa classe \texttt{guidatematica} serve per comporre
% testi da pubblicare nella collana "Guide Tematiche" del GuIT.
% Si appoggia alla classe \texttt{memoir} ma include
% molte personalizzazioni e i pacchetti necessari.
% Differisce dalla classe \texttt{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 pdf\LaTeX,
% Xe\LaTeX\ e Lua\LaTeX,
% \end{abstract}
%
% \section{Caratteristiche della classe}
%   La classe \texttt{guidatematica.cls} serve per comporre
%   libri in italiano con le seguenti specifiche:
% \begin{itemize}
% \item La classe di supporto è la classe
%        \texttt{memoir.cls} facente parte del sistema TeX
%        in qualsiasi distribuzione completa.
% \item Tutte le opzioni inserite nell'istruzione
%        |\documentclass| vengono trasferite integralmente
%        alla classe \texttt{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. 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'autore della guida.
% \item 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.
% \item La classe riconosce da sola il motore di composizione
%        e imposta direttamente quanto occorre nella forma richiesta
%        dal programma di composizione.
% \item La codifica di entrata deve essere \texttt{utf8}; per
%        XeLaTeX essa è implicita; con pdfLaTeX essa viene
%        preimpostata e non è possibile con questa classe specificare
```

```
% una codifica differente.
% \item La codifica dei font di uscita con pdfLaTeX è
%       \texttt{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.
% \item Con pdfLaTeX il pacchetto \texttt{babel} viene caricato
%       con la sola opzione \texttt{italian}; nei rari casi in cui
%       fosse necessario usare anche altre lingue, esse vanno
%       specificate come opzione alla classe, perché non si può
%       chiamare \texttt{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.
% \item 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 ricordassi
%       di specificare una famiglia di font che contenga l'alfabeto
%       desiderato.
% \item Sono stati introdotti molti nuovi comandi e sono stati
%       definiti molti nuovi ambienti. Per maggiori informazioni si
%       rimanda al documento \emph{guidatematica-doc.pdf} distribuito
%       con il ``kit'' |guidatematica.zip|.
%       \end{itemize}
% \StopEventually{}
%
% \section{Codice commentato}
% Le due righe contenti 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 \texttt{memoir}.
% Viene definita l'opzione |ipertesto| per caricare il pacchetto |hypertext|
% che non 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 questo opzioni bisogna ricorrere ad opportuni comandi
% condizionali, per poter ritardare la loro esecuzione ad un momento
% successivo al caricamento della classe |memoir|.
% \iffalse
%<*class>
% \fi
%    \begin{macrocode}
\DeclareOption{ipertesto}{%
\AtBeginDocument{\RequirePackage[colorlinks,linkcolor={blue},
  citecolor={blue!80!black},urlcolor={blue}]{hyperref}}
}

\newif\ifPAD\PADfalse
\DeclareOption{pad}{\PADtrue\tabletfalse\Aivfalse\Bvfalse\Avfalse}
\newif\iftablet\tabletfalse
\DeclareOption{tablet}{\tablettrue\Aivfalse\Bvfalse\Avfalse\PADfalse}
\newif\ifAiv \Aivfalse
\DeclareOption{a4paper}{\Aivtrue\tabletfalse\Bvfalse\Avfalse\PADfalse}
\newif\ifBv \Bvtrue
\DeclareOption{b5paper}{\Aivfalse\Bvtrue\tabletfalse\Avfalse\PADfalse}
\newif\ifAv \Avfalse
\DeclareOption{a5paper}{\Avtrue\Aivfalse\Bvfalse\tabletfalse\PADfalse}

\DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
\ProcessOptions*\relax

\LoadClassWithOptions{memoir}
%    \end{macrocode}
%
% Si carica subito il pacchetto |iftex|  che permette di distinguere
% quale motore di composizione si sta usando fra |pdftex|, |xetex|, oppure |luatex|.
%    \begin{macrocode}
\RequirePackage{iftex}
%    \end{macrocode}
% 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.
%    \begin{macrocode}
\ifPDFTeX
        \RequirePackage[utf8]{inputenc}
        \RequirePackage[T1]{fontenc}
        \RequirePackage[italian]{babel}
        \RequirePackage[tt={oldstyle=false,tabular,monowidth}]{cfr-lm}
\else
        \RequirePackage{fontspec}
        \defaultfontfeatures[\rmfamily,\sffamily]{Ligatures=TeX,Numbers=OldStyle}
        \defaultfontfeatures[\ttfamily,\tvfamily]{Numbers=Lining}
        \setmainfont{Latin Modern Roman}[SmallCapsFont={* Caps}]
        \setsansfont{Latin Modern Sans}
        \setmonofont{Latin Modern Mono}
        \newfontfamily{\tvfamily}{Latin Modern Mono Prop}
        \DeclareTextFontCommand{\texttm}{\ttfamily}
        \DeclareTextFontCommand{\texttv}{\tvfamily}
        \def\textl#1{{\addfontfeature{Numbers=Lining}#1}}
        \RequirePackage{unicode-math}
        \setmathfont{Latin Modern Math}
        \RequirePackage{polyglossia}
        \setmainlanguage{italian}
        \ifx\tmspace\@undefined
          \newlength{\tmspace}\tmspace=\z@
        \fi
\fi
%    \end{macrocode}
%
%
% 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 (immagine pdf) la si sarebbe potuta limitare
% all'uso con |XeLaTeX| o di |Lua\LaTeX|; siccome alla data attuale il pacchetto
% |guit.sty| è sottoposto a revisione per l'uso con i font OpenType, è molto
% meglio usare temporaneamente l'immagine in ogni caso e modificare
% questa classe solo con l'aggiunta di un nuovo pacchetto quando
% |guit.sty| sarà diventato compatibile con tuttii 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.
%    \begin{macrocode}
\RequirePackage{graphicx}
\RequirePackage{metalogo}
\RequirePackage{pict2e}[2009/06/01]
\RequirePackage{microtype}
\RequirePackage{etoolbox}
\RequirePackage{xcolor}
\RequirePackage{natbib}
\RequirePackage{multicol}
\RequirePackage{imakeidx}
\RequirePackage{fancyvrb}
\RequirePackage{afterpage}
\RequirePackage{enumitem}
\RequirePackage[right]{eurosym}
\RequirePackage{listings}
\lstloadlanguages{[LaTeX]TeX}
%    \end{macrocode}
%
% 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 inconvenient 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.
%
%    \begin{macrocode}
\normalfont
\ifPAD\setstocksize{160mm}{120mm}\fi
\iftablet\setstocksize{120mm}{90mm}\fi
```

```
\ifBv\setstocksize{250mm}{176mm}\fi
\ifAiv\setstocksize{293mm}{210mm}\fi
\ifAv\setstocksize{210mm}{146mm}\fi
\settrims{0pt}{0pt}

\settypeblocksize{\paperwidth}{\dimexpr\paperwidth*\paperwidth/\paperheight}{*}
\setlrmargins{*}{*}{*}
\setulmargins{*}{*}{*}
\setheadfoot{\baselineskip}{2\onelineskip}
\setheaderspaces{*}{\onelineskip}{*}
\setmarginnotes{7pt}{\dimexpr\foremargin-3em}{5pt}
\checkandfixthelayout[nearest]
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\def\GetFileInfo#1{%
  \def\filename{#1}%
  \def\@tempb##1 ##2 ##3\relax##4\relax{%
    \def\filedate{##1}%
    \def\fileversion{##2}%
    \def\fileinfo{##3}}%
  \edef\@tempa{\csname ver@#1\endcsname}%
  \expandafter\@tempb\@tempa\relax? ? \relax\relax}
%    \end{macrocode}
%
% 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.
```

```
%    \begin{macrocode}
\let\@copymark\@empty % Di default è vuoto
\newcommand*\setcopymark[1]{\gdef\@copymark{#1}}
\newcommand*\@insertcopymark{%
        \begin{picture}(0,0)\unitlength=1mm\relax
        \if@twoside
                \ifodd\value{page}
            \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
                \else
            \put(-\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
                \fi
        \else
            \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
        \fi
        \end{picture}%
}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\def\cblayoutsinistro{%
\dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
\dimen258=\dimexpr\evensidemargin+1in\relax
\dimen262=1mm\relax
\dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
\dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
     -\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
\put(0,0){\unitlength=\p@
\put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen264){%
    \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
    \Line(0,0)(\strip@pt\paperwidth,\strip@pt\paperheight)%
    \put(\strip@pt\dimexpr\paperwidth/2\relax,%
        \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
```

```
        {\circle{\strip@pt\paperwidth}}%
}
\end{picture}}

\def\cblayoutdestro{%
\dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
\dimen258=\dimexpr\oddsidemargin+1in\relax
\dimen262=1mm\relax
\dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
\dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
    -\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
\put(0,0){\unitlength=\p@
\put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen264){%
    \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
    \Line(0,\strip@pt\paperheight)(\strip@pt\paperwidth,0)%
    \put(\strip@pt\dimexpr\paperwidth/2\relax,%
       \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
       {\circle{\strip@pt\paperwidth}}%
}
\end{picture}}
%    \end{macrocode}
% 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
% può essere utilizzato a inizio o fine documento. Tuttavia,
% per garantire maggiore visibilità, si consiglia di inserire
% |\contribguit| dopo |\frontmatter| e prima del capitolo di
% introduzione o eventuali pagine con informazioni sulle licenze d'uso.
%    \begin{macrocode}
```

```
\providecommand*\Ars{%
  \textsf{\lower -.48ex\hbox{\rotatebox{-20}{A}}\kern -.3em{rs}}\hskip0pt%
  \kern -.05em\TeX\hskip0pt\kern -.17em\lower -.357ex\hbox{nica}}
\newcommand{\contribguit}{\newpage
\thispagestyle{empty}
\noindent{\Large
Associati anche tu al \GuIT
\hfill
\setlength{\unitlength}{1mm}
 \begin{picture}(50,2)
  \setlength{\fboxsep}{0pt}
  \put(1,-3){\colorbox{gray}{\framebox(50,6.5){}}}
  \put(0,-2){\colorbox{white}{\framebox(50,6.5){%
   \href{http://www.guitex.org/home/it/diventa-socio/associarsi-a-guit}{%
    Fai click per associarti}}}}
 \end{picture}%
}
\bigskip

L'associazione per la diffusione di \TeX\ in Italia, riconosciuta
ufficialmente in ambito internazionale, si sostiene \emph{unicamente}
con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile,
il mezzo principale per ringraziare gli autori è diventare socio.
\medskip

Divenendo soci si ricevono gratuitamente:
\begin{itemize}
\item
l'abbonamento alla rivista \Ars;
\item
il DVD \TeX\ Collection;
\item
un eventuale oggetto legato alle attività del \GuIT.
\end{itemize}

L'adesione al \GuIT\ prevede un quota associativa compresa tra \EUR{12,00}
e \EUR{70,00} a seconda della tipologia di adesione prescelta ed ha
validità per l'anno solare in corso.
\newpage
}
%    \end{macrocode}
%
% 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.
%
%    \begin{macrocode}
\nouppercaseheads
%\renewcommand{\footruleheight}{\normalrulethickness}
\renewcommand{\footruleskip}{0pt}
\makeheadrule{headings}{\textwidth}{0pt}
\makeheadrule{myheadings}{\textwidth}{0pt}
\makeevenfoot{plain}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{plain}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenfoot{headings}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{headings}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenfoot{myheadings}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{myheadings}{}{\textsc{\thepage}}{\@insertcopymark}
%    \end{macrocode}
% I mark delle pagine destre e sinistre sono composti centrati
% sia nelle pagine destre sia in quelle sinistre.
%    \begin{macrocode}
\makeevenhead{headings}{}{\textsc{\small\leftmark}}{}
\makeoddhead{headings}{}{\textsc{\small\rightmark}}{}
\makeevenhead{myheadings}{}{\textsc{\small\leftmark}}{}
\makeoddhead{myheadings}{}{\textsc{\small\rightmark}}{}
%    \end{macrocode}
% Vengono definiti i comandi per ``decorare'' le intestazioni delle testatine.
%    \begin{macrocode}
\makepsmarks{headings}{%
\createmark{chapter}{both}{shownumber}{\@chapapp\space}{.\qquad}
\createmark{section}{right}{shownumber}{$\mathsection$\,}{\qquad}
\renewcommand*\indexmark{\markboth{\indexname}{\indexname}}}
%    \end{macrocode}
% Vengono definiti gli stili di pagina modificati con il layout disegnato
% sotto. Non sono da usare sistematicamente, ma probabilmente come
% argomenti di |\thispagestyle|.
%    \begin{macrocode}
\makepagestyle{headingslayout}
```

```
\makeevenhead{headingslayout}{\cblayoutsinistro}{\textsc{\small\leftmark}}{}
\makeevenfoot{headingslayout}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddhead{headingslayout}{\cblayoutdestro}{\textsc{\small\rightmark}}{}
\makeoddfoot{headingslayout}{}{\textsc{\thepage}}{\@insertcopymark}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\makechapterstyle{guidatematica}{%
   \renewcommand*{\chapnumfont}{%
      \fontshape{it}\fontsize{40}{40}\selectfont}
   \renewcommand*{\printchaptername}{}
   \renewcommand*{\chapternamenum}{}
   \renewcommand*{\chaptitlefont}{%
       \fontsize{18}{16}\scshape}% sterlineato
   \renewcommand{\printchapternum}{%
       \noindent\rlap{\makebox[\textwidth][r]{\rlap{\makebox[\foremargin][l]{%
       \chapnumfont \thechapter}}}}\printchaptertitle}
   \renewcommand*{\afterchapternum}{}
}
\AtBeginDocument{\chapterstyle{guidatematica}}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\renewcommand*\partnamefont{\normalfont\large\scshape}
\renewcommand*\partnumfont{\normalfont\large\scshape}
\renewcommand*\parttitlefont{\normalfont\huge\scshape}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\setbeforesecskip{-3.5ex plus-1ex minus-0.2ex}
\setbeforesubsecskip{-3ex plus-1ex minus-0.2ex}
\setbeforesubsubsecskip{-2.5ex plus-1ex minus-0.2ex}
\setbeforeparaskip{1\onelineskip plus1ex minus0.2ex}
```

```
\setbeforesubparaskip{1\onelineskip plus1ex minus0.2ex}
\setaftersecskip{1.5ex plus0.2ex}
\setaftersubsecskip{1.5ex plus0.2ex}
\setaftersubsubsecskip{1.5ex plus0.2ex}
\setafterparaskip{-1em}
\setaftersubparaskip{-1em}
\setsubparaindent{\parindent}

\setsecheadstyle{\large\scshape\raggedright}
\setsubsecheadstyle{\large\scshape\raggedright}
\setsubsubsecheadstyle{\large\scshape\raggedright}
\setparaheadstyle{\small\scshape}
\setsubparaheadstyle{\small\scshape}

\aliaspagestyle{part}{empty}
\aliaspagestyle{chapter}{empty}
\copypagestyle{titlepage}{headings}
\renewcommand\cftpartpagefont{\scshape}
\renewcommand\cftpartfont{\large\scshape}
\renewcommand\cftchapterfont{\large\scshape}
\renewcommand\cftchapterpagefont{\scshape}
\renewcommand*{\cftchapterfillnum}[1]{%
    {\cftchapterleader}\nobreak
    \cftchapterformatpnum{#1}%
    \cftchapterafterpnum\par\nobreak}
\setlength{\cftbeforechapterskip}{1.0em \@plus 2\p@}
%    \end{macrocode}
%
% Grazie alle funzionalità di |memoir| si ridefiniscono anche
% le modalità di comporre le didascalie.
%    \begin{macrocode}
\captiondelim{\quad}
\captionnamefont{\small\scshape}
\captiontitlefont{\small}
\captionstyle[\centering]{}
\hangcaption
\captionwidth{\dimexpr\textwidth-2\parindent\relax}\changecaptionwidth
%    \end{macrocode}
%
% 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 maiuscoletto; se si vuole cambiare font o stile
% basta usare gli appositi comandi nella forma \meta{chiave}
```

```
% = \meta{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.
%    \begin{macrocode}
\renewcommand\descriptionlabel[1]{\hspace\labelsep\normalfont\scshape #1}
\renewcommand\blockdescriptionlabel[1]{\normalfont\scshape #1}
\providecommand\plaindescriptionlabel[1]{\hspace\labelsep\normalfont #1}

\renewlist{blockdescription}{description}{1}
\setlist[blockdescription]{before={\let\makelabel\blockdescriptionlabel},
leftmargin=\z@,labelsep*=0.5em,labelindent=\z@,labelwidth=\z@}

\newlist{plaindescription}{description}{1}
\setlist[plaindescription]{before={\let\makelabel\plaindescriptionlabel}}

\newlist{compactenumerate}{enumerate}{1}
\setlist[compactenumerate,1]{label=\arabic*.,
   noitemsep, partopsep=\z@, topsep=.25\onelineskip}

\newlist{compactitemize}{itemize}{4}
\setlist[compactitemize]{label=•,
   noitemsep,partopsep=\z@,topsep=.25\onelineskip}

\newlist{compactdescription}{description}{1}
\setlist[compactdescription]{%
style=sameline,noitemsep,partopsep=\z@,topsep=.25\onelineskip}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\AtBeginDocument{\setsecnumdepth{subsection}
   \settocdepth{subsection}\maxsecnumdepth{subsection}
   \maxtocdepth{subsection}}
%    \end{macrocode}
%
% Viene ora definita la virgola intelligente; si veda una
% descrizione più dettagliata della sua utilità in matematica
```

```
% per esempio della documentazioone di questa classe
% |gidatematica-doc.pdf|. 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
% \emph{non} è l'inglese. Si fornisce anche un comando
% |\m@thcomma| come cuore della definizione della virgola attiva.
%    \begin{macrocode}
 \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
 \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}
\unless\ifPDFTeX
 \AtBeginDocument{\ifcsstring{xpg@main@language}{english}{\relax}{%
    \mathcode`\,=\string"8000}}
\else
 \AtBeginDocument{\ifcsstring{languagename}{english}{\relax}{%
    \mathcode`\,=\string"8000}}
\fi
{\catcode `,=\active \gdef,{\futurelet\let@token\m@thcomma}}%
%    \end{macrocode}
% 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 analfabetico
% 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
% \fbox{\cs{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.
%    \begin{macrocode}
\AtEndPreamble{%
\gdef\m@thcomma#1{%
\unless\ifcat\noexpand\let@token*%
  \virgola
\else
        \expandafter\expandafter\expandafter
            \ifcsname\expandafter\@gobble\string#1\endcsname
          \virgola
        \else
          \ifnum`#1<`0\relax
            \virgola
          \else
            \ifnum`#1>`9\relax
              \virgola
            \else
              \virgoladecimale
            \fi
          \fi
        \fi
\fi#1}%
}
```

```
%    \end{macrocode}
%
% 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|.
%    \begin{macrocode}
\renewcommand{\bibsection}{\chapter{\bibname}\prebibhook}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\let\imki@idxprologue\empty
\def\imki@columns{2}
\renewenvironment{theindex}
{%
        \clearforchapter
    \csname phantomsection\endcsname
    \chapter{\indexname}%
    \indexmark%
    \parindent\z@
    \parskip\z@ \@plus .3\p@\relax
    \let\item\@idxitem
    \begin{multicols}{\imki@columns}[\imki@idxprologue]
    \raggedright
}
{%
    \end{multicols}\gdef\imki@idxprologue{}\clearpage
}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\AtBeginDocument{\bibliographystyle{guidatematica}%
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\unless\ifPDFTeX
   \ifcsstring{xpg@main@language}{italian}%
     {\csgappto{init@extras@italian}{\nofrench@indent}}{}%
\fi}
%    \end{macrocode}
%
% 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.
%    \begin{macrocode}
\newenvironment{titlepage}{\clearpage
\pagestyle{titlepage}
    \makeoddhead{titlepage}{}{\smash{\Large\scshape\@author}}{}
    \makeoddfoot{titlepage}{}{\smash{\large\scshape\@date}}{}
}{\clearpage}

\providecommand\subtitle[1]{\gdef\@subtitle{#1}}
\let\@subtitle\empty
```

```

\renewcommand\maketitle{
    \begin{titlepage}
    \vspace*{\stretch{.7}}
    {\centering\LARGE\scshape\@title\par}
    \ifdefvoid{\@subtitle}{}{\vspace{2\onelineskip}
    {\centering\scshape\Large\@subtitle\par}}
    \vspace{\stretch{.7}}
    {\centering\includegraphics[width=0.23\paperwidth]{logoguittondo}\par}
    \vspace*{\stretch{.7}}
    \end{titlepage}
}
%    \end{macrocode}
%
% 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.
%
% I 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.
%    \begin{macrocode}
\newenvironment{medaglione}[1][\linewidth]{%
    \begin{lrbox}{0}%
    \begin{minipage}{\dimexpr#1-2\fboxsep-2\fboxrule}
}{%
    \end{minipage}\end{lrbox}\fbox{\usebox{0}}\relax
}

\newenvironment{sintassi}{\flushleft\medaglione}%
```

```
   {\endmedaglione\endflushleft}
%    \end{macrocode}
%
% 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.
%
% {\tolerance=9999\relax
% I comandi sono |\meta| per un \meta{argomento generico};
% |\marg| per un \marg{argomento obbligatorio}; |\oarg| per un
% \oarg{argomento facoltativo}; |\garg| per le
% \garg{coordinate,geometriche}; |\comando| per un
% \comando{\comando} completo del suo backslash; |\cs| per il
% nome (senza backslash) di un \cs{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
% \Sambiente{ambiente}{argomento}; |\Dambiente|
% per descrivere la sintassi di un ambiente che accetta
% un argomento facoltativo per il quale bisogna indicare
% il valore predefinito, come in
% \Dambiente{ambiente}{argomento}{default}; |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 \Arg{123}.\par}
%    \begin{macrocode}
\newcommand*\hz{\nobreak\hskip\z@}
\renewcommand*\meta[1]{$\langle$\textit{\hz#1}$\rangle$}
\renewcommand*\marg[1]{\texttt{\char123}\meta{#1}\texttt{\char125}}
\renewcommand*\oarg[1]{\texttt{[}\meta{#1}\texttt{]}}
\newcommand*\Arg[1]{\texttt{\char123\relax#1\char125}}
\def\GT@splitargs#1,#2!{\def\@tempA{#1}\def\@tempB{#2}}
\newcommand\garg[1]{\GT@splitargs#1!\texttt{(}\meta{\@tempA}%
   \texttt{,}\meta{\@tempB}\texttt{)}}
\newcommand*\comando[1]{\texttt{\string#1}}
\renewcommand*{\cs}[1]{\texttt{\char92#1}\index{#1@\texttt{\char92#1}}}
\let\csindex\cs
```

```
\newcommand*\Sambiente[2]{\cs{begin}\marg{#1}\oarg{#2}\,\dots
   \cs{end}\marg{#1}}
\newcommand*\Dambiente[3]{\cs{begin}\marg{#1}\oarg{#2}\oarg{#3}\,\dots
   \cs{end}\marg{#1}}
\newcommand*\Bambiente[1]{\cs{begin}\texttt{\char123#1\char125}}
\newcommand*\Eambiente[1]{\cs{end}\texttt{\char123#1\char125}}
%    \end{macrocode}
%
% 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
% \emph{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 \emph{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.
%    \begin{macrocode}
%
\DeclareRobustCommand*\ambstyle[1]{\textsf{\slshape#1}}
\DeclareRobustCommand*\classstyle[1]{\texttv{\itshape#1}}
\DeclareRobustCommand*\filestyle[1]{\texttm{\textl{#1}}}
\unless\ifPDFTeX
```

```
    \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape{#1}}}
\else
    \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape\textl{#1}}}
\fi
\DeclareRobustCommand*\progstyle[1]{\textsf{#1}}
\DeclareRobustCommand*\prog[1]{\progstyle{#1}%
   \index{programma!#1@\progstyle{#1}}}
\DeclareRobustCommand*\pack[1]{\packstyle{#1}%
   \index{pacchetto!#1@\packstyle{#1}}}
\DeclareRobustCommand*\class[1]{\classstyle{#1}%
   \index{classe!#1@\classstyle{#1}}}
\DeclareRobustCommand*\file[1]{\filestyle{#1}%
   \index{file!#1@\filestyle{#1}}}
\DeclareRobustCommand*\amb[1]{\ambstyle{#1}%
   \index{ambiente!#1@\ambstyle{#1}}}
%
\DeclareRobustCommand*\opzstyle[1]{\textsl{\textl{#1}}}
\DeclareRobustCommand*\contastyle[1]{\texttm{#1}}
\DeclareRobustCommand*\stilestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\numeristyle[1]{\texttm{#1}}
\DeclareRobustCommand*\umisurastyle[1]{\texttm{#1}}
\DeclareRobustCommand*\chiavestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\descrittorestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\posizionestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\allineamentostyle[1]{\texttm{#1}}
%
\DeclareRobustCommand*\opz[1]{\opzstyle{#1}%
   \index{opzione!#1@\opzstyle{#1}}}
\DeclareRobustCommand*\conta[1]{\contastyle{#1}%
   \index{contatore!#1@\contastyle{#1}}}
\DeclareRobustCommand*\stile[1]{\stilestyle{#1}%
   \index{stile della pagina!#1@\stilestyle{#1}}}
\DeclareRobustCommand*\numeri[1]{\numeristyle{#1}%
   \index{numerazione!#1@\numeristyle{#1}}}
\DeclareRobustCommand*\umisura[1]{\umisurastyle{#1}%
   \index{unit\`a di misura!#1@\umisurastyle{#1}}}
\DeclareRobustCommand*\chiave[1]{\chiavestyle{#1}%
   \index{chiave!#1@\chiavestyle{#1}}}
\DeclareRobustCommand*\descrittore[1]{\descrittorestyle{#1}%
   \index{descrittore di colonna!#1@\descrittorestyle{#1}}}
\DeclareRobustCommand*\posizione[1]{\posizionestyle{#1}%
   \index{posizione degli oggetti flottanti!#1@\posizionestyle{#1}}}
\DeclareRobustCommand*\allineamento[1]{\allineamentostyle{#1}%
   \index{codice di allineamento!#1@\allineamentostyle{#1}}}
%    \end{macrocode}
```

```
%
% Finalmente la macro |\GuIT| per gestire l'immagine del logo del GuIT.
%    \begin{macrocode}
%
\DeclareRobustCommand*\GuIT{{\dimen@=\f@size\p@\dimen2=20.88\p@
\edef\guitscale{\strip@pt\dimexpr\dimen@*\p@/\dimen2\relax}%
\raisebox{-0.93ex}[\z@][\z@]{%
   \includegraphics[scale=\guitscale]{logoguitlineare}}}}
\let\guit\GuIT
%    \end{macrocode}
% \iffalse
%</class>
% \fi
% Questo è tutto.
% \Finale
%
% \endinput
```

5.2 Il file di classe estratto dal file documentato

Il file è autoestraente; l'esecuzione di pdfLaTeX sul file `.dtx` produce quindi il file di classe `guidatematica.cls` e il file di documentazione `guidatematica.pdf`; il primo appare così:

```
%%
%% This is file 'guidatematica.cls',
%% generated with the docstrip utility.
%%
%% The original source files were:
%%
%% guidatematica.dtx  (with options: 'class')
%%   ----------------------------------------
%%   The guidatematica class for typesetting books in the GuIT series
%%   "Guide Tematiche"
%%   Copyright (C) 2012, 2014 GuIT, Gruppo utilizzatori Italiani di TeX e LaTeX
%%   All rights reserved
%%
%%   http://www.guitex.org
%%
%%   License information appended
%%
\NeedsTeXFormat{LaTeX2e}[2009/01/01]
\ProvidesClass{guidatematica}%
   [2014/09/03 v.1.5 Classe simmetrica per comporre testi della collana
```

```
   Guide Tematiche del GuIT]
\DeclareOption{ipertesto}{%
\AtBeginDocument{\RequirePackage[colorlinks,linkcolor={blue},
  citecolor={blue!80!black},urlcolor={blue}]{hyperref}}
}

\newif\ifPAD\PADfalse
\DeclareOption{pad}{\PADtrue\tabletfalse\Aivfalse\Bvfalse\Avfalse}
\newif\iftablet\tabletfalse
\DeclareOption{tablet}{\tablettrue\Aivfalse\Bvfalse\Avfalse\PADfalse}
\newif\ifAiv \Aivfalse
\DeclareOption{a4paper}{\Aivtrue\tabletfalse\Bvfalse\Avfalse\PADfalse}
\newif\ifBv \Bvtrue
\DeclareOption{b5paper}{\Aivfalse\Bvtrue\tabletfalse\Avfalse\PADfalse}
\newif\ifAv \Avfalse
\DeclareOption{a5paper}{\Avtrue\Aivfalse\Bvfalse\tabletfalse\PADfalse}

\DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
\ProcessOptions*\relax

\LoadClassWithOptions{memoir}
\RequirePackage{iftex}
\ifPDFTeX
\RequirePackage[utf8]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage[italian]{babel}
\RequirePackage[tt={oldstyle=false,tabular,monowidth}]{cfr-lm}
\else
\RequirePackage{fontspec}
\defaultfontfeatures[\rmfamily,\sffamily]{Ligatures=TeX,Numbers=OldStyle}
\defaultfontfeatures[\ttfamily,\tvfamily]{Numbers=Lining}
\setmainfont{Latin Modern Roman}[SmallCapsFont={* Caps}]
\setsansfont{Latin Modern Sans}
\setmonofont{Latin Modern Mono}
\newfontfamily{\tvfamily}{Latin Modern Mono Prop}
\DeclareTextFontCommand{\texttm}{\ttfamily}
\DeclareTextFontCommand{\texttv}{\tvfamily}
\def\textl#1{{\addfontfeature{Numbers=Lining}#1}}
\RequirePackage{unicode-math}
\setmathfont{Latin Modern Math}
\RequirePackage{polyglossia}
\setmainlanguage{italian}
\ifx\tmspace\@undefined
  \newlength{\tmspace}\tmspace=\z@
\fi
```

```
\fi
\RequirePackage{graphicx}
\RequirePackage{metalogo}
\RequirePackage{pict2e}[2009/06/01]
\RequirePackage{microtype}
\RequirePackage{etoolbox}
\RequirePackage{xcolor}
\RequirePackage{natbib}
\RequirePackage{multicol}
\RequirePackage{imakeidx}
\RequirePackage{fancyvrb}
\RequirePackage{afterpage}
\RequirePackage{enumitem}
\RequirePackage[right]{eurosym}
\RequirePackage{listings}
\lstloadlanguages{[LaTeX]TeX}
\normalfont
\ifPAD\setstocksize{160mm}{120mm}\fi
\iftablet\setstocksize{120mm}{90mm}\fi
\ifBv\setstocksize{250mm}{176mm}\fi
\ifAiv\setstocksize{293mm}{210mm}\fi
\ifAv\setstocksize{210mm}{146mm}\fi
\settrims{0pt}{0pt}

\settypeblocksize{\paperwidth}{\dimexpr\paperwidth*\paperwidth/\paperheight}{*}
\setlrmargins{*}{*}{*}
\setulmargins{*}{*}{*}
\setheadfoot{\baselineskip}{2\onelineskip}
\setheaderspaces{*}{\onelineskip}{*}
\setmarginnotes{7pt}{\dimexpr\foremargin-3em}{5pt}
\checkandfixthelayout[nearest]
\def\GetFileInfo#1{%
  \def\filename{#1}%
  \def\@tempb##1 ##2 ##3\relax##4\relax{%
    \def\filedate{##1}%
    \def\fileversion{##2}%
    \def\fileinfo{##3}}%
  \edef\@tempa{\csname ver@#1\endcsname}%
  \expandafter\@tempb\@tempa\relax? ? \relax\relax}
\let\@copymark\@empty % Di default è vuoto
\newcommand*\setcopymark[1]{\gdef\@copymark{#1}}
\newcommand*\@insertcopymark{%
\begin{picture}(0,0)\unitlength=1mm\relax
\if@twoside
\ifodd\value{page}
```

```
            \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
\else
            \put(-\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
\fi
\else
            \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
\fi
\end{picture}%
}
\def\cblayoutsinistro{%
\dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
\dimen258=\dimexpr\evensidemargin+1in\relax
\dimen262=1mm\relax
\dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
\dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
     -\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
\put(0,0){\unitlength=\p@
\put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen264){%
    \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
    \Line(0,0)(\strip@pt\paperwidth,\strip@pt\paperheight)%
    \put(\strip@pt\dimexpr\paperwidth/2\relax,%
        \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
        {\circle{\strip@pt\paperwidth}}%
}
\end{picture}}

\def\cblayoutdestro{%
\dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
\dimen258=\dimexpr\oddsidemargin+1in\relax
\dimen262=1mm\relax
\dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
\dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
    -\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
\put(0,0){\unitlength=\p@
\put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen264){%
    \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
```

```
    \Line(0,\strip@pt\paperheight)(\strip@pt\paperwidth,0)%
    \put(\strip@pt\dimexpr\paperwidth/2\relax,%
       \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
       {\circle{\strip@pt\paperwidth}}%
}
\end{picture}}
\providecommand*\Ars{%
  \textsf{\lower -.48ex\hbox{\rotatebox{-20}{A}}\kern -.3em{rs}}\hskip0pt%
  \kern -.05em\TeX\hskip0pt\kern -.17em\lower -.357ex\hbox{nica}}
\newcommand{\contribguit}{\newpage
\thispagestyle{empty}
\noindent{\Large
Associati anche tu al \GuIT
\hfill
\setlength{\unitlength}{1mm}
 \begin{picture}(50,2)
  \setlength{\fboxsep}{0pt}
  \put(1,-3){\colorbox{gray}{\framebox(50,6.5){}}}
  \put(0,-2){\colorbox{white}{\framebox(50,6.5){%
   \href{http://www.guitex.org/home/it/diventa-socio/associarsi-a-guit}{%
    Fai click per associarti}}}}
 \end{picture}%
}
\bigskip

L'associazione per la diffusione di \TeX\ in Italia, riconosciuta
ufficialmente in ambito internazionale, si sostiene \emph{unicamente}
con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile,
il mezzo principale per ringraziare gli autori è diventare socio.
\medskip

Divenendo soci si ricevono gratuitamente:
\begin{itemize}
\item
l'abbonamento alla rivista \Ars;
\item
il DVD \TeX\ Collection;
\item
un eventuale oggetto legato alle attività del \GuIT.
\end{itemize}

L'adesione al \GuIT\ prevede un quota associativa compresa tra \EUR{12,00}
e \EUR{70,00} a seconda della tipologia di adesione prescelta ed ha
```

```
validità per l'anno solare in corso.
\newpage
}
\nouppercaseheads
\renewcommand{\footruleskip}{0pt}
\makeheadrule{headings}{\textwidth}{0pt}
\makeheadrule{myheadings}{\textwidth}{0pt}
\makeevenfoot{plain}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{plain}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenfoot{headings}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{headings}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenfoot{myheadings}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{myheadings}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenhead{headings}{}{\textsc{\small\leftmark}}{}
\makeoddhead{headings}{}{\textsc{\small\rightmark}}{}
\makeevenhead{myheadings}{}{\textsc{\small\leftmark}}{}
\makeoddhead{myheadings}{}{\textsc{\small\rightmark}}{}
\makepsmarks{headings}{%
\createmark{chapter}{both}{shownumber}{\@chapapp\space}{.\qquad}
\createmark{section}{right}{shownumber}{$\mathsection$\,}{\qquad}
\renewcommand*\indexmark{\markboth{\indexname}{\indexname}}}
\makepagestyle{headingslayout}
\makeevenhead{headingslayout}{\cblayoutsinistro}{\textsc{\small\leftmark}}{}
\makeevenfoot{headingslayout}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddhead{headingslayout}{\cblayoutdestro}{\textsc{\small\rightmark}}{}
\makeoddfoot{headingslayout}{}{\textsc{\thepage}}{\@insertcopymark}
\makechapterstyle{guidatematica}{%
   \renewcommand*{\chapnumfont}{%
      \fontshape{it}\fontsize{40}{40}\selectfont}
   \renewcommand*{\printchaptername}{}
   \renewcommand*{\chapternamenum}{}
   \renewcommand*{\chaptitlefont}{%
       \fontsize{18}{16}\scshape}% sterlineato
   \renewcommand{\printchapternum}{%
       \noindent\rlap{\makebox[\textwidth][r]{\rlap{\makebox[\foremargin][l]{%
       \chapnumfont \thechapter}}}}\printchaptertitle}
   \renewcommand*{\afterchapternum}{}
}
\AtBeginDocument{\chapterstyle{guidatematica}}
\renewcommand*\partnamefont{\normalfont\large\scshape}
\renewcommand*\partnumfont{\normalfont\large\scshape}
\renewcommand*\parttitlefont{\normalfont\huge\scshape}
\setbeforesecskip{-3.5ex plus-1ex minus-0.2ex}
\setbeforesubsecskip{-3ex plus-1ex minus-0.2ex}
\setbeforesubsubsecskip{-2.5ex plus-1ex minus-0.2ex}
```

```
\setbeforeparaskip{1\onelineskip plus1ex minus0.2ex}
\setbeforesubparaskip{1\onelineskip plus1ex minus0.2ex}
\setaftersecskip{1.5ex plus0.2ex}
\setaftersubsecskip{1.5ex plus0.2ex}
\setaftersubsubsecskip{1.5ex plus0.2ex}
\setafterparaskip{-1em}
\setaftersubparaskip{-1em}
\setsubparaindent{\parindent}

\setsecheadstyle{\large\scshape\raggedright}
\setsubsecheadstyle{\large\scshape\raggedright}
\setsubsubsecheadstyle{\large\scshape\raggedright}
\setparaheadstyle{\small\scshape}
\setsubparaheadstyle{\small\scshape}

\aliaspagestyle{part}{empty}
\aliaspagestyle{chapter}{empty}
\copypagestyle{titlepage}{headings}
\renewcommand\cftpartpagefont{\scshape}
\renewcommand\cftpartfont{\large\scshape}
\renewcommand\cftchapterfont{\large\scshape}
\renewcommand\cftchapterpagefont{\scshape}
\renewcommand*{\cftchapterfillnum}[1]{%
    {\cftchapterleader}\nobreak
    \cftchapterformatpnum{#1}%
    \cftchapterafterpnum\par\nobreak}
\setlength{\cftbeforechapterskip}{1.0em \@plus 2\p@}
\captiondelim{\quad}
\captionnamefont{\small\scshape}
\captiontitlefont{\small}
\captionstyle[\centering]{}
\hangcaption
\captionwidth{\dimexpr\textwidth-2\parindent\relax}\changecaptionwidth
\renewcommand\descriptionlabel[1]{\hspace\labelsep\normalfont\scshape #1}
\renewcommand\blockdescriptionlabel[1]{\normalfont\scshape #1}
\providecommand\plaindescriptionlabel[1]{\hspace\labelsep\normalfont #1}

\renewlist{blockdescription}{description}{1}
\setlist[blockdescription]{before={\let\makelabel\blockdescriptionlabel},
leftmargin=\z@,labelsep*=0.5em,labelindent=\z@,labelwidth=\z@}

\newlist{plaindescription}{description}{1}
\setlist[plaindescription]{before={\let\makelabel\plaindescriptionlabel}}

\newlist{compactenumerate}{enumerate}{1}
```

```
\setlist[compactenumerate,1]{label=\arabic*.,
   noitemsep, partopsep=\z@, topsep=.25\onelineskip}

\newlist{compactitemize}{itemize}{4}
\setlist[compactitemize]{label=•,
   noitemsep,partopsep=\z@,topsep=.25\onelineskip}

\newlist{compactdescription}{description}{1}
\setlist[compactdescription]{%
style=sameline,noitemsep,partopsep=\z@,topsep=.25\onelineskip}
\AtBeginDocument{\setsecnumdepth{subsection}
   \settocdepth{subsection}\maxsecnumdepth{subsection}
   \maxtocdepth{subsection}}
 \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
 \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}
\unless\ifPDFTeX
 \AtBeginDocument{\ifcsstring{xpg@main@language}{english}{\relax}{%
    \mathcode`\,=\string"8000}}
\else
 \AtBeginDocument{\ifcsstring{languagename}{english}{\relax}{%
    \mathcode`\,=\string"8000}}
\fi
{\catcode `,=\active \gdef,{\futurelet\let@token\m@thcomma}}%
\AtEndPreamble{%
\gdef\m@thcomma#1{%
\unless\ifcat\noexpand\let@token*%
  \virgola
\else
\expandafter\expandafter\expandafter
    \ifcsname\expandafter\@gobble\string#1\endcsname
  \virgola
\else
  \ifnum`#1<`0\relax
    \virgola
  \else
    \ifnum`#1>`9\relax
      \virgola
    \else
      \virgoladecimale
    \fi
  \fi
\fi
\fi#1}%
}
\renewcommand{\bibsection}{\chapter{\bibname}\prebibhook}
```

```
\let\imki@idxprologue\empty
\def\imki@columns{2}
\renewenvironment{theindex}
{%
\clearforchapter
    \csname phantomsection\endcsname
    \chapter{\indexname}%
    \indexmark%
    \parindent\z@
    \parskip\z@ \@plus .3\p@\relax
    \let\item\@idxitem
    \begin{multicols}{\imki@columns}[\imki@idxprologue]
    \raggedright
}
{%
    \end{multicols}\gdef\imki@idxprologue{}\clearpage
}
\AtBeginDocument{\bibliographystyle{guidatematica}%
\unless\ifPDFTeX
   \ifcsstring{xpg@main@language}{italian}%
     {\csgappto{init@extras@italian}{\nofrench@indent}}{}%
\fi}
\newenvironment{titlepage}{\clearpage
\pagestyle{titlepage}
    \makeoddhead{titlepage}{}{\smash{\Large\scshape\@author}}{}
    \makeoddfoot{titlepage}{}{\smash{\large\scshape\@date}}{}
}{\clearpage}

\providecommand\subtitle[1]{\gdef\@subtitle{#1}}
\let\@subtitle\empty

\renewcommand\maketitle{
    \begin{titlepage}
    \vspace*{\stretch{.7}}
    {\centering\LARGE\scshape\@title\par}
    \ifdefvoid{\@subtitle}{}{\vspace{2\onelineskip}
    {\centering\scshape\Large\@subtitle\par}}
    \vspace{\stretch{.7}}
    {\centering\includegraphics[width=0.23\paperwidth]{logoguittondo}\par}
    \vspace*{\stretch{.7}}
    \end{titlepage}
}
\newenvironment{medaglione}[1][\linewidth]{%
    \begin{lrbox}{0}%
    \begin{minipage}{\dimexpr#1-2\fboxsep-2\fboxrule}
```

```
}{%
    \end{minipage}\end{lrbox}\fbox{\usebox{0}}\relax
}

\newenvironment{sintassi}{\flushleft\medaglione}%
   {\endmedaglione\endflushleft}
\newcommand*\hz{\nobreak\hskip\z@}
\renewcommand*\meta[1]{$\langle$\textit{\hz#1}$\rangle$}
\renewcommand*\marg[1]{\texttt{\char123}\meta{#1}\texttt{\char125}}
\renewcommand*\oarg[1]{\texttt{[}\meta{#1}\texttt{]}}
\newcommand*\Arg[1]{\texttt{\char123\relax#1\char125}}
\def\GT@splitargs#1,#2!{\def\@tempA{#1}\def\@tempB{#2}}
\newcommand\garg[1]{\GT@splitargs#1!\texttt{(}\meta{\@tempA}%
   \texttt{,}\meta{\@tempB}\texttt{)}}
\newcommand*\comando[1]{\texttt{\string#1}}
\renewcommand*{\cs}[1]{\texttt{\char92#1}\index{#1@\texttt{\char92#1}}}
\let\csindex\cs
\newcommand*\Sambiente[2]{\cs{begin}\marg{#1}\oarg{#2}\,\dots
   \cs{end}\marg{#1}}
\newcommand*\Dambiente[3]{\cs{begin}\marg{#1}\oarg{#2}\oarg{#3}\,\dots
   \cs{end}\marg{#1}}
\newcommand*\Bambiente[1]{\cs{begin}\texttt{\char123#1\char125}}
\newcommand*\Eambiente[1]{\cs{end}\texttt{\char123#1\char125}}
\DeclareRobustCommand*\ambstyle[1]{\textsf{\slshape#1}}
\DeclareRobustCommand*\classstyle[1]{\texttv{\itshape#1}}
\DeclareRobustCommand*\filestyle[1]{\texttm{\textl{#1}}}
\unless\ifPDFTeX
    \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape{#1}}}
\else
    \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape\textl{#1}}}
\fi
\DeclareRobustCommand*\progstyle[1]{\textsf{#1}}
\DeclareRobustCommand*\prog[1]{\progstyle{#1}%
   \index{programma!#1@\progstyle{#1}}}
\DeclareRobustCommand*\pack[1]{\packstyle{#1}%
   \index{pacchetto!#1@\packstyle{#1}}}
\DeclareRobustCommand*\class[1]{\classstyle{#1}%
   \index{classe!#1@\classstyle{#1}}}
\DeclareRobustCommand*\file[1]{\filestyle{#1}%
   \index{file!#1@\filestyle{#1}}}
\DeclareRobustCommand*\amb[1]{\ambstyle{#1}%
   \index{ambiente!#1@\ambstyle{#1}}}
\DeclareRobustCommand*\opzstyle[1]{\textsl{\textl{#1}}}
\DeclareRobustCommand*\contastyle[1]{\texttm{#1}}
\DeclareRobustCommand*\stilestyle[1]{\texttm{#1}}
```

```
\DeclareRobustCommand*\numeristyle[1]{\texttm{#1}}
\DeclareRobustCommand*\umisurastyle[1]{\texttm{#1}}
\DeclareRobustCommand*\chiavestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\descrittorestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\posizionestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\allineamentostyle[1]{\texttm{#1}}
\DeclareRobustCommand*\opz[1]{\opzstyle{#1}%
   \index{opzione!#1@\opzstyle{#1}}}
\DeclareRobustCommand*\conta[1]{\contastyle{#1}%
   \index{contatore!#1@\contastyle{#1}}}
\DeclareRobustCommand*\stile[1]{\stilestyle{#1}%
   \index{stile della pagina!#1@\stilestyle{#1}}}
\DeclareRobustCommand*\numeri[1]{\numeristyle{#1}%
   \index{numerazione!#1@\numeristyle{#1}}}
\DeclareRobustCommand*\umisura[1]{\umisurastyle{#1}%
   \index{unit\`a di misura!#1@\umisurastyle{#1}}}
\DeclareRobustCommand*\chiave[1]{\chiavestyle{#1}%
   \index{chiave!#1@\chiavestyle{#1}}}
\DeclareRobustCommand*\descrittore[1]{\descrittorestyle{#1}%
   \index{descrittore di colonna!#1@\descrittorestyle{#1}}}
\DeclareRobustCommand*\posizione[1]{\posizionestyle{#1}%
   \index{posizione degli oggetti flottanti!#1@\posizionestyle{#1}}}
\DeclareRobustCommand*\allineamento[1]{\allineamentostyle{#1}%
   \index{codice di allineamento!#1@\allineamentostyle{#1}}}
\DeclareRobustCommand*\GuIT{{\dimen@=\f@size\p@\dimen2=20.88\p@
\edef\guitscale{\strip@pt\dimexpr\dimen@*\p@/\dimen2\relax}%
\raisebox{-0.93ex}[\z@][\z@]{%
   \includegraphics[scale=\guitscale]{logoguitlineare}}}}
\let\guit\GuIT
%% Copyright 2012, 2013 GuIT, Gruppo utilizzatori Italiani di TeX e LaTeX
%%
%% Distributable under the LaTeX Project Public License,
%% version 1.3c or higher (your choice). The latest version of
%% this license is at: http://www.latex-project.org/lppl.txt
%%
%% This Work has the status of ‘maintained’
%%
%% The Current Maintainer is the GuIT staff
%%
%% This work consists of this file guidatematica.dtx,
%% and the derived files guidatematica.cls and guidatematica.pdf
%% plus the associated documentation guidatematica-doc.tex
%% and guidatematica-doc.pdf.
%%
%%
```

```
%% End of file 'guidatematica.cls'.
```

5.3 Il file di documentazione debitamente composto

Infine il file di documentazione è riportato nelle pagine 109 e seguenti.

5.4 Conclusione

È chiaro che con il metodo esposto nel capitolo 3 si possono creare classi o pacchetti di qualsiasi complessità; si consiglia invece di cominciare con i primi metodi esposti e si raccomanda di usare sempre il file documentato `.dtx`, perché solo con il codice ben documentato è possibile riprendere in mano le definizioni usate, magari per usare le vecchie classi per costruirne di altre, o semplicemente per migliorare le prestazioni delle macro già inserite.

Ovviamente le correzioni sono anche facilitate; il bello del software libero, come le classi e i pacchetti del sistema TEX, è che beneficiano del feedback degli utenti che segnalano volentieri agli autori del pacchetto le loro osservazioni e i bug trovati. E gli autori sono ben lieti che vengano loro segnalati gli errori.

Per concludere, vorrei segnalare ancora come fare per caricare eventualmente le proprie classi o pacchetti (di interesse internazionale) nella Rete degli Archivi Completi di TEX (CTAN, Comprehensive TEX Archive Network); si vada per esempio nel sito http://www.tex.ac.uk/upload/ che contiene una *pagina interattiva* per caricare i pacchetti; conviene leggere tutta la pagina e le istruzioni lì contenute.

In buona sostanza, ogni contributo deve essere compresso in formato `.zip`; questo file deve contenere il file `.dtx` e `.pdf` e ogni altro file di ulteriore documentazione che non derivi dalla elaborazione del file `.dtx`; esso deve contenere un ulteriore file `manifest.txt` di puro testo[2] che dice espressamente chi è l'autore o il manutentore e quale sia la licenza per l'uso del pacchetto; infine esso deve contenere un file `README` di puro testo, ma senza estensione, che elenchi il file principali e quelli derivati, come fare

[2] I fine-riga devono essere omogenei: o solo quelli di Windows, o solo quelli di tipo UNIX; qualche volta succede di usare entrambi i tipi di fine-riga, se si è lavorato un po' con macchine Windows e un po' con macchine Linux o Mac, e questo non piace ai curatori dei siti CTAN.

ad ottenerli dal file `.dtx` e in quali cartelle debbano essere collocati se si esegue una installazione manuale.

Poi nella *pagina interattiva* indicata sopra si debbono riempire tutti i campi e si deve scegliere (concordemente a quanto scritto in `manifest.txt` incluso nel file `.zip`) la licenza a cui è sottoposto il pacchetto; conviene scegliere sempre la licenza LPPL (LaTeX Project Public License), a meno che non ci siano altri validi motivi per scegliere una delle altre licenze; però, se non si specifica una licenza libera, il curatore dell'archivio respinge il contributo.

Invece, per caricare questi pacchetti nel sito di GuIT, per ora non ci sono procedure standardizzate né esiste ancora una pagina del sito adatta a raccogliere questi contributi. Al momento di scrivere questa introduzione, il GuIT Web Team è al lavoro per definire l'amministrazione di questa pagina.

La classe guidatematica.dtx*

GuIT†

26 ottobre 2017

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. 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'autore della guida.
- 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.

*Version number v.1.5; ultima revisione 2014/09/03.
†Web: 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 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 ricordassi 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 contenti 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 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 questo opzioni bisogna

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

```
\DeclareOption{ipertesto}{%
\AtBeginDocument{\RequirePackage[colorlinks,linkcolor={blue},
  citecolor={blue!80!black},urlcolor={blue}]{hyperref}}
}

\newif\ifPAD\PADfalse
\DeclareOption{pad}{\PADtrue\tabletfalse\Aivfalse\Bvfalse\Avfalse}
\newif\iftablet\tabletfalse
\DeclareOption{tablet}{\tablettrue\Aivfalse\Bvfalse\Avfalse\PADfalse}
\newif\ifAiv \Aivfalse
\DeclareOption{a4paper}{\Aivtrue\tabletfalse\Bvfalse\Avfalse\PADfalse}
\newif\ifBv \Bvtrue
\DeclareOption{b5paper}{\Aivfalse\Bvtrue\tabletfalse\Avfalse\PADfalse}
\newif\ifAv \Avfalse
\DeclareOption{a5paper}{\Avtrue\Aivfalse\Bvfalse\tabletfalse\PADfalse}

\DeclareOption*{\PassOptionsToClass{\CurrentOption}{memoir}}
\ProcessOptions*\relax

\LoadClassWithOptions{memoir}
```

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

```
\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.

```
\ifPDFTeX
\RequirePackage[utf8]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage[italian]{babel}
\RequirePackage[tt={oldstyle=false,tabular,monowidth}]{cfr-lm}
\else
\RequirePackage{fontspec}
\defaultfontfeatures[\rmfamily,\sffamily]{Ligatures=TeX,Numbers=OldStyle}
\defaultfontfeatures[\ttfamily,\tvfamily]{Numbers=Lining}
\setmainfont{Latin Modern Roman}[SmallCapsFont={* Caps}]
\setsansfont{Latin Modern Sans}
\setmonofont{Latin Modern Mono}
\newfontfamily{\tvfamily}{Latin Modern Mono Prop}
\DeclareTextFontCommand{\texttm}{\ttfamily}
\DeclareTextFontCommand{\texttv}{\tvfamily}
\def\textl#1{{\addfontfeature{Numbers=Lining}#1}}
\RequirePackage{unicode-math}
\setmathfont{Latin Modern Math}
\RequirePackage{polyglossia}
\setmainlanguage{italian}
\ifx\tmspace\@undefined
  \newlength{\tmspace}\tmspace=\z@
```



```
\fi
\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 (immagine pdf) la si sarebbe potuta limitare all'uso con `XeLaTeX` o di `Lua\LaTeX`; siccome alla data attuale il pacchetto `guit.sty` è sottoposto a revisione per l'uso con i font OpenType, è molto meglio usare temporaneamente l'immagine in ogni caso e modificare questa classe solo con l'aggiunta di un nuovo pacchetto quando `guit.sty` sarà diventato compatibile con tuttii 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.

```
\RequirePackage{graphicx}
\RequirePackage{metalogo}
\RequirePackage{pict2e}[2009/06/01]
\RequirePackage{microtype}
\RequirePackage{etoolbox}
\RequirePackage{xcolor}
\RequirePackage{natbib}
\RequirePackage{multicol}
\RequirePackage{imakeidx}
\RequirePackage{fancyvrb}
\RequirePackage{afterpage}
\RequirePackage{enumitem}
\RequirePackage[right]{eurosym}
\RequirePackage{listings}
\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 inconvenient 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.

```
\normalfont
\ifPAD\setstocksize{160mm}{120mm}\fi
\iftablet\setstocksize{120mm}{90mm}\fi
\ifBv\setstocksize{250mm}{176mm}\fi
\ifAiv\setstocksize{293mm}{210mm}\fi
\ifAv\setstocksize{210mm}{146mm}\fi
\settrims{0pt}{0pt}
```

```

\settypeblocksize{\paperwidth}{\dimexpr\paperwidth*\paperwidth/\paperheight}{*}
\setlrmargins{*}{*}{*}
\setulmargins{*}{*}{*}
\setheadfoot{\baselineskip}{2\onelineskip}
\setheaderspaces{*}{\onelineskip}{*}
\setmarginnotes{7pt}{\dimexpr\foremargin-3em}{5pt}
\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.

```
\def\GetFileInfo#1{%
  \def\filename{#1}%
  \def\@tempb##1 ##2 ##3\relax##4\relax{%
    \def\filedate{##1}%
    \def\fileversion{##2}%
    \def\fileinfo{##3}}%
  \edef\@tempa{\csname ver@#1\endcsname}%
  \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.

```
\let\@copymark\@empty % Di default è vuoto
\newcommand*\setcopymark[1]{\gdef\@copymark{#1}}
\newcommand*\@insertcopymark{%
\begin{picture}(0,0)\unitlength=1mm\relax
\if@twoside
\ifodd\value{page}
            \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
\else
            \put(-\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
\fi
\else
            \put(\strip@pt\dimexpr\foremargin/2\relax,\strip@pt\footskip)%
            {\rotatebox{90}{\makebox(0,0)[l]{\@copymark}}}
\fi
\end{picture}%
}
```


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.

```
\def\cblayoutsinistro{%
\dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
\dimen258=\dimexpr\evensidemargin+1in\relax
\dimen262=1mm\relax
\dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
\dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
     -\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
\put(0,0){\unitlength=\p@
\put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen264){%
    \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
    \Line(0,0)(\strip@pt\paperwidth,\strip@pt\paperheight)%
    \put(\strip@pt\dimexpr\paperwidth/2\relax,%
        \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
        {\circle{\strip@pt\paperwidth}}%
}
\end{picture}}

\def\cblayoutdestro{%
\dimen256=\dimexpr\headheight+\topmargin+1in-4pt-\paperheight\relax
\dimen258=\dimexpr\oddsidemargin+1in\relax
\dimen262=1mm\relax
\dimen260=\dimexpr \paperwidth*\p@/\dimen262\relax
\dimen264=\dimexpr \paperheight-\topmargin-\headheight-1in
    -\headsep-\textheight\relax
\begin{picture}(0,0)(\strip@pt\dimen258,-\strip@pt\dimen256)%
\put(0,0){\unitlength=\p@
\put(0,0){\framebox(\strip@pt\paperwidth,\strip@pt\paperheight){}}%
\color{red}%
\put(\strip@pt\dimen258,\strip@pt\dimen264){%
    \framebox(\strip@pt\textwidth,\strip@pt\textheight){}}%
    \Line(0,\strip@pt\paperheight)(\strip@pt\paperwidth,0)%
    \put(\strip@pt\dimexpr\paperwidth/2\relax,%
       \strip@pt\dimexpr\dimen264+\textheight/2\relax)%
       {\circle{\strip@pt\paperwidth}}%
}
\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 può essere utilizzato a inizio o fine documento. Tuttavia, per garantire maggiore visibilità, si consiglia di inserire `\contribguit` dopo `\frontmatter` e prima del capitolo di introduzione o eventuali pagine con informazioni sulle licenze d'uso.

```
\providecommand*\Ars{%
  \textsf{\lower -.48ex\hbox{\rotatebox{-20}{A}}\kern -.3em{rs}}\hskip0pt%
  \kern -.05em\TeX\hskip0pt\kern -.17em\lower -.357ex\hbox{nica}}
\newcommand{\contribguit}{\newpage
\thispagestyle{empty}
\noindent{\Large
Associati anche tu al \GuIT
\hfill
\setlength{\unitlength}{1mm}
 \begin{picture}(50,2)
  \setlength{\fboxsep}{0pt}
  \put(1,-3){\colorbox{gray}{\framebox(50,6.5){}}}
  \put(0,-2){\colorbox{white}{\framebox(50,6.5){%
   \href{http://www.guitex.org/home/it/diventa-socio/associarsi-a-guit}{%
    Fai click per associarti}}}}
 \end{picture}%
}
\bigskip

L'associazione per la diffusione di \TeX\ in Italia, riconosciuta
ufficialmente in ambito internazionale, si sostiene \emph{unicamente}
con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile,
il mezzo principale per ringraziare gli autori è diventare socio.
\medskip

Divenendo soci si ricevono gratuitamente:
\begin{itemize}
\item
l'abbonamento alla rivista \Ars;
\item
il DVD \TeX\ Collection;
\item
un eventuale oggetto legato alle attività del \GuIT.
\end{itemize}

L'adesione al \GuIT\ prevede un quota associativa compresa tra \EUR{12,00}
e \EUR{70,00} a seconda della tipologia di adesione prescelta ed ha
validità per l'anno solare in corso.
\newpage
}
```

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.

```
\nouppercaseheads
%\renewcommand{\footruleheight}{\normalrulethickness}
\renewcommand{\footruleskip}{0pt}
\makeheadrule{headings}{\textwidth}{0pt}
\makeheadrule{myheadings}{\textwidth}{0pt}
\makeevenfoot{plain}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{plain}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenfoot{headings}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{headings}{}{\textsc{\thepage}}{\@insertcopymark}
\makeevenfoot{myheadings}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddfoot{myheadings}{}{\textsc{\thepage}}{\@insertcopymark}
```

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

```
\makeevenhead{headings}{}{\textsc{\small\leftmark}}{}
\makeoddhead{headings}{}{\textsc{\small\rightmark}}{}
\makeevenhead{myheadings}{}{\textsc{\small\leftmark}}{}
\makeoddhead{myheadings}{}{\textsc{\small\rightmark}}{}
```

Vengono definiti i comandi per "decorare" le intestazioni delle testatine.

```
\makepsmarks{headings}{%
\createmark{chapter}{both}{shownumber}{\@chapapp\space}{.\qquad}
\createmark{section}{right}{shownumber}{$\mathsection$\,}{\qquad}
\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`.

```
\makepagestyle{headingslayout}
\makeevenhead{headingslayout}{\cblayoutsinistro}{\textsc{\small\leftmark}}{}
\makeevenfoot{headingslayout}{\@insertcopymark}{\textsc{\thepage}}{}
\makeoddhead{headingslayout}{\cblayoutdestro}{\textsc{\small\rightmark}}{}
\makeoddfoot{headingslayout}{}{\textsc{\thepage}}{\@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.

```
\makechapterstyle{guidatematica}{%
   \renewcommand*{\chapnumfont}{%
      \fontshape{it}\fontsize{40}{40}\selectfont}
   \renewcommand*{\printchaptername}{}
   \renewcommand*{\chapternamenum}{}
   \renewcommand*{\chaptitlefont}{%
       \fontsize{18}{16}\scshape}% sterlineato
   \renewcommand{\printchapternum}{%
       \noindent\rlap{\makebox[\textwidth][r]{\rlap{\makebox[\foremargin][l]{%
       \chapnumfont \thechapter}}}}\printchaptertitle}
```

```
    \renewcommand*{\afterchapternum}{}
}
\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.

```
\renewcommand*\partnamefont{\normalfont\large\scshape}
\renewcommand*\partnumfont{\normalfont\large\scshape}
\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.

```
\setbeforesecskip{-3.5ex plus-1ex minus-0.2ex}
\setbeforesubsecskip{-3ex plus-1ex minus-0.2ex}
\setbeforesubsubsecskip{-2.5ex plus-1ex minus-0.2ex}
\setbeforeparaskip{1\onelineskip plus1ex minus0.2ex}
\setbeforesubparaskip{1\onelineskip plus1ex minus0.2ex}
\setaftersecskip{1.5ex plus0.2ex}
\setaftersubsecskip{1.5ex plus0.2ex}
\setaftersubsubsecskip{1.5ex plus0.2ex}
\setafterparaskip{-1em}
\setaftersubparaskip{-1em}
\setsubparaindent{\parindent}

\setsecheadstyle{\large\scshape\raggedright}
\setsubsecheadstyle{\large\scshape\raggedright}
\setsubsubsecheadstyle{\large\scshape\raggedright}
\setparaheadstyle{\small\scshape}
\setsubparaheadstyle{\small\scshape}

\aliaspagestyle{part}{empty}
\aliaspagestyle{chapter}{empty}
\copypagestyle{titlepage}{headings}
\renewcommand\cftpartpagefont{\scshape}
\renewcommand\cftpartfont{\large\scshape}
\renewcommand\cftchapterfont{\large\scshape}
\renewcommand\cftchapterpagefont{\scshape}
\renewcommand*{\cftchapterfillnum}[1]{%
    {\cftchapterleader}\nobreak
    \cftchapterformatpnum{#1}%
    \cftchapterafterpnum\par\nobreak}
\setlength{\cftbeforechapterskip}{1.0em \@plus 2\p@}
```

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

```
\captiondelim{\quad}
\captionnamefont{\small\scshape}
\captiontitlefont{\small}
\captionstyle[\centering]{}
\hangcaption
\captionwidth{\dimexpr\textwidth-2\parindent\relax}\changecaptionwidth
```

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 maiuscoletto; 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.

```
\renewcommand\descriptionlabel[1]{\hspace\labelsep\normalfont\scshape #1}
\renewcommand\blockdescriptionlabel[1]{\normalfont\scshape #1}
\providecommand\plaindescriptionlabel[1]{\hspace\labelsep\normalfont #1}

\renewlist{blockdescription}{description}{1}
\setlist[blockdescription]{before={\let\makelabel\blockdescriptionlabel},
leftmargin=\z@,labelsep*=0.5em,labelindent=\z@,labelwidth=\z@}

\newlist{plaindescription}{description}{1}
\setlist[plaindescription]{before={\let\makelabel\plaindescriptionlabel}}

\newlist{compactenumerate}{enumerate}{1}
\setlist[compactenumerate,1]{label=\arabic*.,
   noitemsep, partopsep=\z@, topsep=.25\onelineskip}

\newlist{compactitemize}{itemize}{4}
\setlist[compactitemize]{label=•,
   noitemsep,partopsep=\z@,topsep=.25\onelineskip}

\newlist{compactdescription}{description}{1}
\setlist[compactdescription]{%
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.

```
\AtBeginDocument{\setsecnumdepth{subsection}
   \settocdepth{subsection}\maxsecnumdepth{subsection}
   \maxtocdepth{subsection}}
```

Viene ora definita la virgola intelligente; si veda una descrizione più dettagliata della sua utilità in matematica per esempio della documentazioone di questa classe `gidatematica-doc.pdf`. 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.

```
 \DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
 \DeclareMathSymbol{\virgoladecimale}{\mathord}{letters}{"3B}
\unless\ifPDFTeX
 \AtBeginDocument{\ifcsstring{xpg@main@language}{english}{\relax}{%
    \mathcode`\,=\string"8000}}
\else
 \AtBeginDocument{\ifcsstring{languagename}{english}{\relax}{%
    \mathcode`\,=\string"8000}}
\fi
{\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 analfabetico 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.

```
\AtEndPreamble{%
\gdef\m@thcomma#1{%
\unless\ifcat\noexpand\let@token*%
  \virgola
\else
\expandafter\expandafter\expandafter
    \ifcsname\expandafter\@gobble\string#1\endcsname
  \virgola
\else
  \ifnum`#1<`0\relax
    \virgola
  \else
    \ifnum`#1>`9\relax
      \virgola
    \else
      \virgoladecimale
    \fi
  \fi
\fi
\fi#1}%
}
```


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`.

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

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.

```
\let\imki@idxprologue\empty
\def\imki@columns{2}
\renewenvironment{theindex}
{%
\clearforchapter
    \csname phantomsection\endcsname
    \chapter{\indexname}%
    \indexmark%
    \parindent\z@
    \parskip\z@ \@plus .3\p@\relax
    \let\item\@idxitem
    \begin{multicols}{\imki@columns}[\imki@idxprologue]
    \raggedright
}
{%
    \end{multicols}\gdef\imki@idxprologue{}\clearpage
}
```

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.

```
\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.

```
\unless\ifPDFTeX
  \ifcsstring{xpg@main@language}{italian}%
```

```
    {\csgappto{init@extras@italian}{\nofrench@indent}}{}%
\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.

```
\newenvironment{titlepage}{\clearpage
\pagestyle{titlepage}
    \makeoddhead{titlepage}{}{\smash{\Large\scshape\@author}}{}
    \makeoddfoot{titlepage}{}{\smash{\large\scshape\@date}}{}
}{\clearpage}

\providecommand\subtitle[1]{\gdef\@subtitle{#1}}
\let\@subtitle\empty

\renewcommand\maketitle{
    \begin{titlepage}
    \vspace*{\stretch{.7}}
    {\centering\LARGE\scshape\@title\par}
    \ifdefvoid{\@subtitle}{}{\vspace{2\onelineskip}
    {\centering\scshape\Large\@subtitle\par}}
    \vspace{\stretch{.7}}
    {\centering\includegraphics[width=0.23\paperwidth]{logoguittondo}\par}
    \vspace*{\stretch{.7}}
    \end{titlepage}
}
```

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.

I 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.

```
\newenvironment{medaglione}[1][\linewidth]{%
    \begin{lrbox}{0}%
    \begin{minipage}{\dimexpr#1-2\fboxsep-2\fboxrule}
}{%
    \end{minipage}\end{lrbox}\fbox{\usebox{0}}\relax
}

\newenvironment{sintassi}{\flushleft\medaglione}%
   {\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` per indicare fra graffe un argomento specifico `{123}`.

```
\newcommand*\hz{\nobreak\hskip\z@}
\renewcommand*\meta[1]{$\langle$\textit{\hz#1}$\rangle$}
\renewcommand*\marg[1]{\texttt{\char123}\meta{#1}\texttt{\char125}}
\renewcommand*\oarg[1]{\texttt{[}\meta{#1}\texttt{]}}
\newcommand*\Arg[1]{\texttt{\char123\relax#1\char125}}
\def\GT@splitargs#1,#2!{\def\@tempA{#1}\def\@tempB{#2}}
\newcommand\garg[1]{\GT@splitargs#1!\texttt{(}\meta{\@tempA}%
   \texttt{,}\meta{\@tempB}\texttt{)}}
\newcommand*\comando[1]{\texttt{\string#1}}
\renewcommand*{\cs}[1]{\texttt{\char92#1}\index{#1@\texttt{\char92#1}}}
\let\csindex\cs
\newcommand*\Sambiente[2]{\cs{begin}\marg{#1}\oarg{#2}\,\dots
   \cs{end}\marg{#1}}
\newcommand*\Dambiente[3]{\cs{begin}\marg{#1}\oarg{#2}\oarg{#3}\,\dots
   \cs{end}\marg{#1}}
\newcommand*\Bambiente[1]{\cs{begin}\texttt{\char123#1\char125}}
\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 informa-

zioni 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.

```
%
\DeclareRobustCommand*\ambstyle[1]{\textsf{\slshape#1}}
\DeclareRobustCommand*\classstyle[1]{\texttv{\itshape#1}}
\DeclareRobustCommand*\filestyle[1]{\texttm{\textl{#1}}}
\unless\ifPDFTeX
    \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape{#1}}}
\else
    \DeclareRobustCommand*\packstyle[1]{\texttm{\itshape\textl{#1}}}
\fi
\DeclareRobustCommand*\progstyle[1]{\textsf{#1}}
\DeclareRobustCommand*\prog[1]{\progstyle{#1}%
   \index{programma!#1@\progstyle{#1}}}
\DeclareRobustCommand*\pack[1]{\packstyle{#1}%
   \index{pacchetto!#1@\packstyle{#1}}}
\DeclareRobustCommand*\class[1]{\classstyle{#1}%
   \index{classe!#1@\classstyle{#1}}}
\DeclareRobustCommand*\file[1]{\filestyle{#1}%
   \index{file!#1@\filestyle{#1}}}
\DeclareRobustCommand*\amb[1]{\ambstyle{#1}%
   \index{ambiente!#1@\ambstyle{#1}}}
%
\DeclareRobustCommand*\opzstyle[1]{\textsl{\textl{#1}}}
\DeclareRobustCommand*\contastyle[1]{\texttm{#1}}
\DeclareRobustCommand*\stilestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\numeristyle[1]{\texttm{#1}}
\DeclareRobustCommand*\umisurastyle[1]{\texttm{#1}}
\DeclareRobustCommand*\chiavestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\descrittorestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\posizionestyle[1]{\texttm{#1}}
\DeclareRobustCommand*\allineamentostyle[1]{\texttm{#1}}
%
\DeclareRobustCommand*\opz[1]{\opzstyle{#1}%
   \index{opzione!#1@\opzstyle{#1}}}
\DeclareRobustCommand*\conta[1]{\contastyle{#1}%
   \index{contatore!#1@\contastyle{#1}}}
\DeclareRobustCommand*\stile[1]{\stilestyle{#1}%
   \index{stile della pagina!#1@\stilestyle{#1}}}
\DeclareRobustCommand*\numeri[1]{\numeristyle{#1}%
   \index{numerazione!#1@\numeristyle{#1}}}
\DeclareRobustCommand*\umisura[1]{\umisurastyle{#1}%
   \index{unit\`a di misura!#1@\umisurastyle{#1}}}
\DeclareRobustCommand*\chiave[1]{\chiavestyle{#1}%
   \index{chiave!#1@\chiavestyle{#1}}}
\DeclareRobustCommand*\descrittore[1]{\descrittorestyle{#1}%
   \index{descrittore di colonna!#1@\descrittorestyle{#1}}}
\DeclareRobustCommand*\posizione[1]{\posizionestyle{#1}%
   \index{posizione degli oggetti flottanti!#1@\posizionestyle{#1}}}
\DeclareRobustCommand*\allineamento[1]{\allineamentostyle{#1}%
```

Bibliografia

LaTeX 3 Project Team (2006). «LaTeX 2_ε for class and package writers». PDF document. Leggibile con `texdoc clsguide` nella distribuzione TeX Live.

Abrahams, P., Berry, K. e Hargreaves, K. (1991). *TeX per l'impaziente*. Addison Wesley. In italiano.

Beccari, C. (2011a). «Introduzione all'arte della composizione tipografica con LaTeX». PDF document. URL http://www.guitex.org/home/.

— (2011b). «Package canoniclayout». PDF document. Leggibile con `texdoc canoniclayout` nella distribuzione TeX Live.

Beccari, C. e Gregorio, E. (2010). «Package imakeidx». PDF document. Leggibile con `texdoc imakeidx` nella distribuzione TeX Live.

Berry, K. (2011). «The TeX Live guide». PDF document. Leggibile con `texdoc texlive` nella distribuzione TeX Live.

Bezos, J. (2011a). «Customizing lists with the enumitem package». PDF document. Leggibile con `texdoc enumitem` nella distribuzione TeX Live.

— (2011b). «The titlesec, titleps, titletoc, packages». PDF document. Leggibile con `texdoc titlesec` nella distribuzione TeX Live.

Bringhurst, R. (2004). *The elements of typographic style*. Hartley & Marks.

Carlisle, D. (2007). «The file ltxdoc.dtx for use with LaTeX 2_ε». PDF document. Leggibile con `texdoc ltxdoc` nella distribuzione TeX Live.

Charette, F. (10). «Polyglossia: a babel replacement for XeLaTeX». PDF document. Leggibile con `texdoc polyglossia` nella distribuzione TeX Live.

Cochran, S. D. (2005). «The subfig package». PDF document. Leggibile con `texdoc subfig` nella distribuzione TeX Live.

Daly, P. W. (2010). «Natural sciences citations and references». PDF document. Leggibile con `texdoc natbib` nella distribuzione TeX Live.

Gässlein, H., Nieprasch, R. e Tkadlec, J. (2011). «The pict2e package». PDF document. Leggibile con `texdoc pict2e` nella distribuzione TeX Live.

Gilardi, G. (1993). *Il TeX*. Zanichelli/Decibel.

Goossens, M. (1997). *The LaTeX Graphics companion*. Addison Wesley.

— (1999). *The LaTeX Web Companion*. Addison Wesley.

Gregorio, E. (2009). «Appunti di programmazione in LaTeX e TeX». PDF document. URL http://profs.sci.univr.it/~gregorio/egtex.html.

— (2010a). «LaTeX, breve guida ai pacchetti di uso più comune». PDF document. URL http://profs.sci.univr.it/~gregorio/egtex.html.

— (2010b). «Installare TeX live 2010 su ubuntu». PDF document. URL http://profs.sci.univr.it/~gregorio/egtex.html. Nelle istruzioni contenute nel testo bisogna cambiare le indicazioni degli anni corrispondentemente alla versione che si vuole installare.

Knuth, D. E. (1984). *The TeXbook*. Addison Wesley. Nonostante l'anno di pubblicazione nell'arco degli anni sono state stampate molte ristampe rivedute e corrette.

Kopka, H. e Daly, P. W. (2004). *Guide to LaTeX*. Addison Wesley, 4ª edizione.

Lamport, L. (1994). *LaTeX: A document prparation system – User guide and reference manual*. Addison Wesley, 2ª edizione.

Lehman, P. (2011a). «The biblatex package». PDF document. Leggibile con `texdoc biblatex` nella distribuzione TeX Live.

— (2011b). «The etoolbox paclage». PDF document. Leggibile con `texdoc etoolbox` nella distribuzione TeX Live.

Lingnau, A. (2001). «An impoved environment for floats». PDF document. Leggibile con `texdoc float` nella distribuzione TeX Live.

Mittelbach, F. (2011). «The varioref package». PDF document. Leggibile con `texdoc varioref` nella distribuzione TeX Live.

Mittelbach, F. *et al.* (2003). «The DocStrip program». PDF document. Leggibile con `texdoc docstrip` nella distribuzione TeX Live.

— (2004). *The LaTeX Companion*. Addison Wesley, 2ª edizione.

— (2006). «LaTeX font encodings». PDF document. Leggibile con `texdoc encguide` nella distribuzione TeX Live.

Pakin, S. (2004). «How to package your LaTeX package». PDF document. Leggibile con `texdoc dtxtut` nella distribuzione TeX Live.

Pantieri, L. e Gordini, T. (2012). «L'arte di scrivere con LaTeX». PDF document. URL http://www.guitex.org/home/.

Rahtz, S. e Oberdiek, H. (2011). «Hypertext marks in LaTeX: a manual for hyperref». PDF document. Leggibile con `texdoc hyperref` nella distribuzione TeX Live.

Sommerfeldt, A. (2011a). «Customizing captions of floating environments using the caption package». PDF document. Leggibile con `texdoc caption` nella distribuzione TeX Live.

— (2011b). «The subcaption package». PDF document. Leggibile con `texdoc subcaption` nella distribuzione TeX Live.

Umeki, H. (2010). «The geometry package». PDF document. Leggibile con `texdoc geometry` nella distribuzione TeX Live.

van Oostrum, P. (2004). «Page layout in LaTeX». PDF document. Leggibile con `texdoc fancyhdr` nella distribuzione TeX Live.

Wilson, P. (2009). «A few notes on book design». PDF document. Leggibile con `texdoc memdesign` nella distribuzione TeX Live.

— (2010). «The tocloft package». PDF document. Leggibile con `texdoc tocloft` nella distribuzione TeX Live.

Wilson, P. e Madsen, L. (2011). «The memoir class for configurable typesetting – User guide». PDF document. Leggibile con `texdoc memoir` nella distribuzione TeX Live.