

**Università degli Studi di Bergamo**

**Dipartimento di Ingegneria Gestionale, dell’Informazione e della trasmissione**

**VolText**

**Manuale utente**

**Colombi Simone**

**Gambarara Alberto**

**Scarpellini Stefano**

**Indice**

1. **Introduzione**
2. **Grammatica**
   1. **Descrizione generale**
   2. **Sottoelementi**
   3. **Errori della grammatica**
3. **Tool utilizzati**
4. **Interfaccia utente**
5. **Possibili sviluppi futuri**
6. **Introduzione**

VolText è un progetto sviluppato nell’ambito del corso di Linguaggi Formali e Compilatori, tenuto dal professor Giuseppe Psaila nell’Anno Accademico 2020/2021.

L’obiettivo è la progettazione e implementazione di una libreria in Java per la generazione di un documento PDF e di una interfaccia grafica che permetta l’utilizzo semplificato di essa. La parte più inerente al corso è stata la realizzazione di una grammatica che descrive la struttura di un documento PDF usando i costrutti tipici della sintassi HTML, quali div, immagini, testi e liste, ma sfruttando una sintassi simile a quella di JSON.

La libreria è incentrata in particolare sullo sviluppo di documenti che abbiano una grafica semplice come volantini, piccoli annunci, avvisi, biglietti da visita, etc.

Il tool non vuole e non funge da sostituto di LaTeX ma prende gli elementi essenziali, creando una versione diversa sia in termini di linguaggio sia in termini di componenti, eliminando formule, grafici, tabelle, etc. non utili allo sviluppo di documenti grafici.

1. **Grammatica**
   1. **Descrizione generale**

La grammatica è stata progettata e implementata pensando a chi lavora con linguaggi web e che è familiare ad HTML.

Sfruttando una tipica sintassi del linguaggio JSON per la costruzione di tag personalizzati si è strutturata la grammatica come un documento PDF composto da vari elementi che possono essere disegnati all’interno in posizioni e dimensioni a piacimento.

Si riporta la grammatica e, per ogni regola, se ne descrive la funzione.

*A* pdfattr\* (stylesheet)? (page)+ *C*;

pdfattr: 'title:' *STRING* *ENDNLINE*

| 'author:' *STRING* *ENDNLINE*

| 'path:' *STRING* *ENDNLINE*;

stylesheet: 'stylesheet' *O* element\* *C*;

element: '@' *STRING* *O* attrStyle\* *C*;

attrStyle: imganumber

| positionv

| txtval;

page: 'page' *O* pageattr\* pae\* *C*;

pae: elemd | div;

div: 'div' *O* (color | idval | imganumber | elemd | positionv | fitAttr | figure | tvalue)\* *C*;

tvalue: 'cross-point:' *NVAL* *ENDNLINE*;

figure: 'shape:"' ('RECTANGLE' | 'CIRCLE' | 'TRIANGLE') *ENDLINE*;

elemd: text

| list

| img;

img: 'img' *O* imgattr\* imgElem imgattr\* *C*;

imgattr: (idval | imganumber | positionv | fitAttr);

imgElem: 'URL:' *STRING* *ENDNLINE*;

list: 'list' *O* (listattr | listElem)\* *C*;

listElem: 'item:' *STRING* *ENDNLINE*;

text: 'text' *O* txtattr\* txtElem\* txtattr\* txtElem\* txtattr\* *C*;

txtattr: (color | idval | imganumber | positionv | alignment | txtval | fitAttr);

txtElem: 'string:' *STRING* *ENDNLINE*;

//ATTRIBUTES

fitAttr: ('fit-x'

| 'fit-y'

) ':' *TFVAL* *ENDNLINE*;

imganumber: ( 'pos-x'

| 'pos-y'

| 'angle-rotation') ':' *NOTVAL*? *NVAL* (*UNIT*)? *ENDNLINE*

| ('height'

| 'width'

) ':' *NVAL* (*UNIT*)? *ENDNLINE*;

idval: 'id:' *STRING* *ENDNLINE*;

listattr: 'ordered:' *TFVAL* *ENDNLINE*

| 'bullet:' *STRING* *ENDNLINE*

| idval

| imganumber

| txtval

| positionv

| color

| colorBullet

| fitAttr;

txtval: ( 'font-family:'

| 'font-family-ttf:'

| 'font-family-otf:'

) *STRING* *ENDNLINE*

| 'font-size:' *NVAL* *ENDNLINE*

| *TXTATF* ':' *TFVAL* *ENDNLINE*;

colorBullet: 'colorT-bullet:' *STRING* *ENDNLINE*

| 'color-bullet:' *COLORVAL* *ENDNLINE*;

color: 'colorT:' *STRING* *ENDNLINE*

| 'color:' *COLORVAL* *ENDNLINE*;

positionv: 'position:' *POSVAL* *ENDNLINE*;

alignment: 'alignment:' *ALIGNVAL* *ENDLINE*;

pageattr: 'orientation:' *ORIENTATION* *ENDNLINE*

| 'oob:' *TFVAL* *ENDNLINE*

| 'width' ':' *NVAL* *ENDNLINE*

| 'height' ':' *NVAL* *ENDNLINE*

| 'format:' *FORMATVAL* *ENDNLINE*;

//TERMINALI

*NOTVAL*: '-';

*UNIT*: 'mm' | '%' | 'pt';

*FORMATVAL*: 'A0'

| 'A1'

| 'A2'

| 'A3'

| 'A4'

| 'A5'

| 'A6';

*ORIENTATION*: 'hor'

| 'ver';

*TXTATF*: 'bold'

| 'italics'

| 'underline';

*COLORVAL*: '#' ([0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F]

[0-9a-fA-F][0-9a-fA-F]);

*ALIGNVAL*: 'left'

| 'center'

| 'right'

| 'justify';

*TFVAL*: 'true'

| 'false';

*POSVAL*: 'lt'

| 'ct'

| 'rt'

| 'lc'

| 'cc'

| 'rc'

| 'lb'

| 'cb'

| 'rb' ;

*NVAL*: [0-9]+('.'[0-9]+)?;

*ENDLINE*: '";';

*ENDNLINE*: ';';

*O* : ':{';

*A* : '{';

*C* : '}';

*STRING*: '"' (~[\t\r\n])+ '"';

*WS* : [ \t\r\n]+ **->** skip ; // skip spaces, tabs, newlines

La prima regola, *pdf*, permette l’apertura del documento, l’inserimento di attributi quali titolo, autore e percorso di salvataggio del pdf, relativi al documento, la possibilità di inserire una parte di stile del documento (non implementata ma disponibile per futuri sviluppi), una o più pagine e la chiusura del documento.

Una struttura sommaria del tag *stylesheet* è stata realizzata come

stylesheet:{

@IdElemento1:{

<< attributi relativi a quell’elemento, quali *imganumber*, *positionv* e *txtval*>>

}

}

la pagina può contenere vari attributi quali orientamento, oob (out-of-bound), larghezza altezza e formato

1. **Tool utilizzati**

Per lo sviluppo sono stati utilizzati diversi strumenti:

* GitHub: GitHub è un servizio di Version Control utilizzato nel nostro progetto per tenere traccia delle modifiche e dei mutamenti implementati nel nostro software. Ciò ha aiutato anche la condivisione quasi real-time di codice per un lavoro di gruppo migliore.
* GitHub Desktop: client che permette un utilizzo semplificato e rapido di GitHub, utile anche per tenere traccia dei cambiamenti e delle versioni del progetto.
* ANTLR 4: ANTLR (ANother Tool for Language Recognition) è un potente generatore di parser per leggere, elaborare, eseguire o tradurre testo strutturato o file binari. Da una grammatica, ANTLR genera un parser che può costruire e analizzare alberi di analisi. Nella versione 4, aggiornata rispetto alla versione utilizzata durante il corso, sono stati fatti vari cambiamenti <…..>
* Eclipse: IDE per lo sviluppo della libreria e della GUI, che supporta anche vari plugin per ANTLR e JavaFX.
* JavaFX: libreria che permette l’implementazione di una interfaccia grafica per un programma java. In particolare, per Eclipse, è stato utilizzato il plugin e(fx)clipse che permette la creazione di progetti JavaFX e l’utilizzo delle librerie annesse.
* Launch4j: software per l’incapsulamento di eseguibili di tipo exe altamente personalizzabili a partire da eseguibili di tipo jar.

1. **Interfaccia utente**

VolText

1. **Possibili sviluppi futuri**

VolText