Gerador de PDF para casamentos.

Compiladores 2 - Trabalho 3
DC/UFSCar

Rodrigo Pesse de Abreu, R.A: 726588
Gabrieli Santos, R.A: 726523

Apresentação geral da linguagem

Apresentação da gramática final

```
grammar casamento;

programa:
    titulo
    numConvidados
    data
    orcamento
    listaPadrinhos
    listaPresentes
    listaConvidados
    listaServicos
;

titulo:
    'casamento:' STRING
;

numConvidados:
    'convidados:' NUM_INT
;

data:
    'data:' DATA
;

orcamento:
        'orcamento:' 'R$' NUM_REAL
;
```

```
listaPadrinhos:
    'padrinhos:' '['
        padrinho*
    ']'
;

padrinho:
    sigla=NOME ':' completo=STRING
;

listaPresentes:
    'presentes:' '['
        presente*
    ']'
;

presente:
    '{'
        descricao=STRING ','
        url=STRING ','
        '[' (nome+=NOME (',' nome+=NOME)*)? ']'
    '}'
;

listaConvidados:
    'convidados:' '['
        STRING (',' STRING)*
    ']'
;
```

```
60  listaServicos:
61      'servicos:' '{'
62          'fotografo:' fotografo+
63          ('buffet:' buffet+)?
64          'cerimonial:' cerimonial+
65          'local:' local+
66          'musica:' musica+
67          'decoracao:' decoracao+
68          'convites:' convites+
69          'lua_de_mel:' lua_de_mel+
70      '}'
71  ;
72
73  fotografo:
74      '{'
75          'nome:' nome=STRING
76          'contato:' contato=NUM_INT
77          'preco:' 'R$' preco=NUM_REAL
78      '}'
79  ;
80
81  buffet:
82      '{'
83          'nome:' STRING
84          'contato:' NUM_INT
85          'preco:' 'R$' NUM_REAL
86      '}'
87  ;
```

```
89  cerimonial:
90      '{'
91          'nome:' STRING
92          'contato:' NUM_INT
93          'preco:' 'R$' NUM_REAL
94      '}'
95  ;
96
97  local:
98      '{'
99          'nome:' nome=STRING
100         'endereco:' endereco=STRING
101         'contato:' contato=NUM_INT
102         'horario_inicio:' horario_inic=HORARIO
103         'horario_fim:' horario_fim=HORARIO
104         'preco:' 'R$' NUM_REAL
105         'capacidade:' capacidade=NUM_INT
106     '}'
107 ;
108
109 musica:
110     '{'
111         'nome:' STRING
112         'contato:' NUM_INT
113         'preco:' 'R$' NUM_REAL
114         'instrumento:' STRING (',' STRING)*
115         'musica:' STRING (',' STRING)*
116     '}'
117 ;
```

```
119     decoracao:
120         '{'
121             'nome:' nome=STRING
122             'contato:' NUM_INT
123             'preco:' 'R$' NUM_REAL
                'itens_decoracao:' item+=STRING (',' item+=STRING)*
125         '}'
126     ;
127
128     convites:
129         '{'
130             'nome:' STRING
131             'contato:' contato=NUM_INT
132             'quantidade_convites:' quant_convites=NUM_INT
133             'preco_unidade:' 'R$' NUM_REAL
134         '}'
135     ;
136
137     lua_de_mel:
138         '{'
139             'local:' loc=STRING
140             'hospedagem:' hospedagem=STRING
141             'contato_hospedagem:' NUM_INT
142             'preco_total:' 'R$' preco =NUM_REAL
143             'data_ida:' data_ida=DATA
144             'data_volta:' data_volta=DATA
145             'valor_passagem:' 'R$' passagem =NUM_REAL
146         '}'
147     ;
```

```
149     NOME: [a-z][a-z0-9]+
150     ;
151
152     //STRING: ('a'..'z' | 'A'..'Z' | '_' | ' ' | '\r' | '\n') ('a'..'z' | 'A'..'Z' | '0'..'9' | '_' | ' ' | '\r' | '\n')*
153     STRING: '"' ~('"'|'\n')* '"'
154     ;
155
156     DATA: ('0'..'3') ('0'..'9') '/' ('0'..'1') ('0'..'9') '/' ('0'..'2') ('0'..'9') ('0'..'9') ('0'..'9')
157     ;
158
159     HORARIO:
160         ('0'..'2')('0'..'9') 'h' ('0'..'5')('0'..'9')
161     ;
162
163     NUM_INT: ('0'..'9')+
164     ;
165
166     NUM_REAL : ('0' .. '9')+ '.' ('0' .. '9')+
167     ;
168
169     WS: (' ' | '\r' | '\n' |'\t')+ -> skip;
```

Descrição da análise semântica

Entrada na Tabela de Símbolos

```
public class EntradaTabelaDeSimbolos {
    private String sigla, casal;
    private String presente, site;
    private String[] siglas;

    public EntradaTabelaDeSimbolos(String sigla, String casal) {
        this.sigla = sigla;
        this.casal = casal;
    }

    public EntradaTabelaDeSimbolos(String presente, String site, String sigla){
        this.presente = presente;
        this.site = site;
        this.sigla = sigla;
    }
```

Tabela de Símbolos

```
public class TabelaDeSimbolos {
    private final String escopo;
    private final List<EntradaTabelaDeSimbolos> simbolos;
    private LinkedList<TabelaDeSimbolos> pilha;

    public TabelaDeSimbolos(String escopo) {
        simbolos = new ArrayList<EntradaTabelaDeSimbolos>();
        this.escopo = escopo;
    }

    public void adicionarSimbolo(String sigla, String casal) {
        simbolos.add(new EntradaTabelaDeSimbolos(sigla,casal));
    }

    public void adicionarSimbolo(String presente, String site, String sigla){
        simbolos.add(new EntradaTabelaDeSimbolos(presente,site,sigla));
    }

     public boolean existeSimbolo(String sigla) {
        return simbolos.stream().anyMatch((etds) -> (etds.getSigla().equals(sigla)));
    }
```

Análise Semântica

```
@Override
public Void visitPrograma(casamentoParser.ProgramaContext ctx) {
    pilhaDeTabelas.empilhar(new TabelaDeSimbolos("global"));
    super.visitPrograma(ctx);
    pilhaDeTabelas.desempilhar();

    return null;
}

@Override
public Void visitPadrinho(casamentoParser.PadrinhoContext ctx) {
    String textoSigla = ctx.NOME().getText();
    String textoCasal = ctx.STRING().getText();

    //se a sigla de padrinho ja esta na tabela de simbolos
    if (pilhaDeTabelas.existeSimbolo(textoSigla)) {
        imprimirMensagemErroSemantico("A sigla " + textoSigla + " ja foi atribuida anteriormente para outro casal de padrinhos");
    } else {
        pilhaDeTabelas.topo().adicionarSimbolo(textoSigla, textoCasal);
    }
    return null;
}
```

```
@Override
public Void visitListaPresentes(casamentoParser.ListaPresentesContext ctx) {
    pilhaDeTabelas.empilhar(new TabelaDeSimbolos("local"));
    super.visitListaPresentes(ctx);
    pilhaDeTabelas.desempilhar();

    return null;
}

@Override
public Void visitPresente(casamentoParser.PresenteContext ctx) {
    String textoPresente = ctx.descricao.getText();
    String textoSite = ctx.url.getText();
    String textoSigla = ctx.NOME(0).getText();

    if (!pilhaDeTabelas.existeSimbolo(textoSigla)) {
        imprimirMensagemErroSemantico("O casal " + textoSigla + " nao sao padrinhos");
    } else {
        if (!pilhaDeTabelas.topo().existeSimbolo(textoSigla)) {
            pilhaDeTabelas.topo().adicionarSimbolo(textoPresente, textoSite, textoSigla);
        } else {
            imprimirMensagemErroSemantico("O casal " + textoSigla + " foi escalado para mais de um presente");
        }
    }
```

Descrição da geração de código/interpretação

Biblioteca itext

https://itextpdf.com/en

```
public Void visitNumConvidados(casamentoParser.NumConvidadosContext ctx){
    Paragraph p = new Paragraph();
    doc.add(new Paragraph("Número de convidados: "+ctx.NUM_INT().getText()).setFontSize(16));
    return super.visitNumConvidados(ctx);
}
```

```
doc.add(new Paragraph("Lista de presentes:").setFontSize(16));
float [] pointColumnWidths = {50F,250F, 220F};
Table tablePresentes = new Table(pointColumnWidths);
tablePresentes.addCell(new Cell().add((“”)));
tablePresentes.addCell(new Cell().add(("Descrição")));
tablePresentes.addCell(new Cell().add(("URL")));
//aprensentando todos os presentes

for(int i=0; i<ctx.presente().size(); i++ ){
    tablePresentes.addCell(new Cell().add((i+1+“”)));
    tablePresentes.addCell(new Cell().add(ctx.presente(i).descricao.getText()));
    tablePresentes.addCell(new Cell().add(ctx.presente(i).url.getText()));
}
```

```
tablePresentes.setKeepTogether(true);
doc.add(tablePresentes);
```

Apresentação de exemplos

Apresentação do pdf

Dificuldades encontradas

- Grupo de duas pessoas
- Utilização da biblioteca iText (Maven, usabilidade)
- Netbeans lento
- Dificuldade em inserir na tabela de símbolos um array de padrinhos, enquanto cada elemento do array deveria ser comparado com os padrinhos já inseridos